

# Lecture Notes in Computer Science

2351

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

**Springer**

*Berlin*

*Heidelberg*

*New York*

*Barcelona*

*Hong Kong*

*London*

*Milan*

*Paris*

*Tokyo*

Anders Heyden Gunnar Sparr  
Mads Nielsen Peter Johansen (Eds.)

# Computer Vision – ECCV 2002

7th European Conference on Computer Vision  
Copenhagen, Denmark, May 28-31, 2002  
Proceedings, Part II



Springer

## Series Editors

Gerhard Goos, Karlsruhe University, Germany  
Juris Hartmanis, Cornell University, NY, USA  
Jan van Leeuwen, Utrecht University, The Netherlands

## Volume Editors

Anders Heyden  
Gunnar Sparr  
Lund University, Centre for Mathematical Sciences  
Box 118, 22100 Lund, Sweden  
E-mail: {Anders.Heyden,Gunnar.Sparr}@math.lth.se

Mads Nielsen  
The IT University of Copenhagen  
Glentevej 67-69, 2400 Copenhagen NW, Denmark  
E-mail: malte@itu.dk

Peter Johansen  
University of Copenhagen  
Universitetsparken 1, 2100 Copenhagen, Denmark  
E-mail: peterjo@diku.dk

## Cataloging-in-Publication Data applied for

### Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Computer vision : proceedings / ECCV 2002, 7th European Conference on  
Computer Vision, Copenhagen, Denmark, May 28 - 31, 2002. Anders Heyden ...  
(ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ;  
Milan ; Paris ; Tokyo : Springer  
Pt. 2 . - 2002

(Lecture notes in computer science ; Vol. 2351)  
ISBN 3-540-43744-4

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

ISSN 0302-9743

ISBN 3-540-43744-4 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 for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York  
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2002  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna e.K.  
Printed on acid-free paper      SPIN: 10870033      06/3142      5 4 3 2 1 0



# Preface

Premiering in 1990 in Antibes, France, the European Conference on Computer Vision, ECCV, has been held biennially at venues all around Europe. These conferences have been very successful, making ECCV a major event to the computer vision community.

ECCV 2002 was the seventh in the series. The privilege of organizing it was shared by three universities: The IT University of Copenhagen, the University of Copenhagen, and Lund University, with the conference venue in Copenhagen. These universities lie geographically close in the vivid Öresund region, which lies partly in Denmark and partly in Sweden, with the newly built bridge (opened summer 2000) crossing the sound that formerly divided the countries.

We are very happy to report that this year's conference attracted more papers than ever before, with around 600 submissions. Still, together with the conference board, we decided to keep the tradition of holding ECCV as a single track conference. Each paper was anonymously refereed by three different reviewers. For the final selection, for the first time for ECCV, a system with area chairs was used. These met with the program chairs in Lund for two days in February 2002 to select what became 45 oral presentations and 181 posters. Also at this meeting the selection was made without knowledge of the authors' identity.

The high-quality of the scientific program of ECCV 2002 would not have been possible without the dedicated cooperation of the 15 area chairs, the 53 program committee members, and all the other scientists, who reviewed the papers. A truly impressive effort was made. The spirit of this process reflects the enthusiasm in the research field, and you will find several papers in these proceedings that define the state of the art in the field.

Bjarne Ersbøll as Industrial Relations Chair organized the exhibitions at the conference. Magnus Oskarsson, Sven Spanne, and Nicolas Guilbert helped to make the review process and the preparation of the proceedings function smoothly. Ole Fogh Olsen gave us valuable advice on editing the proceedings. Camilla Jørgensen competently headed the scientific secretariat. Erik Dam and Dan Witzner were responsible for the ECCV 2002 homepage. David Vernon, who chaired ECCV 2000 in Dublin, was extremely helpful during all stages of our preparation for the conference. We would like to thank all these people, as well as numerous others who helped in various respects. A special thanks goes to Søren Skovsgaard at the Congress Consultants, for professional help with all practical matters.

We would also like to thank Rachid Deriche and Theo Papadopoulou for making their web-based conference administration system available and adjusting it to ECCV. This was indispensable in handling the large number of submissions and the thorough review and selection procedure.

Finally, we wish to thank the IT University of Copenhagen and its president Mads Tofte for supporting the conference all the way from planning to realization.

March 2002

Anders Heyden  
Gunnar Sparr  
Mads Nielsen  
Peter Johansen

# Organization

## Conference Chair

Peter Johansen

Copenhagen University, Denmark

## 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

Bernd Neumann

University of Hamburg, Germany

Giulio Sandini

University of Genova, Italy

David Vernon

Trinity College, Dublin, Ireland

## Program Chairs

Anders Heyden

Lund University, Sweden

Gunnar Sparr

Lund University, Sweden

## Area Chairs

Ronen Basri

Weizmann Institute, Israel

Michael Black

Brown University, USA

Andrew Blake

Microsoft Research, UK

Rachid Deriche

INRIA, Sophia Antipolis, France

Jan-Olof Eklundh

Royal Institute of Technology, Sweden

Lars Kai Hansen

Denmark Technical University, Denmark

Steve Maybank

University of Reading, UK

Theodore Papadopoulos

INRIA, Sophia Antipolis, France

Cordelia Schmid

INRIA, Rhône-Alpes, France

Amnon Shashua

The Hebrew University of Jerusalem, Israel

Stefano Soatto

University of California, Los Angeles, USA

Bill Triggs

INRIA, Rhône-Alpes, France

Luc van Gool

K.U. Leuven, Belgium &

ETH, Zürich, Switzerland

Joachim Weichert

Saarland University, Germany

Andrew Zisserman

University of Oxford, UK

## Program Committee

Luis Alvarez	University of Las Palmas, Spain
Padmanabhan Anandan	Microsoft Research, USA
Helder Araujo	University of Coimbra, Portugal
Serge Belongie	University of California, San Diego, USA
Marie-Odile Berger	INRIA, Lorraine, France
Aaron Bobick	Georgia Tech, USA
Terry Boulton	Lehigh University, USA
Francois Chaumette	INRIA, Rennes, France
Laurent Cohen	Université Paris IX Dauphine, France
Tim Cootes	University of Manchester, UK
Kostas Daniilidis	University of Pennsylvania, USA
Larry Davis	University of Maryland, USA
Frank Ferrie	McGill University, USA
Andrew Fitzgibbon	University of Oxford, UK
David J. Fleet	Xerox Palo Alto Research Center, USA
David Forsyth	University of California, Berkeley, USA
Pascal Fua	EPFL, Switzerland
Richard Hartley	Australian National University, Australia
Vaclav Hlavac	Czech Technical University, Czech Republic
Michal Irani	Weizmann Institute, Israel
Allan Jepson	University of Toronto, Canada
Peter Johansen	Copenhagen University, Denmark
Fredrik Kahl	Lund University, Sweden
Sing Bing Kang	Microsoft Research, USA
Ron Kimmel	Technion, Israel
Kyros Kutulakos	University of Rochester, USA
Tony Lindeberg	Royal Institute of Technology, Sweden
Jim Little	University of British Columbia, Canada
Peter Meer	Rutgers University, USA
David Murray	University of Oxford, UK
Nassir Navab	Siemens, USA
Mads Nielsen	IT-University of Copenhagen, Denmark
Patrick Perez	Microsoft Research, UK
Pietro Perona	California Institute of Technology, USA
Marc Pollefeys	K.U. Leuven, Belgium
Long Quan	Hong Kong University of Science and Technology, Hong Kong
Ian Reid	University of Oxford, UK
Nicolas Rougon	Institut National des Télécommunications, France
José Santos-Victor	Instituto Superior Técnico, Lisbon, Portugal
Guillermo Sapiro	University of Minnesota, USA
Yoichi Sato	IIS, University of Tokyo, Japan
Bernt Schiele	ETH, Zürich, Switzerland
Arnold Smeulders	University of Amsterdam, The Netherlands

Gerald Sommer	University of Kiel, Germany
Peter Sturm	INRIA, Rhône-Alpes, France
Tomas Svoboda	Swiss Federal Institute of Technology, Switzerland
Chris Taylor	University of Manchester, UK
Phil Torr	Microsoft Research, UK
Panos Trahanias	University of Crete, Greece
Laurent Younes	CMLA, ENS de Cachan, France
Alan Yuille	Smith-Kettlewell Eye Research Institute, USA
Josiane Zerubia	INRIA, Sophia Antipolis, France
Kalle Åström	Lund University, Sweden

## Additional Referees

Henrik Aanaes	Jeffrey E. Boyd	Michael Elad
Manoj Aggarwal	Edmond Boyer	Ahmed Elgammal
Motilal Agrawal	Yuri Boykov	Ronan Fablet
Aya Aner	Chen Brestel	Ayman Farahat
Adnan Ansar	Lars Bretzner	Olivier Faugeras
Mirko Appel	Alexander Brook	Paulo Favaro
Tal Arbel	Michael Brown	Xiaolin Feng
Okan Arikan	Alfred Bruckstein	Vittorio Ferrari
Akira Asano	Thomas Buelow	Frank Ferrie
Shai Avidan	Joachim Buhmann	Mario Figueireda
Simon Baker	Hans Burkhardt	Margaret Fleck
David Barger	Bernard Buxton	Michel Gangnet
Christian Barillot	Nikos Canterakis	Xiang Gao
Kobus Barnard	Yaron Caspi	D. Geiger
Adrien Bartoli	Alessandro Chiuso	Yakup Genc
Benedicte Bascle	Roberto Cipolla	Bogdan Georgescu
Pierre-Louis Bazin	Dorin Comaniciu	J.-M. Geusebroek
Isabelle Begin	Kurt Cornelis	Christopher Geyer
Stephen Benoit	Antonio Criminisi	Peter Giblin
Alex Berg	Thomas E. Davis	Gerard Giraudon
James Bergen	Nando de Freitas	Roman Goldenberg
Jim Bergen	Fernando de la Torre	Shaogang Gong
Marcelo Bertalmio	Daniel DeMenthon	Hayit Greenspan
Rikard Berthilsson	Xavier Descombes	Lewis Griffin
Christophe Biernacki	Hagio Djambazian	Jens Guehring
Armin Biess	Gianfranco Doretto	Yanlin Guo
Alessandro Bissacco	Alessandro Duci	Daniela Hall
Laure Blanc-Feraud	Gregory Dudek	Tal Hassner
Ilya Blayvas	Ramani Duraiswami	Horst Haussecker
Eran Borenstein	Pinar Duygulu	Ralf Hebrich
Patrick Bouthemy	Michael Eckmann	Yacov Hel-Or
Richard Bowden	Alyosha Efros	Lorna Herda

Shinsaku Hiura	Roberto Manduchi	Garbis Salgian
Jesse Hoey	Petros Maragos	Frank Sauer
Stephen Hsu	Eric Marchand	Peter Savadjiev
Du Huynh	Jiri Matas	Silvio Savarese
Naoyuki Ichimura	Bogdan Matei	Harpreet Sawhney
Slobodan Ilic	Esther B. Meier	Frederik Schaffalitzky
Sergey Ioffe	Jason Meltzer	Yoav Schechner
Michael Isard	Etienne Mémín	Chrostoph Schnoerr
Volkan Isler	Rudolf Mester	Stephan Scholze
David Jacobs	Ross J. Micheals	Ali Shahrokri
Bernd Jaehne	Anurag Mittal	Doron Shaked
Ian Jermyn	Hiroshi Mo	Eitan Sharon
Hailin Jin	William Moran	Eli Shechtman
Marie-Pierre Jolly	Greg Mori	Jamie Sherrah
Stiliyan-N. Kalitzin	Yael Moses	Akinobu Shimizu
Behrooz Kamgar-Parsi	Jane Mulligan	Ilan Shimshoni
Kenichi Kanatani	Don Murray	Kaleem Siddiqi
Danny Keren	Masahide Naemura	Hedvig Sidenbladh
Erwan Kerrien	Kenji Nagao	Robert Sim
Charles Kervrann	Mirko Navara	Denis Simakov
Renato Keshet	Shree Nayar	Philippe Simard
Ali Khamene	Oscar Nestares	Eero Simoncelli
Shamim Khan	Bernd Neumann	Nir Sochen
Nahum Kiryati	Jeffrey Ng	Yang Song
Reinhard Koch	Tat Hieu Nguyen	Andreas Soupliotis
Ullrich Koethe	Peter Nillius	Sven Spanne
Esther B. Koller-Meier	David Nister	Martin Spengler
John Krumm	Alison Noble	Alon Spira
Hannes Kruppa	Tom O'Donnell	Thomas Strömberg
Murat Kunt	Takayuki Okatani	Richard Szeliski
Prasun Lala	Nuria Olivier	Hai Tao
Michael Langer	Ole Fogh Olsen	Huseyin Tek
Ivan Laptev	Magnus Oskarsson	Seth Teller
Jean-Pierre Le Cadre	Nikos Paragios	Paul Thompson
Bastian Leibe	Ioannis Patras	Jan Tops
Ricahrd Lengagne	Josef Pauli	Benjamin J. Tordoff
Vincent Lepetit	Shmuel Peleg	Kentaro Toyama
Thomas Leung	Robert Pless	Tinne Tuytelaars
Maxime Lhuillier	Swaminathan Rahul	Shimon Ullman
Weiliang Li	Deva Ramanan	Richard Unger
David Liebowitz	Lionel Reveret	Raquel Urtasun
Georg Lindgren	Dario Ringach	Sven Utcke
David Lowe	Ruth Rosenholtz	Luca Vacchetti
John MacCormick	Volker Roth	Anton van den Hengel
Henrik Malm	Payam Saisan	Geert Van Meerbergen

Pierre Vandergheynst  
Zhizhou Wang  
Baba Vemuri  
Frank Verbiest  
Maarten Vergauwen  
Jaco Vermaak  
Mike Werman  
David Vernon  
Thomas Vetter

Rene Vidal  
Michel Vidal-Naquet  
Marta Wilczkowiak  
Ramesh Visvanathan  
Dan Witzner Hansen  
Julia Vogel  
Lior Wolf  
Bob Woodham  
Robert J. Woodham

Chenyang Xu  
Yaser Yacoob  
Anthony Yezzi  
Ramin Zabih  
Hugo Zaragoza  
Lihi Zelnik-Manor  
Ying Zhu  
Assaf Zomet

## Table of Contents, Part II

### Surface Geometry

A Variational Approach to Recovering a Manifold from Sample Points . . . . .	3
<i>J. Gomes, A. Mojsilovic</i>	
A Variational Approach to Shape from Defocus . . . . .	18
<i>H. Jin, P. Favaro</i>	
Shadow Graphs and Surface Reconstruction . . . . .	31
<i>Y. Yu, J.T. Chang</i>	
Specularities Reduce Ambiguity of Uncalibrated Photometric Stereo . . . . .	46
<i>O. Drbohlav, R. Šára</i>	

### Grouping and Segmentation

Pairwise Clustering with Matrix Factorisation and the EM Algorithm . . . . .	63
<i>A. Robles-Kelly, E.R. Hancock</i>	
Shape Priors for Level Set Representations . . . . .	78
<i>M. Rousson, N. Paragios</i>	
Nonlinear Shape Statistics in Mumford–Shah Based Segmentation . . . . .	93
<i>D. Cremers, T. Kohlberger, C. Schnörr</i>	
Class-Specific, Top-Down Segmentation . . . . .	109
<i>E. Borenstein, S. Ullman</i>	

### Structure from Motion / Stereoscopic Vision / Surface Geometry / Shape

Quasi-Dense Reconstruction from Image Sequence . . . . .	125
<i>M. Lhuillier, L. Quan</i>	
Properties of the Catadioptric Fundamental Matrix . . . . .	140
<i>C. Geyer, K. Daniilidis</i>	
Building Architectural Models from Many Views Using Map Constraints . . . . .	155
<i>D.P. Robertson, R. Cipolla</i>	
Motion – Stereo Integration for Depth Estimation . . . . .	170
<i>C. Strecha, L. Van Gool</i>	

Lens Distortion Recovery for Accurate Sequential Structure and Motion Recovery . . . . .	186
<i>K. Cornelis, M. Pollefeys, L. Van Gool</i>	
Generalized Rank Conditions in Multiple View Geometry with Applications to Dynamical Scenes . . . . .	201
<i>K. Huang, R. Fossum, Y. Ma</i>	
Dense Structure-from-Motion: An Approach Based on Segment Matching . . . . .	217
<i>F. Ernst, P. Wilinski, K. van Overveld</i>	
Maximizing Rigidity: Optimal Matching under Scaled-Orthography . . . . .	232
<i>J. Maciel, J. Costeira</i>	
Dramatic Improvements to Feature Based Stereo . . . . .	247
<i>V.N. Smelyansky, R.D. Morris, F.O. Kuehnel, D.A. Maluf, P. Cheeseman</i>	
Motion Curves for Parametric Shape and Motion Estimation . . . . .	262
<i>P.-L. Bazin, J.-M. Vézien</i>	
Bayesian Self-Calibration of a Moving Camera . . . . .	277
<i>G. Qian, R. Chellappa</i>	
Balanced Recovery of 3D Structure and Camera Motion from Uncalibrated Image Sequences . . . . .	294
<i>B. Georgescu, P. Meer</i>	
Linear Multi View Reconstruction with Missing Data . . . . .	309
<i>C. Rother, S. Carlsson</i>	
Model-Based Silhouette Extraction for Accurate People Tracking . . . . .	325
<i>R. Plaenkers, P. Fua</i>	
On the Non-linear Optimization of Projective Motion Using Minimal Parameters .	340
<i>A. Bartoli</i>	
Structure from Many Perspective Images with Occlusions . . . . .	355
<i>D. Martinec, T. Pajdla</i>	
Sequence-to-Sequence Self Calibration . . . . .	370
<i>L. Wolf, A. Zomet</i>	
Structure from Planar Motions with Small Baselines . . . . .	383
<i>R. Vidal, J. Oliensis</i>	
Revisiting Single-View Shape Tensors: Theory and Applications . . . . .	399
<i>A. Levin, A. Shashua</i>	



Tracking and Rendering Using Dynamic Textures on Geometric Structure from Motion .....	415
<i>D. Cobzas, M. Jagersand</i>	
Sensitivity of Calibration to Principal Point Position .....	433
<i>R.I. Hartley, R. Kaucic</i>	
Critical Curves and Surfaces for Euclidean Reconstruction .....	447
<i>F. Kahl, R. Hartley</i>	
View Synthesis with Occlusion Reasoning Using Quasi-Sparse Feature Correspondences .....	463
<i>D. Jelinek, C.J. Taylor</i>	
Eye Gaze Correction with Stereovision for Video-Teleconferencing .....	479
<i>R. Yang, Z. Zhang</i>	
Wavelet-Based Correlation for Stereopsis .....	495
<i>M. Clerc</i>	
Stereo Matching Using Belief Propagation .....	510
<i>J. Sun, H.-Y. Shum, N.-N. Zheng</i>	
Symmetric Sub-pixel Stereo Matching .....	525
<i>R. Szeliski, D. Scharstein</i>	
New Techniques for Automated Architectural Reconstruction from Photographs .....	541
<i>T. Werner, A. Zisserman</i>	
Stereo Matching with Segmentation-Based Cooperation .....	556
<i>Y. Zhang, C. Kambhamettu</i>	
Coarse Registration of Surface Patches with Local Symmetries .....	572
<i>J. Vanden Wyngaerd, L. Van Gool</i>	
Multiview Registration of 3D Scenes by Minimizing Error between Coordinate Frames .....	587
<i>G.C. Sharp, S.W. Lee, D.K. Wehe</i>	
Recovering Surfaces from the Restoring Force .....	598
<i>G. Kamberov, G. Kamberova</i>	
Interpolating Sporadic Data .....	613
<i>L. Noakes, R. Kozera</i>	
Highlight Removal Using Shape-from-Shading .....	626
<i>H. Ragheb, E.R. Hancock</i>	

A Reflective Symmetry Descriptor . . . . .	642
<i>M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, T. Funkhouser</i>	
Gait Sequence Analysis Using Frieze Patterns . . . . .	657
<i>Y. Liu, R. Collins, Y. Tsin</i>	
Feature-Preserving Medial Axis Noise Removal . . . . .	672
<i>R. Tam, W. Heidrich</i>	
Hierarchical Shape Modeling for Automatic Face Localization . . . . .	687
<i>C. Liu, H.-Y. Shum, C. Zhang</i>	
Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data . . . . .	704
<i>S. Ilic, P. Fua</i>	
Transitions of the 3D Medial Axis under a One-Parameter Family of Deformations . . . . .	718
<i>P. Giblin, B.B. Kimia</i>	
Learning Shape from Defocus . . . . .	735
<i>P. Favaro, S. Soatto</i>	
A Rectilinearity Measurement for Polygons . . . . .	746
<i>J. Žunić, P.L. Rosin</i>	
Local Analysis for 3D Reconstruction of Specular Surfaces – Part II . . . . .	759
<i>S. Savarese, P. Perona</i>	
Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration . . . . .	775
<i>N. Paragios, M. Rousson, V. Ramesh</i>	
Shape from Shading and Viscosity Solutions . . . . .	790
<i>E. Prados, O. Faugeras, E. Rouy</i>	
Model Acquisition by Registration of Multiple Acoustic Range Views . . . . .	805
<i>A. Fusiello, U. Castellani, L. Ronchetti, V. Murino</i>	
<b>Structure from Motion</b>	
General Trajectory Triangulation . . . . .	823
<i>J.Y. Kaminski, M. Teicher</i>	
Surviving Dominant Planes in Uncalibrated Structure and Motion Recovery . . . . .	837
<i>M. Pollefeys, F. Verbiest, L. Van Gool</i>	
A Bayesian Estimation of Building Shape Using MCMC . . . . .	852
<i>A.R. Dick, P.H.S. Torr, R. Cipolla</i>	

Structure and Motion for Dynamic Scenes – The Case of Points Moving in Planes . . . . .	867
<i>P. Sturm</i>	
What Does the Scene Look Like from a Scene Point? . . . . .	883
<i>M. Irani, T. Hassner, P. Anandan</i>	
<b>Author Index</b> . . . . .	899

# Table of Contents, Part I

## Active and Real-Time Vision

Tracking with the EM Contour Algorithm . . . . .	3
<i>A.E.C. Pece, A.D. Worrall</i>	
M2Tracker: A Multi-view Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo . . . . .	18
<i>A. Mittal, L.S. Davis</i>	

## Image Features

Analytical Image Models and Their Applications . . . . .	37
<i>A. Srivastava, X. Liu, U. Grenander</i>	
Time-Recursive Velocity-Adapted Spatio-Temporal Scale-Space Filters . . . . .	52
<i>T. Lindeberg</i>	
Combining Appearance and Topology for Wide Baseline Matching . . . . .	68
<i>D. Tell, S. Carlsson</i>	
Guided Sampling and Consensus for Motion Estimation . . . . .	82
<i>B. Tordoff, D.W. Murray</i>	

## Image Features / Visual Motion

Fast Anisotropic Gauss Filtering . . . . .	99
<i>J.-M. Geusebroek, A.W.M. Smeulders, J. van de Weijer</i>	
Adaptive Rest Condition Potentials: Second Order Edge-Preserving Regularization . . . . .	113
<i>M. Rivera, J.L. Marroquin</i>	
An Affine Invariant Interest Point Detector . . . . .	128
<i>K. Mikolajczyk, C. Schmid</i>	
Understanding and Modeling the Evolution of Critical Points under Gaussian Blurring . . . . .	143
<i>A. Kuijper, L. Florack</i>	
Image Processing Done Right . . . . .	158
<i>J.J. Koenderink, A.J. van Doorn</i>	
Multimodal Data Representations with Parameterized Local Structures . . . . .	173
<i>Y. Zhu, D. Comaniciu, S. Schwartz, V. Ramesh</i>	

The Relevance of Non-generic Events in Scale Space Models . . . . .	190
<i>A. Kuijper, L. Florack</i>	
The Localized Consistency Principle for Image Matching under Non-uniform Illumination Variation and Affine Distortion . . . . .	205
<i>B. Wang, K.K. Sung, T.K. Ng</i>	
Resolution Selection Using Generalized Entropies of Multiresolution Histograms . . . . .	220
<i>E. Hadjidemetriou, M.D. Grossberg, S.K. Nayar</i>	
Robust Computer Vision through Kernel Density Estimation . . . . .	236
<i>H. Chen, P. Meer</i>	
Constrained Flows of Matrix-Valued Functions: Application to Diffusion Tensor Regularization . . . . .	251
<i>C. Chef d'hotel, D. Tschumperlé, R. Deriche, O. Faugeras</i>	
A Hierarchical Framework for Spectral Correspondence . . . . .	266
<i>M. Carcassoni, E.R. Hancock</i>	
Phase-Based Local Features . . . . .	282
<i>G. Carneiro, A.D. Jepson</i>	
What Is the Role of Independence for Visual Recognition? . . . . .	297
<i>N. Vasconcelos, G. Carneiro</i>	
A Probabilistic Multi-scale Model for Contour Completion Based on Image Statistics . . . . .	312
<i>X. Ren, J. Malik</i>	
Toward a Full Probability Model of Edges in Natural Images . . . . .	328
<i>K.S. Pedersen, A.B. Lee</i>	
Fast Difference Schemes for Edge Enhancing Beltrami Flow . . . . .	343
<i>R. Malladi, I. Ravve</i>	
A Fast Radial Symmetry Transform for Detecting Points of Interest . . . . .	358
<i>G. Loy, A. Zelinsky</i>	
Image Features Based on a New Approach to 2D Rotation Invariant Quadrature Filters . . . . .	369
<i>M. Felsberg, G. Sommer</i>	
Representing Edge Models via Local Principal Component Analysis . . . . .	384
<i>P.S. Huggins, S.W. Zucker</i>	
Regularized Shock Filters and Complex Diffusion . . . . .	399
<i>G. Gilboa, N.A. Sochen, Y.Y. Zeevi</i>	

Multi-view Matching for Unordered Image Sets, or “How Do I Organize My Holiday Snaps?” .....	414
<i>F. Schaffalitzky, A. Zisserman</i>	
Parameter Estimates for a Pencil of Lines: Bounds and Estimators .....	432
<i>G. Speyer, M. Werman</i>	
Multilinear Analysis of Image Ensembles: TensorFaces .....	447
<i>M.A.O. Vasilescu, D. Terzopoulos</i>	
‘Dynamism of a Dog on a Leash’ or Behavior Classification by Eigen-Decomposition of Periodic Motions .....	461
<i>R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky</i>	
Automatic Detection and Tracking of Human Motion with a View-Based Representation .....	476
<i>R. Fablet, M.J. Black</i>	
Using Robust Estimation Algorithms for Tracking Explicit Curves .....	492
<i>J.-P. Tarel, S.-S. Ieng, P. Charbonnier</i>	
On the Motion and Appearance of Specularities in Image Sequences .....	508
<i>R. Swaminathan, S.B. Kang, R. Szeliski, A. Criminisi, S.K. Nayar</i>	
Multiple Hypothesis Tracking for Automatic Optical Motion Capture .....	524
<i>M. Ringer, J. Lasenby</i>	
Single Axis Geometry by Fitting Conics .....	537
<i>G. Jiang, H.-t. Tsui, L. Quan, A. Zisserman</i>	
Computing the Physical Parameters of Rigid-Body Motion from Video .....	551
<i>K.S. Bhat, S.M. Seitz, J. Popović, P.K. Khosla</i>	
Building Roadmaps of Local Minima of Visual Models .....	566
<i>C. Sminchisescu, B. Triggs</i>	
A Generative Method for Textured Motion: Analysis and Synthesis .....	583
<i>Y. Wang, S.-C. Zhu</i>	
Is Super-Resolution with Optical Flow Feasible? .....	599
<i>W.Y. Zhao, H.S. Sawhney</i>	
New View Generation with a Bi-centric Camera .....	614
<i>D. Weinshall, M.-S. Lee, T. Brodsky, M. Trajkovic, D. Feldman</i>	
Recognizing and Tracking Human Action .....	629
<i>J. Sullivan, S. Carlsson</i>	

Towards Improved Observation Models for Visual Tracking: Selective Adaptation . . . . .	645
<i>J. Vermaak, P. Pérez, M. Gangnet, A. Blake</i>	
Color-Based Probabilistic Tracking . . . . .	661
<i>P. Pérez, C. Hue, J. Vermaak, M. Gangnet</i>	
Dense Motion Analysis in Fluid Imagery . . . . .	676
<i>T. Corpetti, É. Mémin, P. Pérez</i>	
A Layered Motion Representation with Occlusion and Compact Spatial Support . . . . .	692
<i>A.D. Jepson, D.J. Fleet, M.J. Black</i>	
Incremental Singular Value Decomposition of Uncertain Data with Missing Values . . . . .	707
<i>M. Brand</i>	
Symmetrical Dense Optical Flow Estimation with Occlusions Detection . . . . .	721
<i>L. Alvarez, R. Deriche, T. Papadopoulo, J. Sánchez</i>	
Audio-Video Sensor Fusion with Probabilistic Graphical Models . . . . .	736
<i>M.J. Beal, H. Attias, N. Jojic</i>	
<b>Visual Motion</b>	
Increasing Space-Time Resolution in Video . . . . .	753
<i>E. Shechtman, Y. Caspi, M. Irani</i>	
Hyperdynamics Importance Sampling . . . . .	769
<i>C. Sminchisescu, B. Triggs</i>	
Implicit Probabilistic Models of Human Motion for Synthesis and Tracking . . . . .	784
<i>H. Sidenbladh, M.J. Black, L. Sigal</i>	
Space-Time Tracking . . . . .	801
<i>L. Torresani, C. Bregler</i>	
<b>Author Index</b> . . . . .	813

# Table of Contents, Part III

## Shape

3D Statistical Shape Models Using Direct Optimisation of Description Length .....	3
<i>R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, C.J. Taylor</i>	
Approximate Thin Plate Spline Mappings .....	21
<i>G. Donato, S. Belongie</i>	
DEFORMATION: Deforming Motion, Shape Average and the Joint Registration and Segmentation of Images .....	32
<i>S. Soatto, A.J. Yezzi</i>	
Region Matching with Missing Parts .....	48
<i>A. Duci, A.J. Yezzi, S. Mitter, S. Soatto</i>	

## Stereoscopic Vision I

What Energy Functions Can Be Minimized via Graph Cuts? .....	65
<i>V. Kolmogorov, R. Zabih</i>	
Multi-camera Scene Reconstruction via Graph Cuts .....	82
<i>V. Kolmogorov, R. Zabih</i>	
A Markov Chain Monte Carlo Approach to Stereovision .....	97
<i>J. S��n��gas</i>	
A Probabilistic Theory of Occupancy and Emptiness .....	112
<i>R. Bhotika, D.J. Fleet, K.N. Kutulakos</i>	

## Texture Shading and Colour / Grouping and Segmentation / Object Recognition

Texture Similarity Measure Using Kullback-Leibler Divergence between Gamma Distributions .....	133
<i>J.R. Mathiassen, A. Skavhaug, K. B��</i>	
All the Images of an Outdoor Scene .....	148
<i>S.G. Narasimhan, C. Wang, S.K. Nayar</i>	
Recovery of Reflectances and Varying Illuminants from Multiple Views .....	163
<i>Q.-T. Luong, P. Fua, Y. Leclerc</i>	



Composite Texture Descriptions . . . . .	180
<i>A. Zalesny, V. Ferrari, G. Caenen, D. Auf der Maur, L. Van Gool</i>	
Constructing Illumination Image Basis from Object Motion . . . . .	195
<i>A. Nakashima, A. Maki, K. Fukui</i>	
Diffuse-Specular Separation and Depth Recovery from Image Sequences . . . . .	210
<i>S. Lin, Y. Li, S.B. Kang, X. Tong, H.-Y. Shum</i>	
Shape from Texture without Boundaries . . . . .	225
<i>D.A. Forsyth</i>	
Statistical Modeling of Texture Sketch . . . . .	240
<i>Y.N. Wu, S.C. Zhu, C.-e. Guo</i>	
Classifying Images of Materials: Achieving Viewpoint and Illumination Independence . . . . .	255
<i>M. Varma, A. Zisserman</i>	
Estimation of Multiple Illuminants from a Single Image of Arbitrary Known Geometry . . . . .	272
<i>Y. Wang, D. Samaras</i>	
The Effect of Illuminant Rotation on Texture Filters: Lissajous's Ellipses . . . . .	289
<i>M. Chantler, M. Schmidt, M. Petrou, G. McGunnigle</i>	
On Affine Invariant Clustering and Automatic Cast Listing in Movies . . . . .	304
<i>A. Fitzgibbon, A. Zisserman</i>	
Factorial Markov Random Fields . . . . .	321
<i>J. Kim, R. Zabih</i>	
Evaluation and Selection of Models for Motion Segmentation . . . . .	335
<i>K. Kanatani</i>	
Surface Extraction from Volumetric Images Using Deformable Meshes: A Comparative Study . . . . .	350
<i>J. Tohka</i>	
DREAM <sup>2</sup> S: Deformable Regions Driven by an Eulerian Accurate Minimization Method for Image and Video Segmentation (Application to Face Detection in Color Video Sequences) . . . . .	365
<i>S. Jehan-Besson, M. Barlaud, G. Aubert</i>	
Neuro-Fuzzy Shadow Filter . . . . .	381
<i>B.P.L. Lo, G.-Z. Yang</i>	
Parsing Images into Region and Curve Processes . . . . .	393
<i>Z. Tu, S.-C. Zhu</i>	

Yet Another Survey on Image Segmentation: Region and Boundary Information Integration . . . . .	408
<i>J. Freixenet, X. Muñoz, D. Raba, J. Martí, X. Cufí</i>	
Perceptual Grouping from Motion Cues Using Tensor Voting in 4-D . . . . .	423
<i>M. Nicolescu, G. Medioni</i>	
Deformable Model with Non-euclidean Metrics . . . . .	438
<i>B. Taton, J.-O. Lachaud</i>	
Finding Deformable Shapes Using Loopy Belief Propagation . . . . .	453
<i>J.M. Coughlan, S.J. Ferreira</i>	
Probabilistic and Voting Approaches to Cue Integration for Figure-Ground Segmentation . . . . .	469
<i>E. Hayman, J.-O. Eklundh</i>	
Bayesian Estimation of Layers from Multiple Images . . . . .	487
<i>Y. Wexler, A. Fitzgibbon, A. Zisserman</i>	
A Stochastic Algorithm for 3D Scene Segmentation and Reconstruction . . . . .	502
<i>F. Han, Z. Tu, S.-C. Zhu</i>	
Normalized Gradient Vector Diffusion and Image Segmentation . . . . .	517
<i>Z. Yu, C. Bajaj</i>	
Spectral Partitioning with Indefinite Kernels Using the Nyström Extension . . . . .	531
<i>S. Belongie, C. Fowlkes, F. Chung, J. Malik</i>	
A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models . . . . .	543
<i>M. Harville</i>	
Multivariate Saddle Point Detection for Statistical Clustering . . . . .	561
<i>D. Comaniciu, V. Ramesh, A. Del Bue</i>	
Parametric Distributional Clustering for Image Segmentation . . . . .	577
<i>L. Hermes, T. Zöllner, J.M. Buhmann</i>	
Probabilistic Models and Informative Subspaces for Audiovisual Correspondence . . . . .	592
<i>J.W. Fisher, T. Darrell</i>	
Volterra Filtering of Noisy Images of Curves . . . . .	604
<i>J. August</i>	
Image Segmentation by Flexible Models Based on Robust Regularized Networks . . . . .	621
<i>M. Rivera, J. Gee</i>	

Principal Component Analysis over Continuous Subspaces and Intersection of Half-Spaces . . . . .	635
<i>A. Levin, A. Shashua</i>	
On Pencils of Tangent Planes and the Recognition of Smooth 3D Shapes from Silhouettes . . . . .	651
<i>S. Lazebnik, A. Sethi, C. Schmid, D. Kriegman, J. Ponce, M. Hebert</i>	
Estimating Human Body Configurations Using Shape Context Matching . . . . .	666
<i>G. Mori, J. Malik</i>	
Probabilistic Human Recognition from Video . . . . .	681
<i>S. Zhou, R. Chellappa</i>	
SoftPOSIT: Simultaneous Pose and Correspondence Determination . . . . .	698
<i>P. David, D. DeMenthon, R. Duraiswami, H. Samet</i>	
A Pseudo-Metric for Weighted Point Sets . . . . .	715
<i>P. Giannopoulos, R.C. Veltkamp</i>	
Shock-Based Indexing into Large Shape Databases . . . . .	731
<i>T.B. Sebastian, P.N. Klein, B.B. Kimia</i>	
EigenSegments: A Spatio-Temporal Decomposition of an Ensemble of Images . .	747
<i>S. Avidan</i>	
On the Representation and Matching of Qualitative Shape at Multiple Scales . . . . .	759
<i>A. Shokoufandeh, S. Dickinson, C. Jönsson, L. Bretzner, T. Lindeberg</i>	
Combining Simple Discriminators for Object Discrimination . . . . .	776
<i>S. Mahamud, M. Hebert, J. Lafferty</i>	
Probabilistic Search for Object Segmentation and Recognition . . . . .	791
<i>U. Hillenbrand, G. Hirzinger</i>	
Real-Time Interactive Path Extraction with On-the-Fly Adaptation of the External Forces . . . . .	807
<i>O. Gérard, T. Deschamps, M. Greff, L.D. Cohen</i>	
Matching and Embedding through Edit-Union of Trees . . . . .	822
<i>A. Torsello, E.R. Hancock</i>	
A Comparison of Search Strategies for Geometric Branch and Bound Algorithms . . . . .	837
<i>T. M. Breuel</i>	
Face Recognition from Long-Term Observations . . . . .	851
<i>G. Shakhnarovich, J.W. Fisher, T. Darrell</i>	

**Stereoscopic Vision II**

Helmholtz Stereopsis: Exploiting Reciprocity for  
Surface Reconstruction . . . . . 869  
    *T. Zickler, P.N. Belhumeur, D.J. Kriegman*

Minimal Surfaces for Stereo . . . . . 885  
    *C. Buehler, S.J. Gortler, M.F. Cohen, L. McMillan*

Finding the Largest Unambiguous Component of Stereo Matching . . . . . 900  
    *R. Šára*

**Author Index . . . . . 915**

## Table of Contents, Part IV

### Object Recognition / Vision Systems Engineering and Evaluation

Face Identification by Fitting a 3D Morphable Model Using Linear Shape and Texture Error Functions . . . . .	3
<i>S. Romdhani, V. Blanz, T. Vetter</i>	
Hausdorff Kernel for 3D Object Acquisition and Detection . . . . .	20
<i>A. Barla, F. Odone, A. Verri</i>	
Evaluating Image Segmentation Algorithms Using the Pareto Front . . . . .	34
<i>M. Everingham, H. Muller, B. Thomas</i>	
On Performance Characterization and Optimization for Image Retrieval . . . . .	49
<i>J. Vogel, B. Schiele</i>	

### Statistical Learning

Statistical Learning of Multi-view Face Detection . . . . .	67
<i>S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum</i>	
Dynamic Trees: Learning to Model Outdoor Scenes . . . . .	82
<i>N.J. Adams, C.K.I. Williams</i>	
Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary . . . . .	97
<i>P. Duygulu, K. Barnard, J.F.G. de Freitas, D.A. Forsyth</i>	
Learning a Sparse Representation for Object Detection . . . . .	113
<i>S. Agarwal, D. Roth</i>	

### Calibration / Active and Real-Time and Robot Vision / Image and Video Indexing / Medical Image Understanding / Vision Systems / Engineering and Evaluations / Statistical Learning

Stratified Self Calibration from Screw-Transform Manifolds . . . . .	131
<i>R. Manning, C. Dyer</i>	
Self-Organization of Randomly Placed Sensors . . . . .	146
<i>R.B. Fisher</i>	
Camera Calibration with One-Dimensional Objects . . . . .	161
<i>Z. Zhang</i>	
Automatic Camera Calibration from a Single Manhattan Image . . . . .	175
<i>J. Deutscher, M. Isard, J. MacCormick</i>	

What Can Be Known about the Radiometric Response from Images? . . . . .	189
<i>M.D. Grossberg, S.K. Nayar</i>	
Estimation of Illuminant Direction and Intensity of Multiple Light Sources . . . . .	206
<i>W. Zhou, C. Kambhamettu</i>	
3D Modelling Using Geometric Constraints: A Parallelepiped Based Approach . . . . .	221
<i>M. Wilczkowiak, E. Boyer, P. Sturm</i>	
Geometric Properties of Central Catadioptric Line Images . . . . .	237
<i>J.P. Barreto, H. Araujo</i>	
Another Way of Looking at Plane-Based Calibration: The Centre Circle Constraint . . . . .	252
<i>P. Gurdjos, A. Crouzil, R. Payrissat</i>	
Active Surface Reconstruction Using the Gradient Strategy . . . . .	267
<i>M. Mitran, F.P. Ferrie</i>	
Linear Pose Estimation from Points or Lines . . . . .	282
<i>A. Ansar, K. Daniilidis</i>	
A Video-Based Drowning Detection System . . . . .	297
<i>A.H. Kam, W. Lu, W.-Y. Yau</i>	
Visual Data Fusion for Objects Localization by Active Vision . . . . .	312
<i>G. Flandin, F. Chaumette</i>	
Towards Real-Time Cue Integration by Using Partial Results . . . . .	327
<i>D. DeCarlo</i>	
Tracking and Object Classification for Automated Surveillance . . . . .	343
<i>O. Javed, M. Shah</i>	
Very Fast Template Matching . . . . .	358
<i>H. Schweitzer, J.W. Bell, F. Wu</i>	
Fusion of Multiple Tracking Algorithms for Robust People Tracking . . . . .	373
<i>N.T. Siebel, S. Maybank</i>	
Video Summaries through Mosaic-Based Shot and Scene Clustering . . . . .	388
<i>A. Aner, J.R. Kender</i>	
Optimization Algorithms for the Selection of Key Frame Sequences of Variable Length . . . . .	403
<i>T. Liu, J.R. Kender</i>	

Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration . . . . .	418
<i>S. Granger, X. Pennec</i>	
An Unified Approach to Model-Based and Model-Free Visual Servoing . . . . .	433
<i>E. Malis</i>	
Comparing Intensity Transformations and Their Invariants in the Context of Color Pattern Recognition . . . . .	448
<i>F. Mindru, T. Moons, L. Van Gool</i>	
A Probabilistic Framework for Spatio-Temporal Video Representation & Indexing . . . . .	461
<i>H. Greenspan, J. Goldberger, A. Mayer</i>	
Video Compass . . . . .	476
<i>J. Kořecká and W. Zhang</i>	
Computing Content-Plots for Video . . . . .	491
<i>H. Schweitzer</i>	
Classification and Localisation of Diabetic-Related Eye Disease . . . . .	502
<i>A. Osareh, M. Mirmehdi, B. Thomas, R. Markham</i>	
Robust Active Shape Model Search . . . . .	517
<i>M. Rogers, J. Graham</i>	
A New Image Registration Technique with Free Boundary Constraints: Application to Mammography . . . . .	531
<i>F. Richard, L. Cohen</i>	
Registration Assisted Image Smoothing and Segmentation . . . . .	546
<i>B.C. Vemuri, Y. Chen, Z. Wang</i>	
An Accurate and Efficient Bayesian Method for Automatic Segmentation of Brain MRI . . . . .	560
<i>J.L. Marroquin, B.C. Vemuri, S. Botello, F. Calderon</i>	
A PDE Approach for Thickness, Correspondence, and Gridding of Annular Tissues . . . . .	575
<i>A. Yezzi, J.L. Prince</i>	
Statistical Characterization of Morphological Operator Sequences . . . . .	590
<i>X. Gao, V. Ramesh, T. Boulton</i>	
Image Registration for Foveated Omnidirectional Sensing . . . . .	606
<i>F. Dornaika, J. Elder</i>	
Automatic Model Selection by Modelling the Distribution of Residuals . . . . .	621
<i>T.F. Cootes, N. Thacker, C.J. Taylor</i>	

Assorted Pixels: Multi-sampled Imaging with Structural Models . . . . .	636
<i>S.K. Nayar, S.G. Narasimhan</i>	
Robust Parameterized Component Analysis: Theory and Applications to 2D Facial Modeling . . . . .	653
<i>F. De la Torre, M.J. Black</i>	
Learning Intrinsic Video Content Using Levenshtein Distance in Graph Partitioning . . . . .	670
<i>J. Ng, S. Gong</i>	
A Tale of Two Classifiers: SNoW vs. SVM in Visual Recognition . . . . .	685
<i>M.-H. Yang, D. Roth, N. Ahuja</i>	
Learning to Parse Pictures of People . . . . .	700
<i>R. Ronfard, C. Schmid, B. Triggs</i>	
Learning Montages of Transformed Latent Images as Representations of Objects That Change in Appearance . . . . .	715
<i>C. Pal, B.J. Frey, N. Jojic</i>	
Exemplar-Based Face Recognition from Video . . . . .	732
<i>V. Krüger, S. Zhou</i>	
Learning the Topology of Object Views . . . . .	747
<i>J. Wiegardt, R.P. Würtz, C. von der Malsburg</i>	
A Robust PCA Algorithm for Building Representations from Panoramic Images . . . . .	761
<i>D. Skočaj, H. Bischof, A. Leonardis</i>	
Adjustment Learning and Relevant Component Analysis . . . . .	776
<i>N. Shental, T. Hertz, D. Weinshall, M. Pavel</i>	
<b>Texture, Shading, and Colour</b>	
What Are Textons? . . . . .	793
<i>S.-C. Zhu, C.-e. Guo, Y. Wu, Y. Wang</i>	
Bidirectional Texture Contrast Function . . . . .	808
<i>S.C. Pont, J.J. Koenderink</i>	
Removing Shadows from Images . . . . .	823
<i>G.D. Finlayson, S.D. Hordley, M.S. Drew</i>	
<b>Author Index</b> . . . . .	837



# A Variational Approach to Recovering a Manifold from Sample Points

José Gomes and Aleksandra Mojsilovic

IBM Watson Research Center  
Route 134, Kitchawan Road  
Yorktown Heights, N.Y. 10598 , U.S.A  
{josegome,saska}@watson.ibm.com

**Abstract.** We present a novel algorithm for recovering a smooth manifold of unknown dimension and topology from a set of points known to belong to it. Numerous applications in computer vision can be naturally interpreted as instantiations of this fundamental problem. Recently, a non-iterative discrete approach, *tensor voting*, has been introduced to solve this problem and has been applied successfully to various applications. As an alternative, we propose a *variational formulation* of this problem in the *continuous* setting and derive an *iterative* algorithm which approximates its solutions. This method and tensor voting are somewhat the differential and integral form of one another. Although iterative methods are slower in general, the strength of the suggested method is that it can easily be applied when the ambient space is not Euclidean, which is important in many applications. The algorithm consists in solving a partial differential equation that performs a special anisotropic diffusion on an implicit representation of the known set of points. This results in connecting isolated neighbouring points. This approach is very simple, mathematically sound, robust and powerful since it handles in a homogeneous way manifolds of arbitrary dimension and topology, embedded in Euclidean or non-Euclidean spaces, with or without border. We shall present this approach and demonstrate both its benefits and shortcomings in two different contexts: (i) data visual analysis, (ii) skin detection in color images.

## 1 Introduction

In this section, we state the considered problem. Then, we give an overview of the corresponding state of the art. Finally, we present the organization of the paper.

### 1.1 Statement of the Problem

Consider a set  $\mathcal{P}$  containing  $N$  points of a  $n$ -dimensional manifold  $\Omega$ . In order to illustrate the ideas, we may think of  $\mathcal{P}$  as a data set of  $N$  measures performed on a stochastic (or deterministic) process whose state may be partially (or totally) described using  $n$  numeric values. Consequently, each point in  $\mathcal{P}$  is a sample, each coordinate is a parameter of the process under consideration, and  $\Omega$  is the set of possible values these variables may take *a priori*. The paradigm of experimental disciplines is that each individual sample captures only a partial information about the state of the process and one hopes, by considering multiple samples, to apprehend it in a more comprehensive way. Usually, this

is motivated by the expectation that the variables, *i.e.* the coordinates of the points, may not be independent from one another and that the relationships between the variables capture the "essence" of the measured process.

In this paper, we propose a theoretical and practical framework for analysing such data sets when the measured variables are expected to have complex distributions, strong relationships and more specifically in the limit case of functionally related variables. Formally, this restriction is equivalent to saying that the points of  $\mathcal{P}$  are not distributed arbitrarily in  $\Omega$  but, instead, up to some noise, belong to a submanifold of  $\Omega$  of dimension smaller than  $n$ . Analysing the data set is then equivalent to recovering this submanifold. Let us make this idea clearer through the two following examples.

## 1.2 Examples

**Stereo vision.** is perhaps the most straightforward example in computer vision. Pixel matching algorithms often output a cloud of points in  $\mathbb{R}^3$  supposed to belong to the surfaces of the pictured objects. It remains then to find a surface "passing" through this points. Here,  $\Omega = \mathbb{R}^3$ ,  $\mathcal{P}$  is the set of points and the searched surface is  $\mathcal{M}$ .

**Image manifolds.** arise when considering a set of image features, like spectral, color or texture measures, for classification purposes. If  $n$  such numeric features can be extracted for each image then one may well associate a set of  $N$  images and a cloud of  $N$  points in  $\mathbb{R}^n$ , each point representing one image through its features vector. If complex relationships exist between different features within a certain class of images, one may ideally expect the corresponding points to form a submanifold of  $\mathbb{R}^n$ . A geometric representation of this submanifold may be used to distinguish this class of images from all the other image points. Here,  $\Omega = \mathbb{R}^n$ ,  $\mathcal{P}$  is the set of features vectors and  $\mathcal{M}$  is the submanifold of  $\mathbb{R}^n$ , hopefully of low dimension, modeling the relations between the features.

## 1.3 State of the Art

In this section, we present an overview of the state of the art on recovering inter-variables relationships and the reader will soon see that our motivation is actually an old one. We survey general statistical methods and also algorithms more specific to computer vision.

The case of variables that are well represented by their mean and variance is covered extensively in statistics. Under some normality and independence assumptions, the issue of deciding whether or not such relations exist is addressed by the *hypothesis testing* methodology [17,14]. In this paper, we are interested exclusively in data sets with more complex distributions where such assumptions are not relevant.

In certain situations, this can be addressed by using the statistical *non-parametric methods* because they do not rely on the normality assumption. There are several such methods for estimating a non-parametric correlation between two or more variables. Well known such methods are *Spearman  $r$* ,

*Kendall  $\tau$*  and *coefficient  $\Gamma$*  (cf. [14]). Although these tests vary in their interpretation, they are all appropriate only when  $N$  is small because of the central limit theorem. Also, they are only interesting when the variables under consideration are well represented by their mean and variance. If higher order statistics are necessary, an alternative is to use *Pearson curves* [10] or *Johnson curves* [13]. These are families of simple distributions that can approximate more complex distributions up to their fourth moment. The original Pearson curves have been defined as the solutions to a differential equation and this is closely related to the method we are going to present. Though, neither Pearson curves nor Johnson curves are appropriate when both the distributions and the relations are expected to be even more complex.

Another traditional approach, exploratory data analysis, consists in identifying relations between several variables by searching systematic patterns. Different methods have been proposed. Most of those assume one of the following. There are certain particular values of the variables that represent well the data: cluster analysis. There exists special linear or polynomial combinations of the variables that yield simple relations: *principal component analysis*, *discriminant function analysis*, *multi-dimensional scaling*, *canonical correlation*, *step-wise linear* and *nonlinear regression*, *projection pursuit*. All these techniques are nicely presented in this textbook [16]. When no distributional assumption is available, *neural networks* are often used for their flexibility, although they relate very closely to standard statistical regressions [5]. Actually, they are often equivalent and, even if it is not always explicated, rely on the same assumptions. In addition, although *neural networks* are very powerful, their actual process is difficult to interpret and this may be a drawback in computer vision where geometric considerations are often crucial.

It is also important to mention a special branch of exploratory data analysis, visual data analysis, which relies upon the visualization of data sets and the ability of humans to detect relevant patterns. The most popular technique is brushing [3,24]. This is an interactive method which allows a user to select, *i.e.* brush with color, subsets of the data displayed with a certain representation (say, a scatter plot) and observe simultaneously the corresponding recolored subset in another one (say, a histogram). In this technique, the user is also allowed to manually fit models (curves, surfaces) to the observed distributions. This powerful technique can be further enhanced by the use of complementary data representations as well as animation. The weakness of brushing is that it is not automatic, not quantitative and not objective. Though, it is often the solution of last resort.

In computer vision, the very same “fitting” problem is particularly recurrent because typical detection algorithms output clouds of points that are then subject to high level processing.

Most of the methods in computer vision focus on reconstructing shapes in  $\mathbb{R}^2$  or  $\mathbb{R}^3$  and are motivated by geometric interpretations. As in the case of *neural networks*, they are often mathematically equivalent to known statistical methods but provide an effective way to achieve the result when *statistics* may only provide a mathematical interpretation of the underlying regression and optimality prior conditions. Three methodologies can be identified. The first one consist in using graph theory. For instance, one can consider  $\mathcal{P}$  as a set of graph

vertices [15,6] or use normalized cuts [27] for partitioning the ambient space into regions. Another interesting approach, *alpha shapes* [8] defines meaningful subsets of the Delaunay triangulation of  $\mathcal{P}$  by balancing convexity and resolution scale measures. The second methodology consists in gathering local perceptual or regularity cues for grouping [34,1], like good continuation [7], constant curvature [22] or local measures of confidence [26,31,25,12,9]. The third methodology also results in such local operations but is primarily motivated by mechanical or physical metaphors like in the *dynamic particles* [29]. They often use a variational formulation [18,23] and explicit or implicit geometric models [30,33].

However, there is also a more recent interest for methods meant to work in arbitrary dimension. For instance, *tensor voting* [21] has been introduced as a unified formalism for addressing the issues of grouping noisy sets of points into more regular features. It is founded on *tensor calculus*: the multi-dimensional data is first encoded into a tensor, then elements vote in their neighborhood through a convolution-like operation which is most appropriate in Euclidean ambient spaces. This results in a dense tensor map, containing both orientation and confidence informations, from which curves or surfaces can be extracted by a  $n$ -linear interpolation. This algorithm is not iterative because the smoothness is imposed by convolution. Note that *tensor voting* was inspired from *vector voting* [12].

We suggest another formalism, founded on *variational calculus*, that is interestingly related to almost all the previous methods. The remainder of the paper is as follows. In Section 2, we present the theoretical and practical aspects of the method. In Section 3, we discuss the benefits of this new approach in different applications. We conclude in Section 4.

## 2 Theory and Implementation

As a preliminary, we provide the reader with some informal intuition about the various ingredients of the method. Then, we propose a mathematical formalization of it and discuss its implementation.

### 2.1 Some Intuition

Suppose that  $\mathcal{P}$  is a sparse cloud of points in the plane ( $\Omega = \mathbb{R}^2$ ) and that these points are roughly distributed along a smooth curve. The problem consists in recovering “the” smooth curve  $\mathcal{M}$  that “passes through”  $\mathcal{P}$ . We simply develop the idea that  $\mathcal{P}$ , seen as a subset of  $\Omega$ , can be transformed continuously into the curve  $\mathcal{M}$ . This is achieved through an iterative process in which each point of  $\mathcal{P}$  spreads itself in the direction of its neighbouring points. Little by little, each point in  $\mathcal{P}$  transforms itself into a short piece of curve oriented toward other points of  $\mathcal{P}$ , and grows toward them. Eventually, all those pieces connect to one another so that the final curve is smooth and simply connected. At the same time, outliers are eliminated and the shape is regularized. This “spreading” process transforms continuously  $\mathcal{P}$  into the smooth manifold  $\mathcal{M}$ .

In the next section, we design an objective cost, or energy, associated with this spreading shape at each time instant of its deformation. The iterative minimization of this energy results in the “right” transformation because the energy is designed for that very purpose. Like in any variational method, the

final shape of  $\mathcal{M}$  will correspond to a minimum of the energy. Now, there are obviously two main issues here. The first one is “How to define a proper energy?” and the second one is “How to represent the evolving shape?”.

As far as the energy is concerned, it has to reflect the relevant properties of the desired final shape. In variational methods, the energy is often a weighted sum of several energies. Each term contributes to favor or penalize a certain shape property. The contributions are often contradictory and they compete by summation, like kinetic and potential energy do in mechanics. Hopefully, the achieved minimum of the total energy yields a satisfactory balance between each effect. In our problem, we need at least two energy terms. There should definitely be a *data attachment term*. In effect, shapes which contain a lot of points not belonging to  $\mathcal{P}$  should be penalized. This prevents the spreading process to add too many points and make  $\mathcal{M}$  too “fat”. Symmetrically, shapes which do not contain all the points of  $\mathcal{P}$  should be penalized as well. This is because the final result should not miss parts of  $\mathcal{P}$ . But this data attachment alone is useless because its minimization results in nothing but  $\mathcal{M} = \mathcal{P}$  and that is why it is called the data attachment term. We definitely need a *regularization term* as well. This one should favor better connected and smoother shapes. It will of course be in competition with the data attachment term because  $\mathcal{P}$  is not smooth and is not well connected at all. We do not discuss smoothness further for now since the issue is so well known. A way to obtain well connected shapes is to favor convexity. In effect, it is well known that convex shapes are simply connected, *i.e.* contain only one connected component. So, the regularization term favors convexity. Once again, this term alone is useless because minimizing it would really connect all the points of  $\mathcal{P}$ . The final result would then be the convex hull of  $\mathcal{P}$  and this is not desirable in general. To summarize this paragraph, the total energy will only favor spreading toward other neighbouring points of  $\mathcal{P}$  because this is the only way the two terms may actually reach a satisfying agreement.

The representation of the evolving shape is an important issue as well. In effect, *a priori*, no hypothesis is made neither about the dimension of the final shape  $\mathcal{M}$  nor its topology. This is a domain where implicit representations are usually superior to shape explicit parametrizations. In solving this issue, we were mostly inspired by [2,20] where a curve in  $\mathbb{R}^3$  is represented by a one-parameter family of concentric tubes of increasing radii. The represented curve is the medial axis of the tubes. If the radius  $r$  of the tubes is the parameter of the family then the tubes converge toward the curve when  $r$  tends to 0. This is actually a very general approach which is valid regardless of the dimension and topology. The key is to consider neighborhoods (or approximations) of the represented object with increasing tolerance. Note that those neighborhoods are always hyper-surfaces of the ambient spaces, *i.e.* manifolds of dimension  $n-1$ . For instance, in  $\mathbb{R}^3$ , both concentric tubes and concentric spheres are bi-dimensional although their limits are curves and points. Finally, those hypersurfaces are conveniently encoded as the iso-hypersurfaces of a scalar function defined on  $\Omega$ . This implicit representation makes it not too hard to formulate the problem as a variational one.

## 2.2 Formalization

Let  $u : \Omega \rightarrow [0, 1]$  be a smooth function to be constructed so that the family of hypersurfaces  $\mathcal{S}_\alpha = u^{-1}(\alpha)$ ,  $0 < \alpha < 1$ , tends to a submanifold  $\mathcal{M}$  of  $\Omega$  when  $\alpha$

tends to 0. In that sense,  $u$  can be interpreted as a *weak implicit representation* since

$$\mathcal{M} = \lim_{\alpha \rightarrow 0^+} u^{-1}(\alpha).$$

In particular, the value of  $u$  far from  $\mathcal{P}$  is 1 and tends to 0 as one approaches  $\mathcal{P}$ . We propose to define  $u$  as the solution to a variational problem that is naturally related to the reconstruction one. As a preliminary, we shall present some useful integral criteria, along with their Euler-Lagrange equations. Then, we shall see how to combine them and finally, since variational methods are iterative, we shall describe how to initialize  $u$ . We start with the case  $\Omega = \mathbb{R}^n$ .

**Ingredients.** For the moment, consider independently the four non-negative integrals

$$\int_{\mathcal{P}} u^2, \quad \int_{\Omega/\mathcal{P}} (u-1)^2, \quad \int_{\Omega} \nabla^2 u, \quad \text{and} \quad \int_{\Omega} \nabla u \cdot Q_u \nabla u,$$

where  $Q_u$  is the projector onto the sub tangent space of  $\Omega$  corresponding to negative eigenvalues of the Hessian of  $u$ . In other words, if  $D^2u = P^T D P$ , with  $D = \text{Diag}(\lambda_i)$  and  $P^T P = I$ , then  $Q_u = P^T G P$  with  $G = \text{Diag}(\nu(\lambda_i))$  where  $\nu(\mathbb{R}^-) = \{1\}$  and  $\nu(\mathbb{R}^{*+}) = \{0\}$ . Note that  $Q_u$  is symmetric positive.

The motivation for considering the first three integrals is obvious. The minimization of the first one enforces that  $\mathcal{M}$  passes through  $\mathcal{P}$ . In effect it is null if and only if  $u(\mathcal{P}) = \{0\}$ . Minimizing the second one prevents  $\mathcal{M}$  from passing through other points than those of  $\mathcal{P}$ . In effect, it is null if and only if  $u(\mathcal{M}/\mathcal{P}) = \{1\}$  a.e. in  $\Omega/\mathcal{P}$ . Finally, minimizing the third one enforces the smoothness of  $u$  hence, to some extent<sup>1</sup>, that of  $\mathcal{S}_\alpha$ . The Euler-Lagrange equations of these three first integrals are respectively

$$2u, \quad 2(u-1), \quad \text{and} \quad -2\Delta u$$

on the domains where the corresponding integrals are defined and  $\mathbf{0}$ , *i.e.* the null function, elsewhere in  $\Omega$ .

As for the fourth integral, it is minimized by non-concave functions but this is not as straightforward and we are going to develop this point further. Note that the motivation for considering convex  $u$ 's is that it implies the convexity of its iso-hypersurfaces  $\mathcal{S}_\alpha$  (*cf.* previous section) and this is how the connection between neighbouring points is favored. Here is how we minimize this integral. Although  $Q_u$  depends upon  $D^2u$ , we consider only the first order term in the Euler-Lagrange equation of the integral<sup>2</sup>. It is equal to

$$-2\Delta^- u,$$

<sup>1</sup> *i.e.* to the extent that  $\alpha$  is not a *singular* value of  $u$ .

<sup>2</sup> Observe that the dependence of  $Q_u$  with respect to  $D^2u$  can be thought of as having a contribution due to its eigenvalues  $\lambda_i$  and another one due to the isometry  $P$ . The first contribution is null a.e. because the derivative of  $\nu$  is null a.e.. As for the other dependence, we neglect it here because it introduces third order derivatives in the Euler-Lagrange equation and those are too difficult to evaluate numerically. We have to admit that it is a shortcoming of this method.

where  $\Delta^-u$  is the sum of the negative eigenvalues of the symmetric matrix  $D^2u$ , *i.e.*

$$\Delta^-u = \sum_{i=1, \dots, n} \nu(\lambda_i) \lambda_i \quad (1)$$

This result is obtained by writing

$$\frac{\partial (\nabla u \cdot Q_u \nabla u)}{\partial \nabla u} = (Q_u + Q_u^T) \nabla u = 2Q_u \nabla u$$

and then

$$\begin{aligned} -\nabla \cdot \frac{\partial (\nabla u \cdot Q_u \nabla u)}{\partial \nabla u} &= -2\nabla \cdot Q_u \nabla u \\ &= -2\nabla \cdot Q_u^2 \nabla u && Q_u \text{ is a projector} \\ &= -2Q_u \nabla \cdot Q_u \nabla u && Q_u \text{ is self-adjoint} \\ &= -2P^T G P \nabla \cdot P^T G P \nabla u && \text{by definition} \\ &= -2G P \nabla \cdot G P \nabla u && P^T \text{ is an isometry} \\ &= -2G P \nabla \cdot P \nabla u && G \text{ is a self-adjoint projector} \\ &= -2 \sum_{i=1, \dots, n} \nu(\lambda_i) \frac{\partial^2 u}{\partial p_i^2} \\ &= -2\Delta^-u, \end{aligned}$$

where the  $p_i$ 's are the eigenvector of  $D^2u$  and  $\Delta^-u$  is defined in Eq. 1.

The quantity  $\Delta^-u$  is the negative part of the Laplacian of  $u$  or “the negative Laplacian”. Obviously, it vanishes when  $u$  is a non-concave function. Hence again, this enforces the convexity of the hypersurfaces of  $u$  and consequently the convexity of  $\mathcal{M}$ . Of course, the final  $\mathcal{M}$  *will not* be globally convex because the four criteria are actually going to compete.

**Putting them all together.** Following, it is important to combine these integrals properly in order to achieve the right balance between each effect. We are going to form a weighted sum of those four integrals depending only upon one parameter, the scale  $\sigma$ . We define the scale as the critical distance between two *just distinguishable points* of  $\mathcal{P}$ . This concept from observation theory will serve as a yard stick to “calibrate” our linear combination of contributions.

First, remind that our integrals are not all defined in  $\Omega$ , so we use an indicator function of  $\mathcal{P}$ , *i.e.*  $\mathbb{I}_{\mathcal{P}}(\mathcal{P}) = \{1\}$  and  $\mathbb{I}_{\mathcal{P}}(\Omega/\mathcal{P}) = \{0\}$ . The two first integrals can then be rewritten

$$\int_{\Omega} \mathbb{I}_{\mathcal{P}}(\mathbf{x}) u(\mathbf{x})^2 d\mathbf{x} \quad \text{and} \quad \int_{\Omega} (1 - \mathbb{I}_{\mathcal{P}}(\mathbf{x})) (u(\mathbf{x}) - 1)^2 d\mathbf{x}.$$

Now, we form

$$E = c_1 \int_{\Omega} \mathbb{I}_{\mathcal{P}} u^2 + c_2 \int_{\Omega} (1 - \mathbb{I}_{\mathcal{P}}) (u - 1)^2 + c_3 \int_{\Omega} \nabla^2 u + \int_{\Omega} \nabla u \cdot Q_u \nabla u, \quad (2)$$

and the corresponding Euler-Lagrange equation (up to a factor  $\frac{1}{2}$ )

$$\partial_u E \equiv c_1 \mathbb{I}_{\mathcal{P}} u + c_2(1 - \mathbb{I}_{\mathcal{P}})(u - 1) - c_3 \Delta u - \Delta^- u, \quad (3)$$

where the three  $c_i$ 's are to be expressed in terms of  $\sigma$ .

For symmetry reasons, it is natural to choose  $c_1 = c_2$ . Furthermore, consider the case of two isolated just distinguishable points in  $\mathbb{R}$  located at coordinates 0 and  $\sigma$ . Setting  $c_3 = 0$ , because the smoothing must be negligible with respect to the other effects, the equilibrium condition writes  $c_2(u - 1) - u'' = 0$ , which is a linear differential equation of the second order. It can be integrated by quadrature, supposing that  $u(0) = u(\sigma) = 0$ . Then,  $c_2$  can be determined thanks to the additional equation  $u(\frac{1}{2}\sigma) = \frac{1}{2}$ : in effect, the fact that the two points are just distinguishable means in particular that their middle point can be assigned neither to  $\mathcal{M}$  nor to  $\Omega/\mathcal{M}$ .

Finally, we find

$$c_1 = c_2 = \left( \frac{\ln(7 + \sqrt{48})}{\sigma} \right)^2 \approx \frac{6.938}{\sigma^2} \quad \text{and} \quad c_3 = \varepsilon \ll 1,$$

hence the partial differential equation to be solved  $\frac{\partial u}{\partial t} = -\partial_u E$  or

$$\boxed{\frac{\partial u}{\partial t} = \frac{\beta^2}{\sigma^2} (-\mathbb{I}_{\mathcal{P}} u + (1 - \mathbb{I}_{\mathcal{P}})(1 - u)) + \varepsilon \Delta u + \Delta^- u,} \quad (4)$$

where  $\beta = \ln(7 + \sqrt{48})$  and  $\varepsilon \ll 1$ .

**Non-euclidean ambient spaces.** The case where  $\Omega$  is not Euclidean is important in practice. For instance, curve normal vectors live in  $\mathcal{S}^1$ , surface normal vectors in  $\mathcal{S}^2$ , line directions in  $\mathcal{P}^1$ , color hue in  $\mathcal{S}^1$  and  $k$ -uplets of such variables live in products of those spaces. Non-Euclidean variables occur quite often in *computer vision* and, in general, it is not accurate to consider them as taking values in Euclidean spaces. Fortunately, it is rather straightforward to generalize the presented method when  $\Omega$  can be embedded in a Euclidean space. In that case, one has just to rewrite the previous equations in the tangent plane of  $\Omega$  as introduced in [4]. Practically, one still solves a PDE using a regular grid and  $\Omega$  is represented implicitly by its distance function.

**Initialization of  $u$ .** One can simply initialize  $u$  as follows:  $u_0(\mathcal{P}) = \{0\}$  and  $u_0(\Omega/\mathcal{P}) = \{1\}$ . This has the advantage of being simple and fast but it does not account for repetitions in  $\mathcal{P}$  and, since the grid has integer coordinates, rounding effects are important.

Another way that is more robust to outliers and behaves better regarding redundant samples and rounding effects is to set:

$$u_0(\mathbf{x}) = \prod_{\mathbf{p} \in \mathcal{P}} \left( 1 - \epsilon e^{-\frac{(\mathbf{x}-\mathbf{p})^2}{\sigma^2}} \right) \quad (5)$$

Of course, this can be implemented efficiently by considering an approximation of the exponential having a compact support.



### 2.3 Implementation Issues

The function  $u$  is sampled over a regular grid of  $\Omega$  if it is Euclidean or over the Euclidean space of higher dimension in which  $\Omega$  has been embedded (*cf.* Section. 2) otherwise. The equation (4) is discretized using the standard explicit forward scheme for the time derivative (*i.e.*  $u_{t+dt} = u_t + (\dots)dt$ ) and the standard explicit centered schemes for the spacial derivatives (*i.e.* the Hessian of  $u$ ). The hypersurfaces  $\mathcal{S}_\alpha$  are extracted by  $n$ -linear interpolation. As far as the indicatrix function is concerned, it can be set to

$$\mathbb{I}_{\mathcal{P}}(\mathbf{x}) = e^{-\frac{\text{dist}^2(\mathbf{x}, \mathcal{P})}{\sigma^2}}$$

or any other reasonable approximation. Naturally, the finest possible scale,  $\sigma$ , is determined by the resolution of the grid. It is achieved by taking  $\sigma = 1$  (*i.e.* two grid nodes are two distinguishable points) but it can be set to a lower value if needed. The most natural way to work with it is to set  $\alpha = 1$  and choose an appropriate grid size. An approximation to the “negative Laplacian” may be computed as

$$\Delta^- u \approx \text{Trace}(N(D^2 u))$$

where  $N$  is an appropriate polynomial. In effect, we have  $\text{Trace}(N(D^2 u)) = \sum_{i=1, \dots, n} N(\lambda_i)$  because  $D^2 u$  is symmetric and thus,  $D^2 u$  and  $N(D^2 u)$  share the same eigenvectors. So, the polynomial  $N$  has just to “turn off” positive  $\lambda_i$ ’s for a reasonable range of values. This idea has been suggested to us by [30]. Of course,  $\Delta^- u$  may also be simply computed by diagonalizing  $D^2 u$ . These approximations of  $\Delta^- u$  are only valid if  $\Omega$  is Euclidean, however it is straightforward to extend them if  $\Omega$  is only embedded in a Euclidean space. Finally, in practice, one can really set  $\varepsilon = 0$  because the other spatial centered schemes are already diffusive enough for assuring a regular solution.

## 3 Applications

In this section, we discuss the benefits and limitations of the proposed technique in two different contexts: *multi-dimensional data analysis* and *skin detection in color images*.

### 3.1 Application to Multi-dimensional Data Analysis

A simple example will demonstrate how this technique can be helpful for the understanding of multi-dimensional data. When the number of variables is high, *i.e.* greater than 2 or 3, a traditional approach is to study slices or projections of the data with lower dimensions. Typically this approach yields some partial structural information. One limitation of analysing slices of the data is that one may fail to detect existing structural information. This is due in part to the fact that samples that where “almost” lying on the selected slice will be totally invisible. Fig. 2 depicts this situation on a real data set. Let us see how one may solve this with our method. Fig. 1 shows the iterative reconstruction of this data set using Eq.4 and Fig. 3 shows the resulting slice from which patterns can be detected more confidently. This is a rather trivial example since there are only three variables and a three-dimensional scatterplot yields the structural



**Fig. 1.** This figure shows the iterative resolution of Eq. 4. The curve and the surface are reconstructed to a certain extent.

information. However, if there are more than 3 variables, one may still perform exactly the same reconstruction in the whole dimensionality of the data set. This process cannot be easily visualized. But, once the reconstruction is done, standard analysis tools can be used to detect more reliably structural information in slices, scatterplots or other low-dimensional representations. The higher the number of variables, the more likely the reconstruction will yield information that would otherwise stay invisible. In practice, this technique would be most appropriate when the number of variables is smaller than 10 due to important memory requirements. This point is detailed in Section 4.

### 3.2 Application to Skin Detection in Color Images

Typical algorithms for detecting skin in color images proceed in two steps. First, a local skin color and/or skin texture detection is performed. Then, the geometry

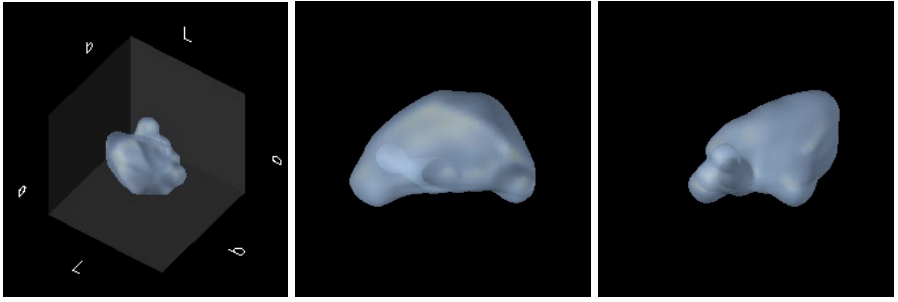


**Fig. 2.** On the left, a cloud of points in  $\mathbb{R}^3$  distributed along a one-dimensional curve and a connected bi-dimensional surface. On the right, the trace of the cloud of points on a selected slice. Although it is clear from the three-dimensional scatterplot that this trace is distributed along a curve, this is not visible on the slice alone.



**Fig. 3.** The cloud of points of Fig. 2 has been reconstructed, *cf.* also Fig. 1, and its trace over the same slice is shown. By connecting points in the full dimensionality of the data set, one makes it possible to detect structure more reliably. Of course, if the reconstruction is performed on the initial slice, *i.e.* in two dimensions, this information cannot be recovered hence the importance of reconstructing in the full dimensionality, especially when considering more than 2 or 3 variables.

of the detected regions is regularized using morphological operators. This section concerns only the first step and deals with the local detection of skin color using the approach described in this paper: 200 images depicting people have been segmented manually and the set  $\mathcal{P}$  is formed by the colors, in the *CIE Lab* system, of the millions of pixels corresponding to human skin.  $\Omega$  is the set of existing colors in the same color system and  $\mathcal{M}$  is supposed to approximate the set of colors corresponding only to human skin. After reconstructing  $\mathcal{M}$  (*cf.* Fig. 4), it is then possible to test whether the condition  $u(L, a, b) < u_0$  is satisfied, where  $(L, a, b)$  is the color of a tested pixel and  $u_0$  is a selected threshold related to the probability of a color to belong to  $\mathcal{M}$ . Consequently, this test may be used as a *local skin color detector*. Of course, it is important in this learning approach to eliminate the differences in lighting and camera responses within the learning set and we did this partially. To compensate for differences in lighting conditions, we have applied a simple model of the Von Kries adaptation [11]. The algorithm searches the image for likely representatives of white and black, and uses these values to compute a modified Von Kries adaptation. Although the spectrum of the light source cannot be completely recovered from the image, this model provides good results, as long as the spectrum of the light source is not too wildly skewed or irregular [28]. Fig. 5 shows a comparison of our detection method with two others on both images containing skin and images not containing skin. Of course, one of the strengths of the presented method is that it can be applied with more than three variables and this can be used in order to further enhance the segmentation. For instance, in Fig. 6, we added a fourth variable which is the variance of  $|(L, a, b) - (L_0, a_0, b_0)|$  in a small neighborhood of each pixel, where  $(L_0, a_0, b_0)$  is the color of the considered pixel and  $|\cdot|$  denotes the Euclidean distance. This measure, noted  $v(L, a, b)$  accounts somehow to the

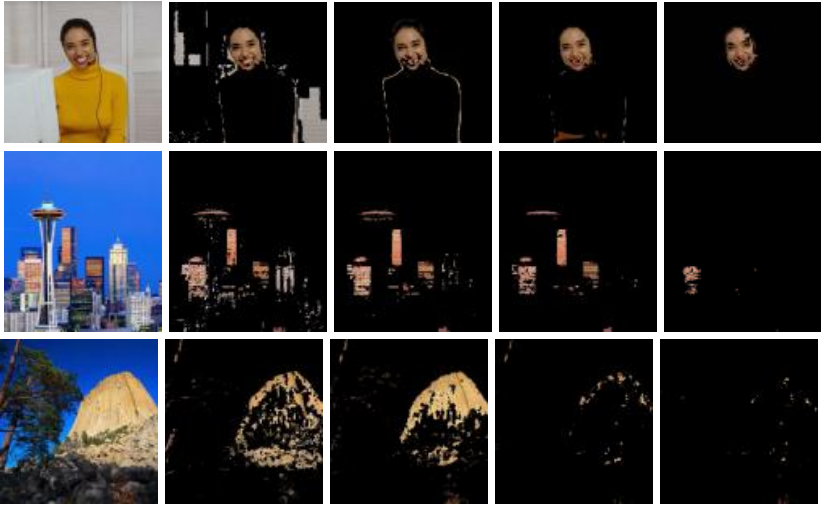


**Fig. 4.** This figure shows the learned shape of the set of human skin colors in the *CIE Lab* system from different standpoints.

presence of texture and this enhances the results in particular on images which do not contain human skin. It is also fair to say that the value of the threshold  $u_0$  is rather arbitrary and no physical interpretation has been attached to it, although it should not be difficult to relate it to a probability.



**Fig. 5.** Examples of results of skin color detection using (from the left to the right) the methods proposed in [11], in [28] and in this paper with the variables  $(L, a, b)$ .



**Fig. 6.** Examples of results of skin color detection using (from the left to the right) the methods proposed in [32], in [19] and in this paper with the variables  $(L, a, b)$  and, in last column, with the variables  $(L, a, b, v(L, a, b))$ . Notice how the additional variable, related to texture, enhances the results in images that do not contain skin.

## 4 Conclusion

In this paper, we have presented a new solution to the fundamental problem of recovering a manifold from a set of points known to belong to it. It is founded on variational calculus and results in a partial differential equation which can be interpreted explicitly as an anisotropic diffusion that “connects” neighbouring points. This is particularly interesting when dealing with non-Euclidean ambient spaces, where convolutions are not easy to implement. Although this technique is surprisingly simple, it gathers various good ideas from the three classes of existing methods surveyed in Section 1. In particular, the variational formulation of *geodesic snakes* [18], the fact that the initial solution is the set of points itself like in *dynamic particles* [29], the balance between convexity and scale like in *alpha shapes* [8], and finally the geometric implicit representation like in the level set methods [33]. All this makes it a very powerful tool in the most difficult situations as shown by our experiments. As for the limitations of the method, one has to be aware that, in practice, it is limited to dimensions smaller than about 10 because of the inherent computational complexity. If a narrow band of “voxels” is used, then the complexity is linear in the size of the reconstructed objects, both in time and memory. But, the involved constant may be very important due to the voxel-based representation. Another difficulty that has not been addressed is the choice of the metric to use when the variables are of different nature. However, the framework we have presented is certainly an appropriate one to introduce these considerations. One has also to be aware of the two layers of approximations introduced by the weak implicit representation (*cf.* Section 2).

We would like to end this paper with the following general notes. Although the method performs extremely well in practice, there are several theoretical issues which remain to be investigated. Like in any other variational method, the Euler-Lagrange equations provide only necessary conditions to the minimization of the energy and it should be clarified in which cases this is actually not sufficient. Further more, the link to standard statistical methods should be clarified as well. In particular, is the performed reconstruction equivalent to any known statistical regression ? Future work on this technique will focus on answering these questions.

## References

1. Narendra Ahuja and Mihran Tuceryan. Extraction of early perceptual structure in dot patterns. integrating region, boundary, and component Gestalt. *CVGIP*, 48(3):304–356, December 1989.
2. Luigi Ambrosio and Halil M. Soner. Level set approach to mean curvature flow in arbitrary codimension. *J. of Diff. Geom.*, 43:693–737, 1996.
3. A. Becker and W. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
4. Marcelo Bertalmio, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro. Variational problems and partial differential equations on implicit surfaces: The framework and examples in image processing and pattern formation. *UCLA Research Report*, June 2000.
5. C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.
6. J.D. Boissonnat. Representation of objects by triangulating points in 3-d space. In *Proceedings of ICPR*, pages 830–832, 1982.
7. J. Dolan and R. Weiss. Perceptual grouping of curved lines. In *Image Understanding Workshop*, pages 1135–1145, 1989.
8. H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.
9. P. Fua and P. Sander. Segmenting unstructured 3d points into surfaces. In *Proceedings of the European Conference on Computer Vision*, pages 676–680, Santa Margherita Ligure, Italy, 1992.
10. L. R. Lamberson G. F. Gruska, K. Mirkhani. *Non-Normal data Analysis*. Garden City, MI: Multifac Publishing., 1967.
11. Tziritas G. Garcia C. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, 1999.
12. G. Guy. Inference of multiple curves and surfaces from sparse data. Technical Report 96-345, USC IRIS, Ph.D. thesis, 1996.
13. G. Hahn and S. Shapiro. *Statistical Models in Engineering*. John Wiley & Sons, 1967.
14. S. Hays, W. Statistics, and Y. Holt. *Statistics*. Holt, Rinehart and Winston, New York, 1981.
15. Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.
16. StatSoft Inc. *Electronic Statistics Textbook*, volume <http://www.statsoftinc.com/textbook/stathome.html>. Tulsa, StatSoft, 2001.
17. S. Kachigan. *Statistical Analysis*. Radius Press, 1986.
18. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, London, June 1987.

19. J. M. G. Lammens. *A computational model of color perception and color naming (Ph.D. thesis)*. University of New York, Bufalo, 1994.
20. L. Lorigo, O. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, and C-F. Westin. Co-dimension 2 geodesic active contours for mra segmentation. In *Proceedings of the International Conference on Information Processing in Medical Imaging*, pages 126–139, June 1999.
21. Gérard Medioni, Mi-Suen Lee, and Chi-Keung Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.
22. P. Parent and S. Zucker. Trace inference, curvature consistency and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(8), 1989.
23. Tomaso Poggio and Federico Girosi. A theory of networks for approximation and learning. Technical Report 1140, AIM, 1989.
24. David. A. Rabenhorst. Interactive exploration of multidimensional data. In *Proceedings of the SPIE Symposium on Electronic Imaging*, volume 2179, pages 277–286, 1994.
25. S. Sarkar, K. Boyer, and I. inference. and management of spatial information using bayesian networks: perceptual organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3), 1993.
26. A. Shashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):90–94, 1988.
27. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1997.
28. K. Sobottka and I. Pitas. Extraction of facial regions and features using color and shape information. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages C421–C425, Vienna, Austria, 1996.
29. R. Szelisky, D. Tonnesen, and D. Terzopoulos. Modelling surfaces of arbitrary topology with dynamic particles. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 82–87, New York, June 1993.
30. Gabriel Taubin. A signal processing approach to fair surface design. *Computer Graphics*, 29(Annual Conference Series):351–358, 1995.
31. K.K. Thornber and R.L. Williams. Analytic solution of stochastic completion fields. In *Proceedings of SCV*, page 11B Segmentation and Grouping II, 1995.
32. G. Wyszecky and W. S. Stiles. *Color science: Concepts and methods, quantitative data and formulae*. John Wiley and Sons, New York, 1982.
33. H. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80(3):295–319, 2000.
34. S.W. Zucker and R.A. Hummel. Toward a low-level description of dot clusters: Labeling edge, interior, and noise points. In *Proceedings of CGIP*, volume 9, pages 213–233, 1979.

# A Variational Approach to Shape from Defocus<sup>\*</sup>

Hailin Jin and Paolo Favaro

Department of Electrical Engineering, Washington University, Saint Louis – MO 63130

Department of Computer Science, University of California, Los Angeles – CA 90095

{hljin,fava}@ee.wustl.edu

**Abstract.** We address the problem of estimating the three-dimensional shape and radiance of a surface in space from images obtained with different focal settings. We pose the problem as an infinite-dimensional optimization and seek for the global shape of the surface by numerically solving a partial differential equation (PDE). Our method has the advantage of being global (so that regularization can be imposed explicitly), efficient (we use level set methods to solve the PDE), and geometrically correct (we do not assume a shift-invariant imaging model, and therefore are not restricted to equifocal surfaces).

## 1 Introduction

Shape from defocus (SFD) consists of reconstructing the three-dimensional shape and radiance (“texture”) of a scene from a number of images taken with different focal settings. It is one of the classical problems in Computer Vision. This problem can be posed as the inversion of certain integral equations that describe the imaging process. Once an optimality criterion has been chosen, the problem can then be solved uniquely under suitable conditions on the radiance and the shape of the scene [13].

What makes SFD possible is the fact that the image of a scene at a certain position on the image plane (a “pixel”) depends upon the radiance on a region of the scene, as well as on the shape of such a region. What makes SFD possible, however, also makes it difficult: the image at a given pixel is obtained by integrating the (unknown) radiance of the scene against an (unknown) kernel that depends upon its shape. Given values of the integral at each pixel, one needs to estimate both the radiance and the kernel, which is known to be a severely ill-posed inverse problem in its full generality.

Several approaches have been presented to address this problem, which is an instance of “blind deblurring”, or “blind deconvolution” if one is willing to make the simplifying assumption of shift-invariant kernels, as we describe in Section 1.1. Typically, the depth of the scene is computed after approximating (locally) the radiance of the scene using various classes of functions or filters.

In this paper, rather than estimating depth at each pixel, we formulate shape from defocus within a variational framework as the problem of estimating an infinite-dimensional surface in space. We derive the optimality conditions and design a numerical algorithm to efficiently reach a (local) minimum. We do not make the assumption that the imaging kernel is shift-invariant – one that is patently violated at occluding boundaries – and therefore we can easily handle complex shapes.

---

<sup>\*</sup> This research is sponsored by ARO grant DAAD19-99-1-0139 and Intel grant 8029 to Stefano Soatto.



## 1.1 Relation to Previous Work

In the literature of computer vision, a number of algorithms have been proposed to estimate depth from defocus. The main assumption, common to most algorithms available in the literature, is that the scene is locally approximated by a plane parallel to the image plane [2,7,8,10,12,15,17,19,22,23,24,25,26]. This is called the *equifocal* assumption and it allows describing the imaging process as a linear convolution; the price to pay, however, is a fundamental trade-off between robustness and precision. In order to increase the reliability of the estimation, one would want to integrate over regions that are as large as possible; on the other hand, for the equifocal assumption to be valid, one would want regions to be as small as possible. In particular, at occluding boundaries the equifocal assumption is violated altogether.

Several algorithms have been also proposed to solve the problem in the shift-variant case as, for example, in [18]. [4] presents several methods for this purpose. The *block-variant blur* methods correct the assumption of local equifocal imaging by taking into account contributions from the neighboring regions. Other techniques are the complex spectrogram and the Wigner distributions, which are applied in a *space-frequency* framework. A successful approach employs a *Markov random field* model to obtain a *maximum a-posteriori* estimate of the blurring parameters.

## 1.2 Main Contributions

The equifocal assumption impacts both the shape reconstruction and radiance restoration.

Shape estimation is affected by how well an equifocal plane approximates (locally) the observed surface. This, in particular, implies that, within the surface, the best candidates will have equifocal tangent planes. It is clear that, unless the whole surface is close to be a plane parallel to the image plane, these candidates will be isolated points or curves. This implies that, in general, the estimation will be incorrect almost everywhere. Also, notice that this behavior does not depend on the smoothness of the surface. For example, if we consider a slanted plane, which is a smooth surface, any algorithm relying on the assumption above will result in a biased shape estimation.

Since the equifocal assumption at a point holds in general only locally, it is always associated with the choice of a domain around that point. Typically, such a domain is a square window. This is also the domain where the radiance is reconstructed. Notice, however, that when the chosen window is not in focus, it will receive contributions from the radiance lying on the neighboring regions, which are not accounted for in the imaging model. This implies that, regardless of the information carried by the texture of the radiance, the restoration of regions with higher intensity gradient (energy) will be favored over that of regions with lower intensity gradient.

These limitations motivate us to take a different approach. In this paper we forgo the equifocal assumption by approximating the scene with tangent planes, so that we can integrate visual information over the entire image. This results in superior resistance to noise (as also noticed in [4]). We formulate the problem within a variational framework, so that we can regularize the reconstruction process via imposing smoothness, and we do not make explicit approximations of the shape; rather, we estimate shape via

optimization on the infinite-dimensional space of smooth surfaces. We compute the necessary optimality conditions and numerically implement a partial differential equation to converge to a (local) minimum. Last, but not least, we achieve superior computational efficiency by estimating global shape, as opposed to depth at each pixel since the radiance on overlapping regions does not need to be recomputed.

## 2 Optimal Shape from Defocus

Let  $P$  be a generic point on the surface  $s$  in the scene with coordinates  $\mathbf{X} \in \mathbb{R}^3$ . Suppose that we are imaging the scene using a real aperture optical system with focal setting  $\eta$ . By exploiting the additive nature of the energy transport phenomena, we model the image formation process with an integral of the form

$$I_\eta(\mathbf{x}) = \int_s h_\eta(\mathbf{X}, \tilde{\mathbf{X}}) r(\tilde{\mathbf{X}}) dA(\tilde{\mathbf{X}}) \quad (1)$$

where  $\mathbf{x} = \pi_\eta(\mathbf{X}) \in \mathbb{R}^2$  is the projection of  $P$  on the image plane, which depends on the geometry of the optics, and in particular on the focal setting  $\eta$ , and  $dA(\tilde{\mathbf{X}})$  is the Euclidean area form of  $s$  at  $\tilde{\mathbf{X}}$ .  $r$  is the *radiance density* of the scene and  $h_\eta$  is the imaging kernel, which depends on the geometry of the imaging device.  $h_\eta$  satisfies the normalization constraint, i.e. for any surface  $s$

$$\int_s h_\eta(\mathbf{X}, \tilde{\mathbf{X}}) dA(\tilde{\mathbf{X}}) = 1. \quad (2)$$

We are able to measure the intensity  $I(\mathbf{x})$  at each point  $\mathbf{x}$ . Our goal is to reconstruct both the radiance  $r$  and the shape  $s$  of the scene from a collection of images obtained with different focal settings.

Suppose we have  $L$  images with different settings  $\eta_1, \dots, \eta_L$ . We collect and organize these images into an array  $I \doteq [I_{\eta_1}, \dots, I_{\eta_L}]^T$ , and do the same for the respective kernels  $h \doteq [h_{\eta_1}, \dots, h_{\eta_L}]^T$ . The right-hand side of equation (1) can also be interpreted as the *synthetic image* generated by a given surface  $s$  radiating energy with a given radiance density  $r$ . In this case we denote the collection of all such images with  $J \doteq [J_{\eta_1}, \dots, J_{\eta_L}]^T$ .

### 2.1 Cost Functional

Inverting the integral in equation (1) based on measurements  $I$  is an ill-posed problem. Furthermore, often (1) is only an approximation of the model that generates the data. We will therefore look for solutions that minimize a suitable *optimization criterion*. In [6] Csiszár presents a derivation of “sensible” optimization criteria for the problem above, and concludes that the only two that satisfy a set of consistency axioms are the  $L_2$ -norm – when the quantities at play are unconstrained – and the information-divergence – when both the radiance and the kernel are constrained to be non-negative. The latter criterion applies to our case since the radiance represents an energy density and the kernel

represents surface area. Therefore, without further discussion, we adopt the information-divergence (or I-divergence) as a cost functional for the discrepancy between measured images  $I$  and the synthetic images  $J$ :

$$\Psi(I|J) = \int_s \Phi(I(\mathbf{x})|J(\mathbf{x}))dA \quad (3)$$

where

$$\Phi(I(\mathbf{x})|J(\mathbf{x})) = I(\mathbf{x}) \log \frac{I(\mathbf{x})}{J(\mathbf{x})} - I(\mathbf{x}) + J(\mathbf{x}) \quad (4)$$

$\mathbf{x} = \pi(\mathbf{X})$ ,  $\mathbf{X}$  belongs to the shape  $s$ . Notice that our cost functional (3) is defined for the surface  $s$ , instead of the image domain as commonly seen in the literature. This allows us to derive a geometric flow (see Section 2.2) to minimize the cost functional with respect to the surface. To emphasize the dependency of  $J$  on the surface  $s$  and the radiance  $r$ , we write, with an abuse of notation,  $J = J(s, r)$ .

Hence, the problem of retrieving both shape  $s$  and radiance  $r$  from a collection of images  $I \doteq [I_{u_1}, \dots, I_{u_L}]^T$  can be formulated as that of finding a minimizer  $(\hat{s}, \hat{r})$  for the I-divergence between  $I$  and  $J = J(s, r)$ :

$$(\hat{s}, \hat{r}) = \arg \min_{(s, r)} \Psi(I|J(s, r)). \quad (5)$$

## 2.2 Radiance and Shape Estimation

The problem of minimizing the cost functional (3) involves solving a nonlinear optimization problem for two unknowns, which are both infinite-dimensional. To our best knowledge, there is no direct solution to minimize simultaneously both the shape  $s$  and radiance  $r$ , so we choose to divide the optimization into two sub-problems through an *alternating minimization* technique. Suppose we are given an initial guess for the radiance  $r_0$  and the surface  $s_0$  (see Section 3.4 for more details on initialization), then the algorithm can be written as:

$$\begin{cases} \hat{r}_{k+1} = \arg \min_r \Psi(I|J(\hat{s}_k, r)) \\ \hat{s}_{k+1} = \arg \min_s \Psi(I|J(s, \hat{r}_{k+1})). \end{cases} \quad (6)$$

The enabling step to use such an alternating minimization relies on having two iterations that independently lower the value of the cost functional, so that their combination leads toward the (local) minimum. For the first part we employ an iterative formula on the radiance  $r$ , which is constrained to be strictly positive, obtained from the Kuhn-Tucker conditions [11] on the cost functional. For the second part we use a gradient descent flow implemented using level set methods [16].

**Radiance Iteration.** Any radiance  $r$  that minimizes the cost function must satisfy the following necessary conditions:

$$\int_s \frac{h(\mathbf{X}, \tilde{\mathbf{X}})I(\pi(\mathbf{X}))}{\int_s h(\mathbf{X}, \tilde{\mathbf{X}})r(\tilde{\mathbf{X}})dA(\tilde{\mathbf{X}})}dA(\mathbf{X}) \begin{cases} = \int_s h(\mathbf{X}, \tilde{\mathbf{X}})dA(\mathbf{X}) & \forall \tilde{\mathbf{X}} : r(\tilde{\mathbf{X}}) > 0 \\ \leq \int_s h(\mathbf{X}, \tilde{\mathbf{X}})dA(\mathbf{X}) & \forall \tilde{\mathbf{X}} : r(\tilde{\mathbf{X}}) = 0 \end{cases} \quad (7)$$

These are the Kuhn-Tucker conditions, and since no closed-form solution is generally available, we seek for an iterative procedure such that the radiance will converge to a fixed point. Following Snyder et al. [21] we define the iteration as:

$$\hat{r}_{k+1}(\tilde{\mathbf{X}}) = \hat{r}_k(\tilde{\mathbf{X}}) \frac{1}{\int_s h(\mathbf{X}, \tilde{\mathbf{X}}) dA(\mathbf{X})} \int_s \frac{h(\mathbf{X}, \tilde{\mathbf{X}}) I(\pi(\mathbf{X}))}{h(\mathbf{X}, \tilde{\mathbf{X}}) \hat{r}_k(\tilde{\mathbf{X}}) dA(\tilde{\mathbf{X}})} dA(\mathbf{X}). \quad (8)$$

It can be shown that this iteration provably minimizes the chosen cost functional (with respect to  $r$ ) even when the iteration is computed without the correct shape  $s$  [10].

**Gradient Descent Flow** We minimize (3) with respect to the surface  $s$  by introducing an auxiliary time variable  $t$  and deforming the surface  $s(t) \doteq \{P(\mathbf{X}, t) : \mathbf{X} \in s\}$  in time  $t$  starting from an initial surface  $s(0)$ . The evolution is governed by a partial differential equation:

$$P_t = \mathcal{EL}(\mathbf{X}, s)N, \quad (9)$$

where  $N$  is the unit normal to the surface at  $P$  and  $\mathcal{EL}(\mathbf{X}, s) = 0$  is the Euler-Lagrange equation of the cost function (3). Note that in equation (9) we let the surface deform only along the normal direction, because the deforming in the tangent space will not change the shape of the surface (what it changes is the parameterization). This PDE is the *gradient descent flow*. When this flow converges to the steady state, i.e.  $P_t = 0$ , the Euler-Lagrange equation is satisfied. And it can also be shown that the cost functional monotonically decreases with respect to  $t$  (if a suitable initial shape  $P(\mathbf{X}, 0)$  is provided). Therefore, we are guaranteed to reach a (local) minimum of the cost function.

Away from discontinuities, we approximate the surface locally around  $P(\mathbf{X}, t)$  with the tangent plane  $T_P$ . Note that this approximation is fundamentally different from the equifocal assumption, because the tangent plane is not necessarily parallel to the focal plane. Hence the resulting kernel  $h$  is not shift-invariant. We assume that the radiance  $r$  is defined on a neighborhood around  $s$  (see Section 3.3 for more details on how to extend the radiance into  $\mathbb{R}^3$ ). A point  $\mathbf{Y}$  in  $T_P$  satisfies the identity  $N^T \mathbf{Y} = N^T \mathbf{X}$ , where  $N$  is the unit normal vector of  $s$  at  $P$ . Let  $K$  be the transformation bringing local coordinates  $(u, v)$  to points in  $T_P$ ,  $K : \Omega \subset \mathbb{R}^2 \rightarrow T_P$ .  $K$  can always be assumed to take the following form as long as the third component of  $N$  is non-zero:

$$\mathbf{Y} = K(\mathbf{X}, u, v) = \mathbf{X} + [u \ v \ k(u, v)]^T. \quad (10)$$

Immediately we have

$$k(u, v) = -\frac{N_1 u + N_2 v}{N_3}, \quad (11)$$

where  $N_1$ ,  $N_2$  and  $N_3$  are the components of  $N$ . Hence  $K$  explicitly depends on  $\mathbf{X}$  and  $N$ . Under these assumptions, the model image  $J$  can be computed as:

$$J(\pi(\mathbf{X})) = \int h(\mathbf{X}, \mathbf{X} + K(u, v)) r(\mathbf{X} + K(u, v)) du dv. \quad (12)$$

In [9], Faugeras and Keriven prove that the Euler-Lagrange equation for (3) takes the following form:

$$\mathcal{EL}(\mathbf{X}, s) = H\Phi - \Phi_{\mathbf{X}} \cdot N - H(\Phi_N \cdot N) + \text{Tr}((\Phi_{\mathbf{X}N})_{T_P} + dN \circ (\Phi_{NN})_{T_P}) = 0 \quad (13)$$

where  $Tr(\cdot)$  denotes the trace;  $H$  is the mean curvature, and  $\Phi_{\mathbf{X}}$  and  $\Phi_N$  stand for the derivatives of  $\Phi$  with respect to  $\mathbf{X}$  and  $N$  respectively.  $\Phi_{NN}$  and  $\Phi_{\mathbf{X}N}$  are the second order derivatives of  $\Phi$  and  $(\Phi_{NN})_{T_P}$  and  $(\Phi_{\mathbf{X}N})_{T_P}$  are their restrictions to the tangent plane  $T_P$ .  $dN$  is the differential of the Gauss map of the surface, which involves the second fundamental form. Writing the expressions explicitly, we have:

$$\begin{aligned}\Phi_{\mathbf{X}} &= \log\left(\frac{I}{J}\right) I_{\mathbf{X}} + \left(1 - \frac{I}{J}\right) J_{\mathbf{X}} \\ \Phi_N &= \left(1 - \frac{I}{J}\right) J_N \\ I_{\mathbf{X}} &= \nabla I \cdot \frac{\partial \pi(\mathbf{X})}{\partial \mathbf{X}} \\ J_{\mathbf{X}} &= \int (h_{\mathbf{X}} r + h \nabla r) dudv \\ J_N &= \int \frac{\partial K}{\partial N} (h_{\mathbf{X}} r + h \nabla r) dudv\end{aligned}$$

Note that we have skipped the arguments of all the functions for ease of notation. Similarly we can compute  $\Phi_{\mathbf{X}N}$  and  $\Phi_{NN}$ .

Finally, the gradient descent flow of  $P(\mathbf{X}, t)$  is:

$$P_t = (H\Phi - \Phi_{\mathbf{X}} \cdot N - H(\Phi_N \cdot N) + \text{Tr}((\Phi_{\mathbf{X}N})_{T_P} + dN \circ (\Phi_{NN})_{T_P})) N. \quad (14)$$

Since the flow (14) depends only on the first and second derivatives of  $\Phi$  with respect to  $\mathbf{X}$  and  $N$  and geometric quantities, namely,  $H$ ,  $N$ ,  $dN$ ,  $T_P$ , the flow is independent of any particular parameterization of the surface one chooses. In this sense, the flow is intrinsic. Experimentally, we find that the following first-order approximation of the flow (14) yields very similar results to those of the full second-order flow, while avoiding the time-consuming computation of  $\Phi_{\mathbf{X}N}$  and  $\Phi_{NN}$ :

$$P_t = (H\Phi - \Phi_{\mathbf{X}} \cdot N - H(\Phi_N \cdot N)) N. \quad (15)$$

### 3 Implementation

#### 3.1 Level Set Iteration

We implement the flow (14) and (15) using level set methods. The level set methods were originally developed by Osher and Sethian [16]. Since then, the methods have gained popularity in various fields. Many fast numerical schemes have been proposed based on it. For a complete account refer to [20]. The level set implementation of any geometric flow begins by embedding the initial interface  $P(\mathbf{X}, 0)$  as a level set of a scalar function  $\psi_0(\mathbf{X})$  which is then taken to be the initial condition for a function over time  $\psi(\mathbf{X}, t)$ :

$$\psi_0 : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad \psi : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}, \quad \psi(\mathbf{X}, 0) = \psi_0(\mathbf{X}).$$

The choice of a particular level set is arbitrary but is typically taken to be zero. The key point is that the interface is continuously embedded within the same fixed level set of  $\psi$

at all times. Thus, choosing the zero level set we have

$$\psi_0(P(\mathbf{X}, 0)) = 0, \quad \text{and} \quad \psi(P(\mathbf{X}, t), t) = 0.$$

Differentiating with respect to  $t$  therefore yields:

$$\psi_t + \nabla \psi \cdot P_t = 0 \tag{16}$$

an evolution equation for  $\psi$  (where  $\nabla \psi = \psi_{\mathbf{X}}$ ) which evolves the interface  $P(\mathbf{X}, t)$  described implicitly by  $\psi(\mathbf{X}, t) = 0$  for all  $t$ .

### 3.2 Intersection with the Surface

In the radiance iteration it is necessary to determine which point on the surface  $s$  corresponds to which point on the image plane, in order to establish the blurring radius of the kernel  $h$ . To be more specific, one needs to compute the intersection of a ray, which depends on the imaging model, with the surface  $s$ . Obtaining explicitly all the possible intersections with a discrete representation of the surface, for instance a triangulated mesh, turns out to be a computationally expensive task. Rather, it is possible to do this very efficiently by exploiting the advantage of an implicit formulation of the shape, i.e. the level set function  $\psi$  or the signed distance function. Let the ray be defined by a point  $X_0$  and a direction  $v$ . Let  $X$  be the intersection we are looking for.  $X$  satisfies the following (nonlinear) ordinary differential equation:

$$\begin{cases} \frac{dX}{dt} = c(X) \cdot v \\ X(0) = X_0 \end{cases} \tag{17}$$

where  $c(\cdot)$  is a scalar speed function defined as follows

$$c(X) = \begin{cases} \text{sign}(\psi(X)) & \text{if } |\psi(X)| > 1 \\ \psi(X) & \text{if } |\psi(X)| \leq 1 \end{cases}. \tag{18}$$

The rationale is that we move  $X$  according to  $c(\cdot)$  so that  $X$  is lead towards the surface. When  $X$  crosses the surface,  $c(X)$  will change sign accordingly, and therefore  $X$  will be forced to move in the opposite direction. Hence,  $X$  will oscillate around the intersection of the ray with the surface, reducing the overshoot at each step. Finally, we decide for  $X$  to be the intersection when the oscillation remains within a fixed band around the surface. This happens typically within a few iterations (3 to 5).

### 3.3 Radiance Extension

As mentioned in Section 2.2, we assume that the radiance is defined in a neighborhood around the surface  $s$ . Since the radiance is originally defined only on the surface (or the zero level set of  $\psi$ ), we need to find a way to extend it. One way to do so, which is widely used in the literature of applied mathematics [5,27], is to extend  $r$  such that it is constant along the normals of  $s$ . This means that the extension should satisfy

$$\nabla r \cdot \nabla \psi = 0. \tag{19}$$

To solve the above equation we numerically search for the steady state solution of the following PDE:

$$\frac{\partial r}{\partial t} + \text{sign}(\psi)(\nabla r \cdot \nabla \psi) = 0. \quad (20)$$

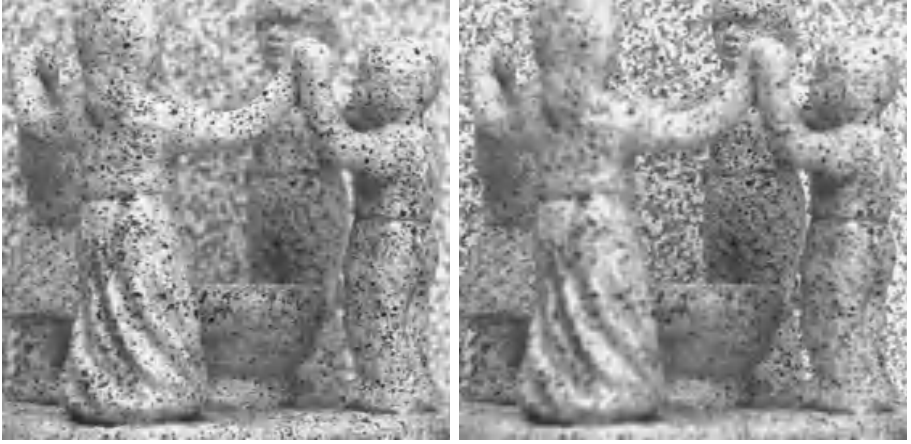
Note that this keeps  $r$  on the zero level set of  $\psi$  (the surface  $s$ ) unchanged. However, as a result of this process, the data is now defined in a neighborhood of  $s$ . The equation (20) can be efficiently solved using the fast marching technique [20].

### 3.4 Initialization

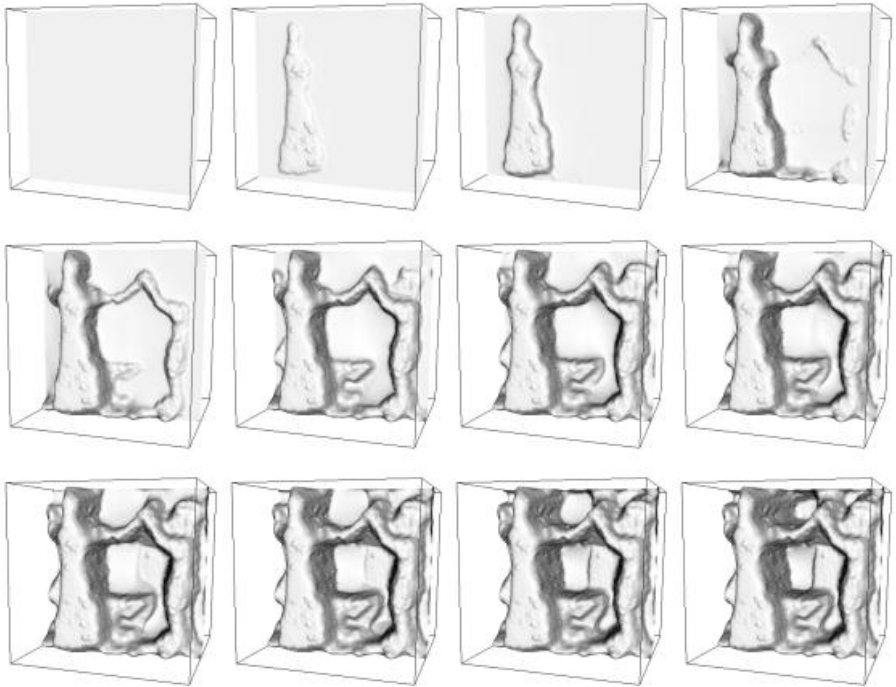
To start the alternating minimization one needs to have an initial guess for both radiance and surface. Since we have no prior knowledge on either unknown, we proceed as follows: choose one of the input images  $I_{u_1}$ , taken with focal setting  $u_1$ ; define the initial surface as a plane parallel to the focal plane passing through the focal depth  $u_1$ ; compute the radiance by back-projecting the image  $I_{u_1}$  onto the defined surface. As we see in our experiments, such a choice is not crucial to the estimation process. However, we also notice that a good initialization speeds up the minimization procedure considerably. Therefore, during the first steps of our algorithm we perform the surface estimation using a simple search of the minimum of the cost functional computed over a small grid of possible depths, assuming the surface is locally a plane. This initial surface is then used for the radiance iteration step after being smoothed. Later we substitute the search step with the level set iteration and proceed with the minimization as described in the previous sections.

## 4 Experiments

In this section we report some experiments with real images. Figure 1 shows two images which are obtained by changing the position of the image plane along the optical axis, and keeping the lens position fixed with respect to the scene. Moving the image plane necessarily involves scaling the images, which we avoid by employing a telecentric optical model (see [14]) and registering the two images using auxiliary patterns. Images are taken with an 8-bit camera containing two independently moving CCDs (kindly made available to us by S. K. Nayar). The near and far focused images in Figure 1 have focal depths of approximately  $0.9m$  and  $1.1m$  respectively. The focal length is  $35mm$  and the lens aperture is  $F/8$ . The scene has been chosen so as to test the performance of the proposed algorithm when the usual equifocal assumption does not hold. It can be noticed that the scene presents significant depth variations and several occluding boundaries. In particular, at the occluding boundaries of the statues and in the folds of the skirts, the planar approximation fails. Furthermore, the blurring radii are up to  $4 - 5$  pixels, so that the window size would have to be at least of 10 pixels, which would not allow for fine depth retrieval. In Figure 2 we show the corresponding surface evolution from the level set iteration. Figure 3 shows three steps of the radiance iteration during the alternating minimization procedure. Then, in Figure 4 we show the final estimate of the shape coded in gray level (256 values), where darker means closer to the viewer and brighter means farther from the viewer. Three views of the final shape which has been texture-mapped with the final radiance are also shown.



**Fig. 1. Original images:** the left image is near-focused (0.9m). The right image is far-focused (1.1m). As it can be noticed, in both images the blurring is quite large, the shape is non-trivial and presents several discontinuities due to occluding boundaries. The blurring radii for both images are about 3-4 pixels.

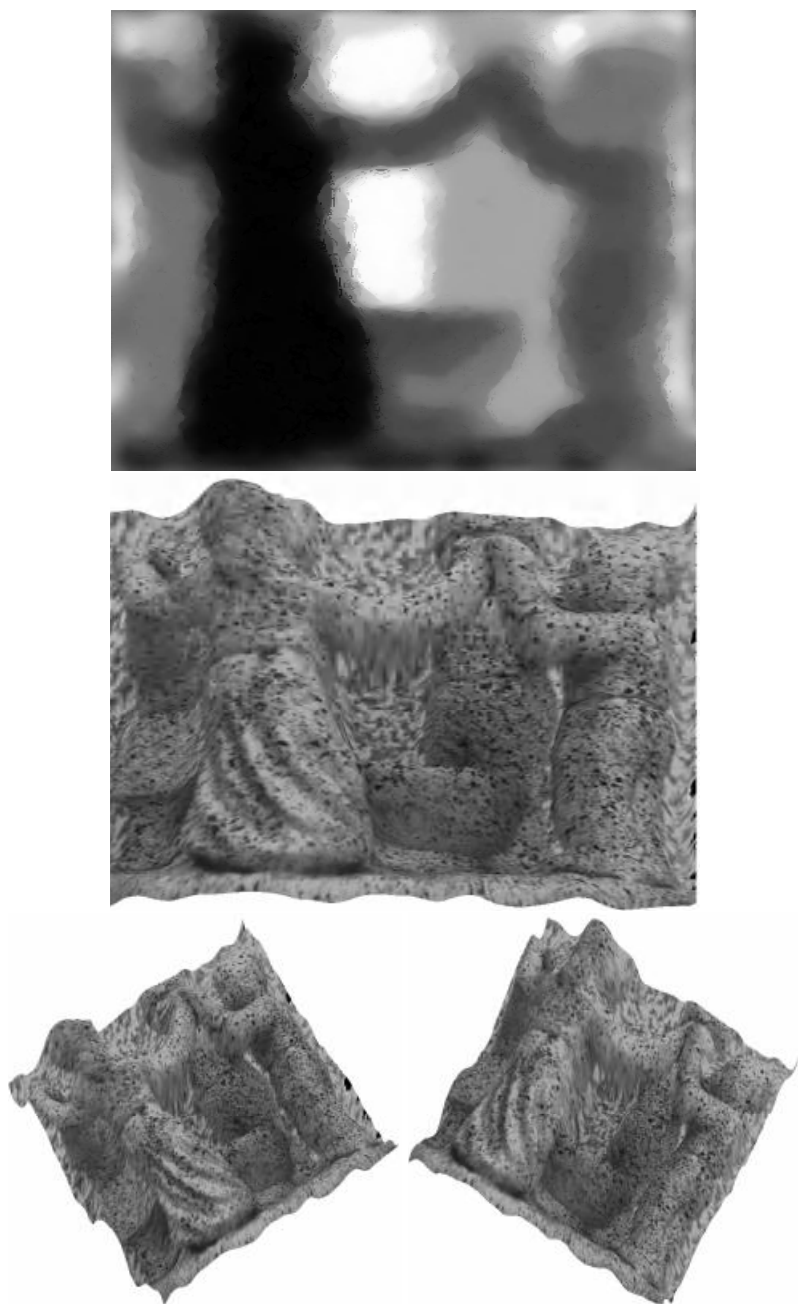


**Fig. 2. Shape evolution:** Twelve snapshots of the shape evolution: the surface is gradually converging to the final shape, starting from a plane placed at depth 0.9m.





**Fig. 3.** Three snapshots from the radiance iteration. **Top:** Initial radiance obtained from the near focused image; **Middle:** Radiance obtained after one iteration; **Bottom:** Radiance after three steps. It can be noticed that the radiance is gradually sharpening after each iteration, as is particularly visible in the background. The final radiance is sharp everywhere.



**Fig. 4.** *Top: depth rendered in gray levels (256 values) where darker means closer to the viewer and brighter means farther from the viewer; middle, bottom-left and bottom-right : three views of the final estimated shape, texture-mapped using the final estimated radiance.*

## 5 Conclusion

In estimating shape from defocus, the equifocal assumption is a well-known limitation. It introduces several disruptions in the reconstruction process such as image overlapping, windowing effects, edge bleeding, etc. We present a novel approach to shape from defocus based on an alternating minimization algorithm which does not make use of the equifocal assumption so as to overcome the above limitations. The radiance of the scene is estimated through an iterative scheme which provably converges to a minimum, while the shape is estimated using a gradient descent flow, which is then implemented numerically using level set methods. We show that the combination of these two steps leads to a (local) minimum of the discrepancy between the measured image and the modeled image. Also, by implementing the shape estimation with level set methods, we implicitly impose smoothness on the estimated shape in a completely automatic fashion.

**Acknowledgements.** The authors wish to thank Stefano Soatto and Anthony J. Yezzi for their helpful suggestions and invaluable support.

## References

1. Special issue on blind system identification and estimation. *Proceedings of the IEEE*, October 1998.
2. N. Asada, H. Fujiwara, and T. Matsuyama. Edge and depth from focus. *Intl. J. of Comp. Vision*, 26(2):153–163, 1998.
3. W. Boothby. *Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.
4. S. Chaudhuri and A. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.
5. S. Chen, B. Merriman, S. Osher and P. Smereka. A simple level set method for solving Stefan problems. *Journal of Computational Physics*, 135, 1995.
6. I. Csiszár. Why least-squares and maximum entropy; an axiomatic approach to inverse problems. *Annals of statistics*, 19:2033–2066, 1991.
7. T. Darell and K. Wohn. Depth from focus using a pyramid architecture. *Pattern Recognition Letters*, 11(2):787–796, 1990.
8. J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:97–108, 1993.
9. O. Faugeras and R. Keriven. Variational principles, surface evolution pdes, level set methods and the stereo problem. *INRIA Technical report*, 3021:1–37, 1996.
10. P. Favaro and S. Soatto. Shape and reflectance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, pages 755–768, June 2000.
11. D. Luenberger. *Optimization by vector space methods*. Wiley, 1968.
12. J. Marshall, C. Burbeck, and D. Ariely. Occlusion edge blur: a cue to relative visual depth. *Intl. J. Opt. Soc. Am. A*, 13:681–688, 1996.
13. A. Mennucci and S. Soatto. On observing shape from defocused images. In *Proc. of the Intl. Conf. on Image Analysis and Processing*, pages 550–555, 1999.
14. M. Watanabe and S. K. Nayar. Telecentric optics for constant magnification imaging. In *Technical Report CUCS-026-95*, pages Dept. of Computer Science, Columbia University, New York, USA, 1995.

15. S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.
16. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi equations. *J. of Comp. Physics*, 79:12–49, 1988.
17. A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:523–531, 1987.
18. A. N. Rajagopalan and S. Chaudhuri. A variational approach to recovering depth from defocused images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1158–64, October 1997.
19. Y. Schechner and N. Kiryati. The optimal axial interval in estimating depth from defocus. In *Proc. of the Intl. Conf. of Comp. Vision*, pages 843–848, 1993.
20. J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, 1996.
21. D. L. Snyder, T. Schulz, and J. A. O’Sullivan. Deblurring subject to nonnegativity constraints. *IEEE Trans. on Signal Processing*, 40(5):1143–1150, 1992.
22. S. Soatto and P. Favaro. A geometric approach to blind deconvolution with application to shape from defocus. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 10–17, June 2000.
23. M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *Intl. J. of Computer Vision*, 13:271–294, 1994.
24. M. Watanabe and S. K. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.
25. Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, pages 68–73, 1993.
26. D. Ziou. Passive depth from defocus using a spatial domain approach. In *Proc. of the Intl. Conf. of Computer Vision*, pages 799–804, 1998.
27. H. K. Zhao, T. Chen, and B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics* 127, pages 179–195, 1996.

# Shadow Graphs and Surface Reconstruction

Yizhou Yu and Johnny T. Chang

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, IL 61801, USA  
{yyz,jtchang}@uiuc.edu

**Abstract.** We present a method to solve shape-from-shadow using shadow graphs which give a new graph-based representation for shadow constraints. It can be shown that the shadow graph alone is enough to solve the shape-from-shadow problem from a dense set of images. Shadow graphs provide a simpler and more systematic approach to represent and integrate shadow constraints from multiple images. To recover shape from a sparse set of images, we propose a method for integrated shadow and shading constraints. Previous shape-from-shadow algorithms do not consider shading constraints while shape-from-shading usually assumes there is no shadow. Our method is based on collecting a set of images from a fixed viewpoint as a known light source changes its position. It first builds a shadow graph from shadow constraints from which an upper bound for each pixel can be derived if the height values of a small number of pixels are initialized properly. Finally, a constrained optimization procedure is designed to make the results from shape-from-shading consistent with the upper bounds derived from the shadow constraints. Our technique is demonstrated on both synthetic and real imagery.

**Keywords.** Surface Geometry, Shape-from-Shadow, Shadow Graph, Shading, Optimization

## 1 Introduction

In this paper, we consider the problem of shape-from-shadow and its integration with shape-from-shading. Shape-from-shadow tries to reconstruct a surface using multiple shadow images. It has a few advantages compared to other surface reconstruction techniques. For example, shadow constraints are insensitive to specular reflection and spatial variations of reflectance, and are able to impose long-range height constraints. The basic conclusion from previous work along this direction [20,13,9,4,14] says that with enough number of shadow images, the underlying surface can be recovered. However, the proposed algorithms for this problem are either complicated or heuristic. The major reason for this is that it was not clear how to effectively represent shadow constraints and integrate the information from multiple shadow images.

To clearly understand this problem, we introduce shadow graphs which can effectively represent and integrate shadow constraints from multiple images. We prove that the shadow graph alone is enough to solve the shape-from-shadow problem from a dense set of images. Simple operations on a shadow graph enable us to derive the structures of

the underlying surface. This approach is simpler and more systematic than the previous methods.

Usually most of the pixels in an image are not shadowed. However, shape-from-shadow neglects rest of the shading information in the input images. As we can see, shadow constraints are usually inequalities which are not as powerful as equalities. Consequently, it usually requires a dense set of input images to obtain good results. On the other hand, shape-from-shading [10] and photometric stereo [22] are effective approaches for a large class of surfaces including faces and sculptures. Both techniques use the pixelwise shading information to constrain surface normals, and do not allow shadows in the input images. They need an integration step to reconstruct a surface. This step tends to accumulate errors from pixel to pixel. Although theoretically they can uniquely recover the underlying surface, the final relative height values between distant points may not come out very accurately.

To take the advantages from both shape-from-shadow and shape-from-shading, we also develop a method of recovering shape from both shadow and shading constraints. A constrained optimization procedure is developed to make the results from shape-from-shading consistent with the upper bounds derived from shadow constraints.

## 1.1 Related Work

A few algorithms explicitly make use of shadow constraints [13,9,4,21]. Most of them belong to shape-from-shadow(darkness) algorithms. Some shape-from-shadow algorithms [13] use a shadowgram as an intermediate representation which is derived from a dense set of lighting directions. [9] assumes the underlying surface has a spline representation because shadows only provide a relatively sparse set of constraints. The number of unknown coefficients in the spline model is designed to scale with the number of shadow constraints. [4] introduces a shape-from-shadow algorithm using relaxation. A pair of upper-bound and lower-bound surfaces are constructed by updating the height values at pixels with violated shadow constraints. Like shape-from-shading, shape-from-shadow can recover unknown lighting directions as well [14].

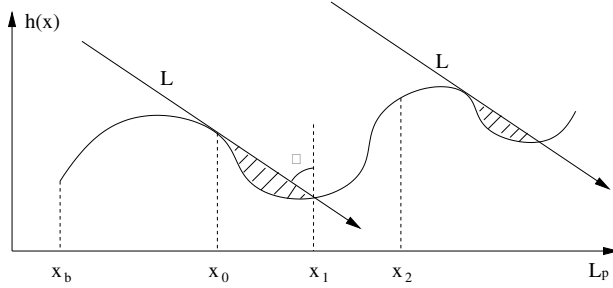
The computation of shape-from-shading has been typically characterized as that of finding surface orientation from one single image followed by a step that converts the orientation information into height under integrability constraints. The surface is usually assumed to be Lambertian. [15] introduces an algorithm that allows direct computation of height from shading. Since the unknowns directly represent pixelwise height values, this approach can be more naturally integrated with other methods of recovering shape, such as stereo and shape-from-shadow. [5] presents provably convergent algorithms for this problem.

Photometric stereo [22] can usually obtain better results than shape-from-shading because of the use of multiple input images. This approach has been generalized to recover shape for metallic and hybrid surfaces with both diffuse and specular reflection [11,17]. The lighting direction for each image is usually assumed to be known. However, both surface shape and lighting directions can be recovered simultaneously from SVD decomposition up to a bas-relief transformation [2,1]. Shadowed pixels in each image can be masked out in the process with the hope that there are still enough images covering them [12,7].

[21] considers recovery of shape from shading under a uniform hemispherical light source. Partial shadowing is taken into account because only a part of the light source is visible from every surface point. Interreflections are also considered in the algorithm presented in [18].

## 2 Shadow Graphs

We consider recovering terrain-like height fields in this paper. For the convenience of discrete representation based on pixels, a height field is assumed to be a piecewise constant function with every pixel corresponding to a piece with constant height. Every piece of the height field is represented by the point corresponding to the center of the pixel. We also assume that the distance between the camera and the surface is large enough so that the orthographic projection model is accurate. Let us first check what



**Fig. 1.** 2D schematic of shadowed and nonshadowed regions on a terrain-like surface.  $L$  is the parallel lighting direction.  $x_0$  is an occluder,  $x_1$  is on the shadow boundary caused by  $x_0$ , and  $x_2$  is a non-shadowed point.

kind of constraints are available from images with shadows. Let  $h(x)$  be a height field defined on a planar domain  $D$  with a finite area in the image plane and  $L$  be the lighting direction pointing downwards with a tilt angle  $\theta (< 90^\circ)$ . The normal orientation of this height field is denoted as  $\mathbf{n}(x)$ . The boundary curve of domain  $D$  is  $\Omega$ . The projected vector of  $L$  in the domain  $D$  is  $L_p$ . Let  $\mathbf{x}_i$  and  $\mathbf{x}_j$  be two arbitrary 2D points in  $D$ . The line segment between them is denoted as a vector interval  $[\mathbf{x}_i, \mathbf{x}_j]$  for convenience. Based on whether a point on the height field is in shadow or not under lighting direction  $L$ , there are two different sets of constraints (Fig. 1).

- If any point on the line segment  $[\mathbf{x}_0, \mathbf{x}_1]$  is in shadow, the points at  $\mathbf{x}_0$  and  $\mathbf{x}_1$  are the delimiting points of this shadow segment, and the point at  $\mathbf{x}_0$  is the occluding point generating this shadow segment, we have the following *shadow* constraints.

$$h(\mathbf{x}) \leq h(\mathbf{x}_0) - \frac{\|\mathbf{x} - \mathbf{x}_0\|_2}{\tan \theta}, \forall \mathbf{x} \in [\mathbf{x}_0, \mathbf{x}_1]; \quad (1)$$

$$h(\mathbf{x}_1) = h(\mathbf{x}_0) - \frac{\|\mathbf{x}_1 - \mathbf{x}_0\|_2}{\tan \theta}; \quad (2)$$

$$\mathbf{L} \cdot \mathbf{n}(\mathbf{x}_0) = 0 \quad (3)$$

- where the last equation means the lighting vector  $\mathbf{L}$  falls inside the tangential plane at  $\mathbf{x}_0$  if the original continuous height field is locally differentiable at  $\mathbf{x}_0$ .
- If the point at  $\mathbf{x}_2$  is not in shadow, we have the following *antishadow* constraints.

$$h(\mathbf{x}) \leq h(\mathbf{x}_2) + \frac{\|\mathbf{x} - \mathbf{x}_2\|_2}{\tan \theta}, \forall \mathbf{x} \in [\mathbf{x}_b, \mathbf{x}_2] \quad (4)$$

where  $\mathbf{x}_b \in \Omega$  and the line segment  $[\mathbf{x}_b, \mathbf{x}_2]$  is in the same direction as  $\mathbf{L}_p$ .

Let us first focus on how to represent the inequality constraints (1) and (4) in a graph.

**Definition 1.** A shadow graph is a weighted directed graph  $G = (V, E, W)$  where the set of nodes  $V$  is the set of points defined on domain  $D$ , an edge  $e = (\mathbf{x}_i, x_j) \in E$  indicates  $h(\mathbf{x}_j)$  is dependent on  $h(\mathbf{x}_i)$  and  $h(\mathbf{x}_i) - h(\mathbf{x}_j) \geq W(e)$  where the edge weight  $W(e)$  can be any real number.

A shadow graph can be induced from an image of the height field under an arbitrary lighting direction  $\mathbf{L}$ . Shadowed pixels can be detected from the image, and an occluder can be located for each continuous shadow segment with the knowledge of the lighting direction. For example, if  $[\mathbf{x}_i, x_j]$  is a shadow segment and the vector from  $\mathbf{x}_i$  to  $\mathbf{x}_j$  is in the direction of the projected lighting direction  $\mathbf{L}_p$ , the point at  $\mathbf{x}_i$  is the occluder of all the points in  $[\mathbf{x}_i, x_j]$ . There should be an edge  $(\mathbf{x}_i, x)$  with weight  $\frac{\|\mathbf{x} - \mathbf{x}_i\|_2}{\tan \theta}$  in the induced graph for all  $\mathbf{x} \in (\mathbf{x}_i, x_j]$ . This graph basically encodes the shadow constraints available from the image. All the edge weights in this graph should be positive. However, this graph can have negative weights if the additional antishadow constraints in Eq. (4) are represented as well.

Suppose we have multiple images of the height field under a set of lighting directions  $\{\mathbf{L}_k\}_{k=1}^m$ . Each of the images has its own shadow graph. Finally, the edges from all of these individual graphs can be accumulated into one graph that is corresponding to all the images. Note that this graph does not have the specific lighting information, which is not particularly important because all the constraints essential to the height field are kept there.

**Proposition 1.** A shadow graph with positive weights is a directed acyclic graph.

**Proof** Suppose there is a circular path in the graph and a node  $v$  is on the path. Since all the arcs on this path have positive weights, it is easy to conclude that  $h(v) > h(v)$  by starting from  $v$ , going through this path and back to  $v$ . A contradiction.

When dealing with real images with noise, shadow detection cannot be expected to be error free. Inaccurate shadow segmentations may result in cycles in the induced shadow graphs. Since cycles can lead to the above contradiction, we must convert a cyclic graph into an acyclic one by removing some of the edges in the graph. Since we would like to make the least amount of change to the graph, a sensible criterion for an optimal conversion is that the total accumulated weight for the removed edges should be minimized. However, graph conversion under this criterion is NP-hard [8]. To obtain



an efficient solution for this problem, we adopt the permutation-based approximation algorithm in [8] which tends to remove more edges than necessary. After applying this algorithm, for each of the removed edges, we still run a depth-first search to check whether the graph is still acyclic after the edge is inserted back into the graph. These two steps together lead to a polynomial time approximation that removes the least number of edges.

**Definition 2.** *The transitive closure of a shadow graph  $G$  is defined to be a new graph  $G^c = (V, E^c, W^c)$  on the same set of nodes such that  $(\mathbf{x}_i, x_j) \in E^c$  as long as there is a path from  $\mathbf{x}_i$  to  $\mathbf{x}_j$  in  $G$ , and  $W((\mathbf{x}_i, x_j))$  is set to be the maximum accumulated weight among the paths from  $\mathbf{x}_i$  to  $\mathbf{x}_j$ .*

There are a set of nodes  $V_t \subset V$  in  $G^c$  that do not have any incident edges with positive weights, which means they are not shadowed by any other points in any of the images. The highest point(s) of the height field surely belongs to this set because there is no other points which can occlude it(them) from the light sources. The absolute height values of the nodes in  $V_t$  are unrecoverable from shadow constraints. However, if we can recover their height values from other approaches such as stereo processing, the information embedded in  $G^c$  can be used for obtaining an upper bound of the height at any point in  $V - V_t$ . The set of edges in  $G^c$  connecting  $V_t$  and  $V - V_t$  becomes the most important for this purpose. Suppose there is a node  $v \in V - V_t$  and a set of associated edges  $E_v \subset E^c$  such that if an edge  $e = (v_t, v) \in E_v$ ,  $v_t \in V_t$ . The upper bound of the height at the point corresponding to node  $v$  can be obtained from

$$U(h(v)) = \min_{(v_t, v) \in E_v} (h(v_t) - W((v_t, v))). \quad (5)$$

Let us examine the asymptotic behavior of this upper bound when we increase the number of input images with lighting directions covering the whole lighting hemisphere. The set  $V_t$  will shrink and approach its limit which is the set of the highest points of the height field. Otherwise, assume there is a pair of nodes  $v_1, v_2 \in V_t$  and  $h(v_2) < h(v_1)$ . We can always design a lighting direction from which the point corresponding to  $v_1$  shadows the point corresponding to  $v_2$ , which means  $v_2 \notin V_t$ , a contradiction. Since eventually  $V_t$  only has nodes at the same height, we do not need to seek their relative height through other reconstruction techniques. Our interest should be focused on the relative height of other points compared to the highest points whose height can always be set to zero.

**Proposition 2.** *Eq. (5) gives an upper bound for the height at any node in  $V - V_t$  provided that the estimation of the height for the nodes in  $V_t$  is accurate. With an increasing number of input images with lighting directions covering the whole lighting hemisphere, Eq. (5) converges asymptotically to the correct relative height, with respect to the highest points in  $V_t$ , at any point in  $V - V_t$ .*

**Proof** The first part is obvious. The second part can be proved by induction. Since we only have a finite number of points according to our surface model, we can sort the points in decreasing order of their height. The highest points in the sorted list are assumed to be at height zero. Suppose the point at  $\mathbf{x}_m$  is the  $k$ -th element in the sorted list and the height

of its  $k - 1$  preceding elements can be recovered to an arbitrary precision independently of the height of the rest of the elements in the list. Now we show that the height of the point at  $\mathbf{x}_m$  can also be recovered to an arbitrary precision independently of the height of its following elements in the list. Note that all the surface points are lit if we have a vertical lighting direction. If we increase the tilt angle of the light, the point at  $\mathbf{x}_m$  will certainly be shadowed since it is not one of the highest points. Given a certain density of the lighting direction, there exist two adjacent directions  $\mathbf{L}_r$  and  $\mathbf{L}_l$  such that this point at  $\mathbf{x}_m$  is non-shadowed when the light is at  $\mathbf{L}_r$  and becomes shadowed when the light moves to  $\mathbf{L}_l$ . An upper bound for this point can be obtained from  $\mathbf{L}_l$  and an occluder at  $\mathbf{x}_o$  whose height is recovered to an arbitrary precision. When we increase the density of the lighting direction, the difference between  $\mathbf{L}_r$  and  $\mathbf{L}_l$  becomes arbitrarily small and the upper bound for the point at  $\mathbf{x}_m$  also becomes arbitrarily close to its true height.

Shadowgrams introduced in [13,4] also have the capability to recover correct surface geometry. But they are more complicated than shadow graphs because they explicitly keep lighting directions in the representation.

It is clear that the antishadow constraints can be derived from the shadow constraints if we have a very dense set of images since the height field itself can be recovered from the shadow constraints alone according to the above Proposition. However, if we only have a sparse set of images, this is not necessarily true. Representing these antishadow constraints in a shadow graph usually can provide additional information. According to Eq. (4), antishadow constraints transform to additional edges with negative weights. Cycles can appear in the resulting graph. However, the accumulated weight of any cycle can not be positive according to the following Proposition.

**Proposition 3.** *The accumulated weight of a circular path in a shadow graph must be either zero or negative.*

**Proof** Suppose  $\mathbf{x}_0, x_1, \dots, x_n \in V$  are consecutive nodes of a circular path, i.e.  $(\mathbf{x}_i, x_{i+1}) \in E (i = 0, \dots, n - 1)$  and  $(\mathbf{x}_n, x_0) \in E$ . From the definition of a shadow graph,  $h(\mathbf{x}_i) - h(x_{i+1}) \geq W((\mathbf{x}_i, x_{i+1})) (i = 0, \dots, n - 1)$  and  $h(\mathbf{x}_n) - h(x_0) \geq W((\mathbf{x}_n, x_0))$ . Therefore,

$$\begin{aligned} & \sum_{i=0}^{n-1} W((\mathbf{x}_i, x_{i+1})) + W((\mathbf{x}_n, x_0)) \\ & \leq \sum_{i=0}^{n-1} (h(\mathbf{x}_i) - h(x_{i+1})) + (h(\mathbf{x}_n) - h(x_0)) = 0. \end{aligned}$$

The transitive closure of a shadow graph  $G$  with cycles is still well-defined because negative cycles do not interfere with the objective to seek paths with maximum accumulated weights according to the definition. The resulting graph  $G^c$  can still be used for obtaining an upper bound of the height for any point in  $V - V_t$ . Since there may be negative edges pointing from nodes in  $V - V_t$  to some nodes in  $V_t$ , these edges can be

used for obtaining a lower bound for some nodes in  $V - V_t$ . Since it is not guaranteed that there is an edge from each node in  $V - V_t$  to some node in  $V_t$  given a sparse set of images, we can only obtain lower bounds for a subset of nodes in  $V - V_t$ . And these lower bounds may appear useful in combination with other surface reconstruction techniques.

### 3 Integrated Shadowing and Shading Constraints

Given a sparse set of images with known lighting directions, we would like to recover shape using both shadow and shading constraints. As we have seen, shadows impose explicit constraints over surface height values, but they are usually not sufficient if applied alone. On the other hand, shading information imposes constraints over normal orientation. We are going to explore two options for integrating shadow constraints with shading information.

#### 3.1 Enforcing Shadowing Constraints with Penalty Terms

Since shape-from-shading is not the focus of this paper, we adopt the direct height from shading algorithm in [15] as the base for solving shading constraints. Since this technique computes a height field directly rather than through surface normals, it is relatively easy to incorporate shadow constraints and enforce surface upper/lower bounds from the previous section. The shape-from-shading problem is formulated to minimize the following cost function in [15]:

$$E_1 = \sum_{i,j} [\alpha(\rho R(p_{ij}, q_{ij}) - I(i, j))^2 + \lambda(u_{ij}^2 + v_{ij}^2)] \quad (6)$$

where  $\rho$  is the surface albedo,  $I$  is the observed image intensity,  $p_{ij}, q_{ij}, u_{ij}, v_{ij}$  are the symmetric first and second finite differences of the surface height field  $\{h_{ij}\}$ ,  $\alpha$  and  $\lambda$  are two constant coefficients, and  $R$  is the Lambertian reflectance model. The first term in Eq. (6) corresponds to the photometric error term. And the second is a regularization term on the smoothness of the surface.

This formulation can be easily generalized to accommodate multiple input images and shadow masks as follows.

$$E_2 = \alpha \sum_k \sum_{i,j} l_{ij}^k (\rho_{ij} R^k(p_{ij}, q_{ij}) - I^k(i, j))^2 + \lambda \sum_{i,j} (u_{ij}^2 + v_{ij}^2) \quad (7)$$

where  $I^k(i, j)$  represents the  $k$ -th input image with corresponding reflectance map  $R^k$ ,  $l_{ij}^k$  is a binary shadow mask indicating whether pixel  $(i, j)$  in the  $k$ -th image is lit by the light source or not, and  $\rho_{ij}$  is the unknown pixelwise surface albedo. This treatment is similar to photometric stereo, but solves the height field directly instead. With multiple images, the regularization term becomes much less important, and can be set close to zero. However, it may still have some effects at pixels that are lit in less than three different images.

To further incorporate the constraints in Eq. (1) and (4) into the above formulation, we notice that the constraints have the same form which looks like

$$h_{ij} - h_{i'j'} \geq d_{ij i'j'}. \quad (8)$$

To enforce this kind of inequalities in a gradient-based minimization method, a differentiable half-sided parabola is adopted as a penalty function.

$$S(i, j, i', j') = \begin{cases} (h_{ij} - h_{i'j'} - d_{ij i'j'})^2, & \text{if } h_{ij} - h_{i'j'} \leq d_{ij i'j'}; \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

The penalty functions for all the inequalities and equalities can be inserted as additional terms into Eq. (7). The new cost function for surface reconstruction is given as follows.

$$E_3 = \alpha \sum_k \sum_{i,j} l_{ij}^k (\rho_{ij} R^k(p_{ij}, q_{ij}) - I^k(i, j))^2 + \lambda \sum_{i,j} (u_{ij}^2 + v_{ij}^2) + \beta \sum_k (\sum_{m_k} S^k(i_{m_k}, j_{m_k}, i'_{m_k}, j'_{m_k}) + T_k) \quad (10)$$

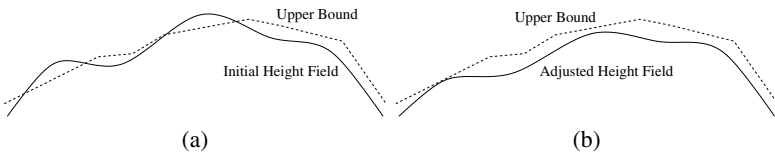
where  $m_k$  is the index of the inequality constraints from the  $k$ -th image,  $S^k(i_{m_k}, j_{m_k}, i'_{m_k}, j'_{m_k})$  represents the actual penalty terms contributed by the  $k$ -th image, and  $T_k$  represents the collection of penalty terms for the equality constraints associated with shadows, such as those in Eq. (2) and Eq. (3). In our experiments, we use iterative minimization algorithms and set  $\alpha = 1 - \lambda$ ,  $\beta = 1$ .  $\lambda$  is initialized to 0.1 and divided by a constant factor after each iteration.

All the above three cost functions can be minimized by the standard conjugate gradient algorithm [19].

### 3.2 Enforcing Upper and Lower Bounds

In the above formulation, shadow constraints are enforced as soft constraints by using penalty terms in the original shape-from-shading algorithm. It is not guaranteed that all constraints are satisfied. Sometimes, it is more desirable to consider shadow constraints as hard constraints since they are less sensitive to specular reflection and albedo variations, and to consider shading constraints as soft ones since a little bit deviation in shading is not very noticeable. The upper and lower bounds discussed in Section 2 can serve this purpose and can be estimated as follows. (Note that the height of the nodes in the set  $V_t$  is unknown at the beginning, and they can be estimated from a solution of the height field from Section 3.1.)

1. Obtain an initial estimation of the height value for each point by minimizing Eq. (10);



**Fig. 2.** (a) Some parts of the height field recovered from minimization may exceed the upper bound; (b) We need to globally adjust the initial height field to maintain its original smoothness instead of simply clipping it against the upper bound.

2. Adjust the initial height values of the nodes in  $V_t$  to satisfy all the antishadow constraints among them as in the following convergent procedure;
  - a) fix the height of the highest point in  $V_t$ ;
  - b) loop through the rest of the points and check whether the considered point is in the shadow of some other point in  $V_t$  because of the violation of a antishadow constraint; if so, raise the considered point to the minimum height that can eliminate the violation.
3. Calculate the upper and lower bounds for nodes in  $V - V_t$  from the transitive closure  $G^c$ .

To enforce the upper and lower bounds, our complete algorithm still takes an initial solution of the height field from minimizing Eq. (10). However, there are multiple possibilities to improve this initial solution:

1. For each point, if it is higher than its upper bound, push it down to the upper bound; if it is lower than its lower bound, raise it to the lower bound.
2. Use a constrained optimization algorithm such as sequential quadratic programming to enforce the upper and lower bounds.
3. Fix a subset of the adjusted points from the first step and minimize Eq. (10) with those fixed points as additional boundary conditions; alternate adjustment and minimization (with a few additional fixed points every iteration) until all the bounds are satisfied.

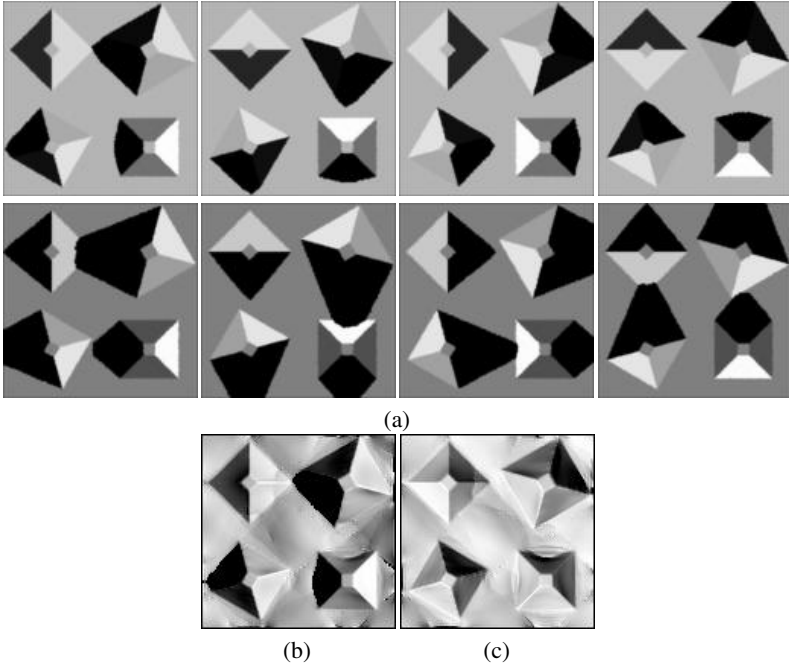
The first scheme chooses to satisfy all the hard constraints by using brute force and ignoring all the shading constraints, therefore tends to have unnatural discontinuities at those adjusted places. The second scheme chooses to apply some constrained optimization algorithm to automatically and iteratively adjust the heights so that the bounds are satisfied at the end. Unfortunately, constrained optimization algorithms such as sequential quadratic programming (SQP) are usually computationally expensive on high-dimensional data such as images. For example, the SQP software package [6] we tried took two hours to finish one iteration on 64x64 images on a Pentium III 800MHz workstation. The last scheme chooses to adapt unconstrained optimization algorithms so that they allow a part of the variables to be fixed. To achieve that, we can simply set the corresponding derivatives to be zero. We fix a few additional points within their bounds before unconstrained minimization takes place in every iteration, therefore can satisfy all the bounds in a finite number of iterations since we only try to recover height values at a finite number of points (pixels). An intuitive illustration is given in Fig. 2.

In practice, we chose the last scheme with some additional details. After initialization, the height values of the nodes in  $V_t$  and the upper and lower bounds are fixed in all iterations. In every iteration, we subtract the upper bounds from the current estimation of the height field to obtain a difference field. Then the set of local maxima in the difference field are located. Those points corresponding to the local maxima are lowered to their corresponding upper bounds and fixed thereafter. The same procedure is repeated for lower bounds before the unconstrained minimization in Eq. (10) takes place once again with the newly fixed points as additional boundary conditions. We hope that the shading constraints solved during minimization can automatically adjust the neighborhoods of those fixed points so that there will be much less violated bounds in the next iteration. This

can also avoid having many unnatural discontinuities since the minimization procedure serves as a smoothing operator by considering all constraints simultaneously.

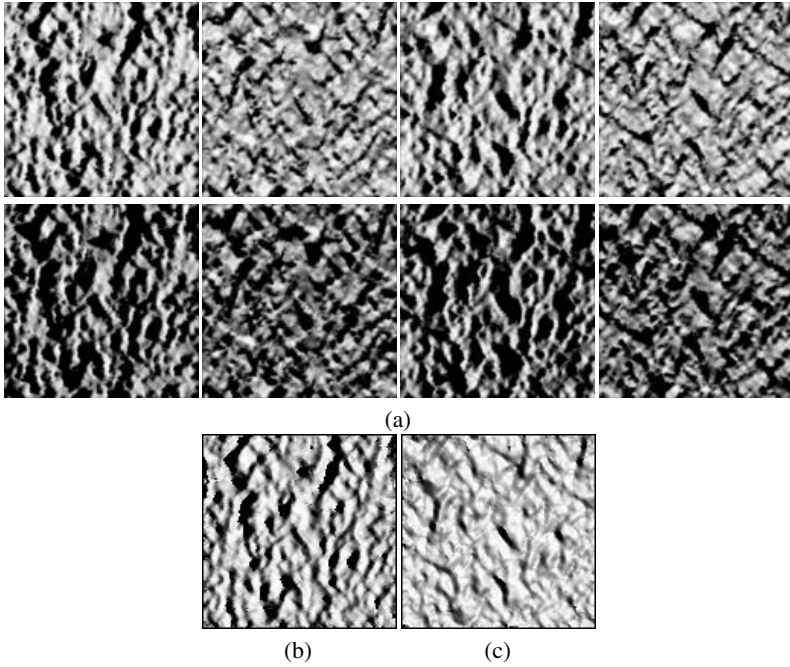
### 3.3 Experiments

We have tested the algorithm using integrated shadow and shading constraints on both synthetic and real imagery.



**Fig. 3.** (a) Input images for the pyramid scene. The tilt angle of the lighting directions in the top row is 45 degrees, the bottom row 60 degrees. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.

**Synthetic data.** Eight synthetic images were generated as input for each of the three representative datasets we chose. Four of them were lit from a tilt angle of 45 degrees and the others were lit from 60 degrees to create images with significant amount of shadow. We also generated two images for each example from the recovered height field. The first image is lit from the same lighting direction as the first input image to verify both shadowed and non-shadowed regions. The second image is lit from a novel lighting direction which is different from the ones for the input images to show that the recovered height fields can be useful for creating images with correct appearance



**Fig. 4.** (a) Input images for the plaster material sample. It is lit from the same set of lighting directions as in Fig. 3. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.

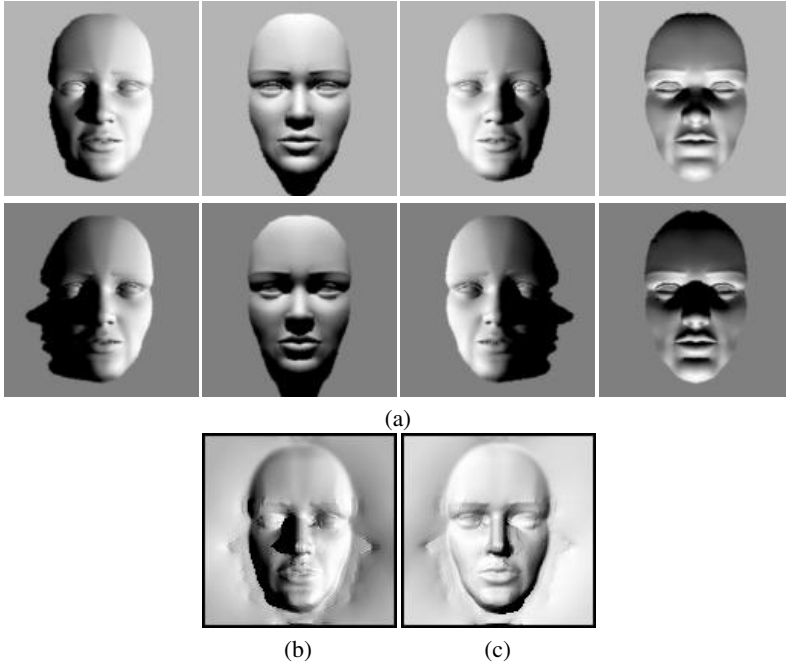
**Table 1.** Comparison of the three approaches on the three datasets: i) minimizing  $E_2$  in Eq.(7), ii) minimizing  $E_3$  in Eq.(10), iii) enforcing bounds as in Section 3.2. The top table shows the RMS errors of the recovered height fields using noise free input images, and the bottom one shows the RMS errors using images with 5% noise. All numbers are given in the unit of a pixel.

0% noise	$E_2$	$E_3$	$E_3$ +Bounds
Pyramids	3.6579	2.2424	2.1984
Plaster	1.9344	1.4548	1.4210
Face	4.4164	3.3335	3.3399

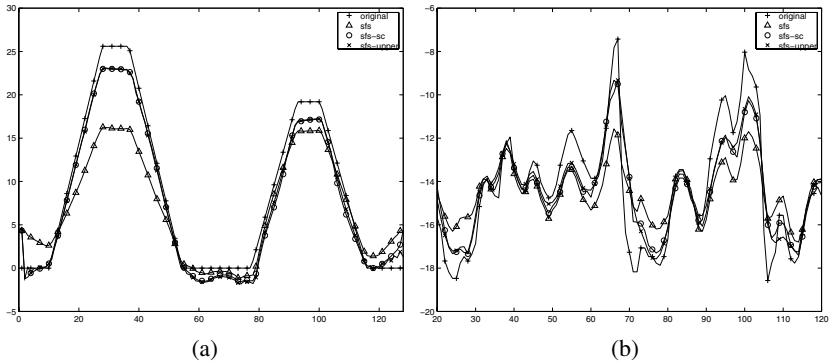
5% noise	$E_2$	$E_3$	$E_3$ +Bounds
Pyramids	3.7621	2.2675	2.2100
Plaster	1.9400	1.4089	1.3959
Face	4.4522	3.4298	3.4159

from novel lighting conditions. We also compared the recovered height fields with the ground truth to obtain error measurements which are shown in Table 1. In our examples, most points are lit from at least one lighting direction. The height field can be recovered from shape-from-shading or photometric stereo alone. However, the additional shadow constraints can definitely improve the accuracy of the results because shading-based techniques can introduce accumulated errors from pixel to pixel while shadow constraints are very good at enforcing long-range relative height constraints.

The first dataset is an artificial scene with four pyramids shown in Fig. 3(a). The pyramids have different height and orientation. The two synthetic images from the recovered



**Fig. 5.** (a) Input images for the face model. It is lit from the same set of lighting directions as in Fig. 3. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.



**Fig. 6.** Comparison of the cross sections of four height fields: the ground truth is shown as 'original'; the one from minimizing Eq. (7) is shown as 'sfs'; the one from minimizing Eq. (10) is shown as 'sfs-sc'; and the one from enforcing bounds is shown as 'sfs-upper'. (a) Cross sections for the pyramid scene; (b) cross sections for the plaster sample.

height field are shown in Fig. 3(b)&(c). The second dataset is a previously recovered height field of a real plaster sample using the approach presented in [16]. This height



field serves as the ground truth to test the algorithm in this paper although we do not know the accuracy of this dataset. The input images are shown in Fig. 4(a) and the synthetic images from the height field recovered by the current algorithm are shown in Fig. 4(b)&(c). The third dataset is a face model shown in Fig. 5(a). And Fig. 5(b)&(c) give the images generated from the recovered face. In this example, the background plane is pushed down along the shadow boundaries in some of the input images to satisfy the shadow constraints. This is because shape-from-shading related techniques are better at estimating normal orientation than at estimating height values, and generated an inaccurate initial solution for our algorithm. A similar situation was also shown in the pyramid scene. Nevertheless, our algorithm still managed to enforce the shadow constraints and make the generated images look similar to the input ones.

Fig. 6 shows two comparisons of the cross sections. In each of the comparisons, there are four curves including the ground truth, the curve from minimizing Eq. (7), the curve from minimizing Eq. (10) and the curve from enforcing the upper bounds. The results from minimizing Eq. (7) are not as good as the other two versions because it does not consider shadow constraints.

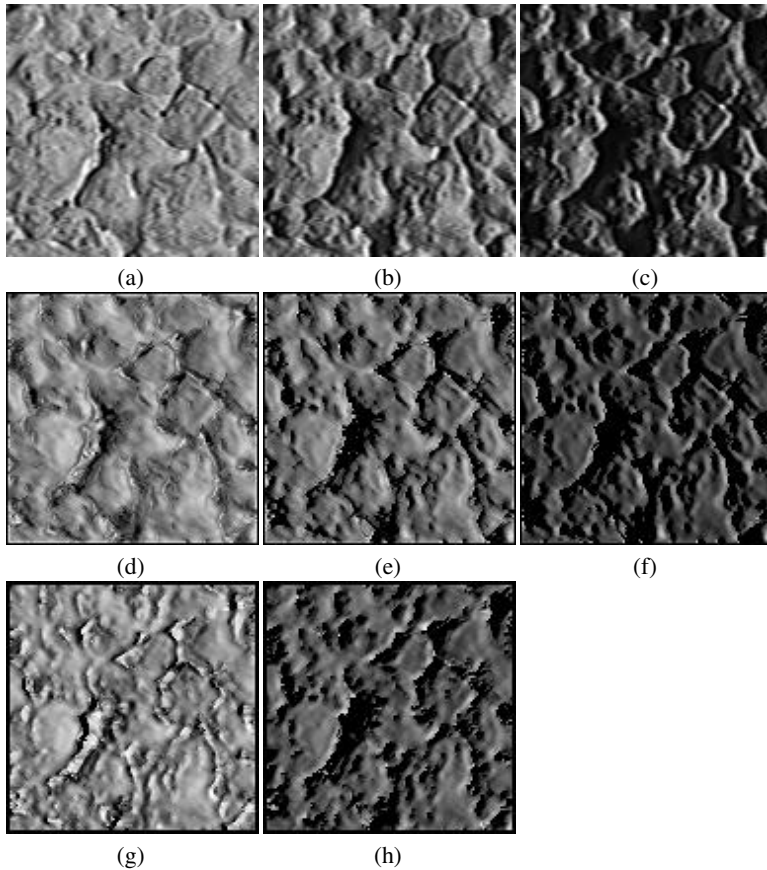
**Real data.** We also did test on a real dataset. Three 128x128 images of a concrete sample from the CURET database [3] were used as the input to our final algorithm. They have various amount of shadow (Fig. 7(a)-(c)). We use 15 as the intensity threshold to detect shadowed pixels. The lighting directions of the input images are actually coplanar. Traditional photometric stereo would have problem to recover the height field. However, our algorithm successfully recovered it since it exploits shadow constraints and a regularization term. Minimizing  $E_3$  in Eq. (10) took 5 minutes on a Pentium III 800MHz processor, and the iterative procedure for enforcing bounds took another half an hour. Synthetic images were generated from the recovered height field. The recovered dataset was illuminated from both original lighting directions (Fig. 7(d)-(f) of the input images and novel lighting directions (Fig. 7(g)-(h)).

## 4 Summary

We presented the concept of shadow graphs and proved that the shadow graph alone is enough to solve the shape-from-shadow problem from a dense set of images. We also developed a method of recovering shape from both shadow and shading constraints. A constrained optimization procedure has been developed to make the results from shape-from-shading consistent with the upper bounds derived from shadow constraints.

Future work includes more robust techniques that allow inaccurate shadow segmentation and the simultaneous recovery of shape and lighting directions from both shading and shadows.

**Acknowledgment.** This work was supported by National Science Foundation CAREER Award CCR-0132970.



**Fig. 7.** (a)-(c) Real images of a concrete sample; (d)-(f) synthetic images of the recovered height field illuminated from original lighting directions; (g)-(h) synthetic images of the recovered height field illuminated from two novel lighting directions.

## References

1. P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *Int. Journal Comp. Vision*, 28(3):1–16, 1998.
2. P. Belhumeur, D. Kriegman, and A. Yuille. The bas-relief ambiguity. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 1040–1046, 1997.
3. K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real-world surfaces. In *Proceedings of CVPR*, pages 151–157, 1997.
4. M. Daum and G. Dudek. On 3-d surface reconstruction using shape from shadows. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 461–468, 1998.
5. P. Dupuis and J. Oliensis. Shape from shading: Provably convergent algorithms and uniqueness results. In *Computer Vision-ECCV 94*, pages 259–268, 1994.
6. Fsqp software. <http://gachinese.com/aemdesign/FSQPframe.htm>. Originally developed at the Institute for Systems Research, University of Maryland.

7. A. Georgiades, P. Belhumeur, and D. Kriegman. Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. In *IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pages 47–54, 1999.
8. R. Hassin and S. Rubinstein. Approximations for the maximum acyclic subgraph problem. *Information Processing Letters*, 51:133–140, 1994.
9. M. Hatzitheodorou. The derivation of 3-d surface shape from shadows. In *Proc. Image Understanding Workshop*, pages 1012–1020, 1989.
10. B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics & Image Processing*, 33:174–208, 1986.
11. K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Trans. Patt. Anal. Mach. Intel.*, 3(6):661–669, 1981.
12. D. Jacobs. Linear fitting with missing data: Applications to structure from motion and characterizing intensity images. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 206–212, 1997.
13. J. Kender and E. Smith. Shape from darkness. In *Int. Conf. on Computer Vision*, pages 539–546, 1987.
14. D.J. Kriegman and P.N. Belhumeur. What shadows reveal about object structure. In *Computer Vision-ECCV 98*, 1998.
15. Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 552–558, 1991.
16. X. Liu, Y. Yu, and H.-Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *Proceedings of SIGGRAPH*, pages 97–106, 2001.
17. S.K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Trans. Robotics and Automation*, 6(4):418–431, 1990.
18. S.K. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflections. *International Journal of Computer Vision*, 6(3):2–11, 1991.
19. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge Univ. Press, New York, 1988.
20. S.A. Shafer. *Shadows and silhouettes in computer vision*. Kluwer Academic Publishers, 1985.
21. A. J. Stewart and M.S. Langer. Towards accurate recovery of shape from shading under diffuse lighting. *IEEE Patt. Anal. Mach. Intel.*, 19(9):1020–1025, 1997.
22. R.J. Woodham. Photometric method for determining surface orientation from multiple images. In B.K.P. Horn and M.J. Brooks, editors, *Shape from Shading*, pages 513–532. MIT Press, 1989.

# Specularities Reduce Ambiguity of Uncalibrated Photometric Stereo

Ondřej Drbohlav and Radim Šára

Center for Machine Perception, Department of Cybernetics  
Faculty of Electrical Engineering, Czech Technical University in Prague  
Technická 2, 166 27 Prague 6, Czech Republic  
{drbohlav,sara}@cmp.felk.cvut.cz

**Abstract.** Lambertian photometric stereo with uncalibrated light directions and intensities determines the surface normals only up to an invertible linear transformation. We show that if object reflectance is a sum of Lambertian and specular terms, the ambiguity reduces into a 2dof group of transformations (compositions of isotropic scaling, rotation around the viewing vector, and change in coordinate frame handedness).

Such ambiguity reduction is implied by the *consistent viewpoint constraint* which requires that all lights reflected around corresponding specular normals must give the same vector (the viewing direction). To employ the constraint, identification of specularities in images corresponding to four different point lights in general configuration suffices. When the consistent viewpoint constraint is combined with integrability constraint, binary convex/concave ambiguity composed with isotropic scaling results. The approach is verified experimentally.

We observe that an analogical result applies to the case of uncalibrated geometric stereo with four affine cameras in a general configuration observing specularities from a single distant point light source.

## 1 Introduction

Photometric stereo [13] is a method that recovers local surface geometry and reflectance properties from images of an object that are taken by a fixed camera under varying distant illumination. The principle of photometric stereo is in inverting a parametric model of surface reflectance. A usual version of photometric stereo uses a single distant point light source at a time to illuminate an object, and assumes Lambertian surface reflectance which implies that brightness value  $I_{i,j}$  of  $i$ -th pixel in the image capturing the object appearance under  $j$ -th point light source is (see Fig. 1)

$$I_{i,j} = E_j \rho_i \cos \theta_{i,j} = (\rho_i \mathbf{n}_i)^\top (E_j \mathbf{l}_j), \quad (1)$$

where  $E_j$  is the intensity of the light source,  $\mathbf{l}_j$  is the light source direction,  $\mathbf{n}_i$  is the normal vector of a surface patch that projects into the  $i$ -th pixel,  $\theta_{i,j}$

is the angle between  $\mathbf{n}_i$  and  $\mathbf{l}_j$  (the angle of incidence), and  $\rho_i$  is a reflectance parameter of a small surface patch. This parameter is called *albedo* and describes what portion of incident light is re-emitted back into space in the form of diffuse reflection.

It is well understood that the reflectance model described by equation (1) is bilinear. To see that, it is convenient to denote  $\rho_i \mathbf{n}_i$  and  $E_j \mathbf{l}_j$  by  $\mathbf{b}_i$  and  $\mathbf{s}_j$ , respectively; then, the above equation takes very simple, compact form

$$\mathbf{I} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N]^\top [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M] = \mathbf{B}^\top \mathbf{S}, \quad (2)$$

where  $\mathbf{I}$  is the matrix which has  $I_{i,j}$  from (1) as its elements, matrix  $\mathbf{B}$  collects the  $\mathbf{b}_i$ 's and matrix  $\mathbf{S}$  analogically collects the  $\mathbf{s}_j$ 's. For the sake of brevity, we call  $\mathbf{B}$  the *normals* and  $\mathbf{S}$  the *lights*.

In the original version of photometric stereo [13], the light source directions  $\mathbf{l}_j$  and intensities  $E_j$  are calibrated, thus the lights  $\mathbf{S}$  are known. To evaluate normals and albedos, it suffices to right-multiply the matrix  $\mathbf{I}$  in (2) by the inverse (or pseudo-inverse if the number of lights is greater than 3) of  $\mathbf{S}$ ; by that the normals  $\mathbf{B}$  are acquired. Normals  $\mathbf{n}_i$ 's are then  $\mathbf{b}_i$ 's scaled to unity, and albedos  $\rho_i$ 's are the lengths of  $\mathbf{b}_i$ 's.

If, however, the light sources  $\mathbf{S}$  are *not* known, then (2) represents a bilinear calibration-estimation problem [8] whose ambiguity can be phrased as follows:

**Uncalibrated photometric stereo ambiguity.** Let there be images of an object of Lambertian reflectance observed from a fixed viewpoint, but illuminated sequentially from different unknown directions by a distant point light source. Then it is possible to factorize the input data matrix  $\mathbf{I}$  from (2) into *pseudonormals*  $\bar{\mathbf{B}}$  and *pseudolights*  $\bar{\mathbf{S}}$  [6] that give the true normals  $\mathbf{B}$  and the true lights  $\mathbf{S}$  up to an unknown linear invertible transformation  $\mathbf{A} \in GL(3)$ :  $\bar{\mathbf{B}} = \mathbf{A}\mathbf{B}$ ,  $\bar{\mathbf{S}} = \mathbf{A}^{-\top}\mathbf{S}$ .

This ambiguity exists because it holds that  $\mathbf{I} = \bar{\mathbf{B}}^\top \bar{\mathbf{S}} = \mathbf{B}^\top \mathbf{A}^\top \mathbf{A}^{-\top} \mathbf{S} = \mathbf{B}^\top \mathbf{S}$ . The uncalibrated photometric stereo ambiguity can be reduced and/or removed only if additional information about lights or normals is available. This information may have different form. First possibility is to estimate normal vectors and albedos in several points by an independent method and use them to disambiguate the photometric stereo (note that due to the symmetry of (2), the value of this information is the same as if light directions and intensities are known). Another possibility is to assume that at least six light sources are of equal (or known relative) intensity, or that albedo is uniform (or known up to a global scaler) for at least six normals at a curved surface. Such possibilities were employed and/or discussed in [6,14,1], and it was shown that such knowledge reduces the ambiguity from the  $GL(3)$  group into the group of scaled orthogonal transformations  $\mathbf{A} = \lambda \mathbf{O}$  ( $\mathbf{O} \in O(3)$ ,  $\lambda \neq 0$ ). Yet another important possibility is given by the integrability constraint that requires the normals recovered by photometric stereo to correspond to a continuous surface [1,4]. As shown by

Belhumeur et al. [1], in this case the original ambiguity is reduced into ambiguity represented by the group of four-parametric (or three-parametric, if overall scaling is not counted) generalized bas-relief transformations. And, integrability constraint together with the knowledge of six albedos (or six light intensities as described above) reduces the original ambiguity into binary convex/concave ambiguity composed with isotropic scaling.

In our recent work on uncalibrated photometric stereo [3] we showed that inherent symmetries of reflectance models that are separable with respect to the viewing and illumination directions can be exploited to construct two new geometrical constraints. The constraints are represented by projections of normals onto planes perpendicular to the viewing and illumination directions, respectively. We constructed the constraints using polarization measurement under the assumption of separable reflectance model for smooth dielectrics and showed that the two constraints alone combined together reduce the ambiguity to convex/concave ambiguity composed with isotropic scaling.

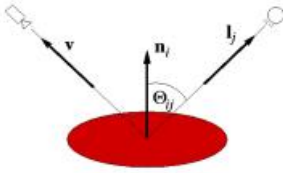
In this paper we show that if object reflectance is a sum of Lambertian reflectance and a mirror-like reflectance, then the original ambiguity represented by a group  $GL(3)$  reduces into a two-parametric group of transformations. These transformations are compositions of isotropic scaling (1dof), rotation around the viewing vector (1dof), and change in the global coordinate frame handedness (binary ambiguity). This ambiguity reduction is implied by a condition that all lights reflected around corresponding specular normals must give the same vector (the viewing direction). We call this condition the *consistent viewpoint constraint*. We show that specularities in as few as *four* images corresponding to four different distant point lights in general configuration are sufficient to utilize the consistent viewpoint constraint.

By this result, we make a step towards *uncalibrated* photometric stereo for objects whose reflectance includes not only body (diffuse) component, but also interface (specular) component. Such composite reflectance models are certainly not new to photometric stereo applications, see e.g. [2,11,10,9], but in those methods, in contrast to the ours, the light sources are supposed to be known.

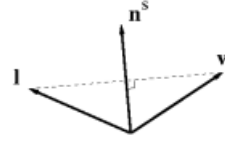
The specific representative of composite reflectance model (the superposition of Lambertian and mirror-like reflectance) is selected in this work because as specularities are sparse in the images, they can be treated as *outliers* to the *Lambertian* reflectance model. This gives us a valuable possibility to study the problem as Lambertian photometric stereo with additional information represented by the consistent viewpoint constraint.

## 2 Consistent Viewpoint Constraint

The problem we will analyze is photometric stereo with uncalibrated lights  $\mathbf{S}$  for objects whose reflectance is given by superposition of Lambertian and specular terms. As discussed in Section 1, we treat this problem as uncalibrated Lambertian photometric stereo with additional geometrical information provided by



**Fig. 1.** Reflectance geometry. For Lambertian reflectance, brightness seen by a camera is dependent on cosine of the angle of incidence and independent on the viewing direction.



**Fig. 2.** Specular geometry configuration. Surface normal  $\mathbf{n}^S$  is a bisector between the viewing direction  $\mathbf{v}$  and the illumination direction  $\mathbf{l}$ .

specularities. In this section we review the geometry of mirror-like reflection, and formulate the constraint implied by the observation of specularities in images.

It is well understood that specularities occur at loci where light reflects on a smooth surface in a mirror-like manner towards the observing sensor. Hence, mirror-like reflection carries valuable information about geometrical configuration of the viewing vector, the illumination vector, and the surface normal: if a specularity is observed in an image, then at the corresponding surface point, surface normal is a bisector of the viewing and illumination vectors (see Fig. 2). Therefore for the viewing direction it holds that

$$\mathbf{v} = 2(\mathbf{l} \cdot \mathbf{n}^S)\mathbf{n}^S - \mathbf{l} = 2(\hat{\mathbf{s}} \cdot \hat{\mathbf{b}}^S)\hat{\mathbf{b}}^S - \hat{\mathbf{s}} \quad (3)$$

where  $\hat{\cdot}$  denotes normalization to unity, and  $\mathbf{n}^S$  is a normal that is observed as specular under illumination of direction  $\mathbf{l}$ . The right-most part of the equation essentially states the same fact in “natural” photometric stereo variables. We call  $\mathbf{l}$  and  $\mathbf{n}^S$  (as well as  $\mathbf{s}$  and  $\mathbf{b}^S$ ) a *specular pair*.

The equation may be viewed as a formula for computing viewpoint direction from known light  $\mathbf{s}$  and specular normal  $\mathbf{b}^S$ . The key fact to be observed is that this relation states: no matter which specular pair is used for viewing direction evaluation, all give the same result.

**Consistent viewpoint constraint.** A collection of specular pairs follows the *consistent viewpoint constraint* if they all, by (3), give the same viewing direction  $\mathbf{v}$ .

Does the consistent viewpoint constraint reduce the uncalibrated photometric stereo ambiguity? We will analyze what transformations may be applied to the true normals and the true lights, such that the transformed specular pairs, inserted into (3), all give the same vector. Let us denote this vector  $\mathbf{u}$  and write the equivalent of (3) for the transformed lights and normals:

$$\mathbf{u} = \frac{2[(\mathbf{A}^{-\top}\mathbf{s}) \cdot (\mathbf{A}\mathbf{b}^S)]\mathbf{A}\mathbf{b}^S}{\|\mathbf{A}^{-\top}\mathbf{s}\| \|\mathbf{A}\mathbf{b}^S\|^2} - \frac{\mathbf{A}^{-\top}\mathbf{s}}{\|\mathbf{A}^{-\top}\mathbf{s}\|}, \quad (4)$$

where  $\|\cdot\|$  are explicitly written normalization factors. Multiplying both sides of the equation by  $\|\mathbf{A}^{-\top}\mathbf{s}\| \|\mathbf{A}\mathbf{b}^S\|^2 \mathbf{A}^\top$ , we get

$$\alpha(\mathbf{s}, \mathbf{b}^S) \mathbf{w} = 2(\mathbf{s} \cdot \mathbf{b}^S) \mathbf{P} \mathbf{b}^S - (\mathbf{b}^S \cdot \mathbf{P} \mathbf{b}^S) \mathbf{s}, \quad (5)$$

where  $\alpha(\mathbf{s}, \mathbf{b}^S) = \|\mathbf{A}^{-\top}\mathbf{s}\| \|\mathbf{A}\mathbf{b}^S\|^2$  absorbs (unknown) scaling factors,  $\mathbf{P}$  denotes  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{w}$  denotes  $\mathbf{A}^\top \mathbf{u}$ ; and we applied the fact that  $(\mathbf{A}^{-\top}\mathbf{s}) \cdot (\mathbf{A}\mathbf{b}^S) = \mathbf{s} \cdot \mathbf{b}^S$ . Note that in this equation, vector  $\mathbf{w}$  may be treated as fully independent on  $\mathbf{P}$  because  $\mathbf{P} = \mathbf{A}^\top \mathbf{A}$  gives  $\mathbf{A}$  only up to arbitrary orthogonal transformation. We show in Appendix that for a convex smooth specular object illuminated from all directions it must hold that  $\mathbf{P} = \lambda^2 \mathbf{I}$ ,  $\lambda \neq 0$  ( $\mathbf{P}$  is a scaled identity). From that it follows that the only transformations under which the consistent viewpoint constraint is preserved are  $\mathbf{A} = \lambda \mathbf{O}$ ,  $\mathbf{O} \in O(3)$ . Fixing the coordinate frame by a usual choice (image plane spans plane  $x - y$ , viewing direction coincides with axis  $z$ ), the allowable transformations  $\mathbf{A}$  are those that preserve the viewing direction. Writing them explicitly,

$$\mathbf{A} = \lambda \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \mathbf{R}_z(\xi) \quad \xi \in \langle 0, 2\pi \rangle, \lambda > 0. \quad (6)$$

In this equation,  $\mathbf{R}_z(\xi)$  stands for rotation around the  $z$ -axis (the viewing direction) by angle  $\xi$ . The ambiguity in sign of the third coordinate was included in (6) only for the sake of completeness, but naturally the correct sign is easily set by orienting the normals towards the viewing direction (normals that are inclined from the viewpoint are invisible). The ambiguity in sign of the first coordinate stays unresolved until some additional constraint is applied (or until it is resolved manually).

In this paper we resolve the remaining ambiguity using the integrability constraint. Integrability constraint fixes both the sign of the first coordinate (thus the handedness of the coordinate system) *and* the rotation angle  $\xi$  modulo  $\pi$ . This means that the final ambiguity is convex/concave ambiguity composed with isotropic scaling. This result follows from the fact that the intersection of the  $O(3)$  group (of which transformations (6) with  $\lambda = 1$  are a sub-group) with the generalized bas-relief group is a two-element set of the identity transformation and the transformation that reflects the first two coordinates [1].

Finally, let us observe how many specular pairs do we need to establish the consistent viewpoint constraint. Equation (5) represents three scalar equations for each specular pair. After eliminating the unknown constant  $\alpha(\mathbf{s}, \mathbf{b}^S)$ , there are two independent equations per specular pair. The unknowns  $\mathbf{w}$  and  $\mathbf{P}$  are both up to scale, so the number of degrees of freedom to fix is 2 (from the vector  $\mathbf{w}$ ) plus 5 (from the symmetric matrix  $\mathbf{P}$ ). We thus observe that at least *four* specular pairs in general configuration are needed to apply the consistent viewpoint constraint.

So far, we have not analyzed which configurations of four specular pairs are singular, nor the problem of (possible) finite solution multiplicity for non-singular configurations. However, in experiments we observed unique solution



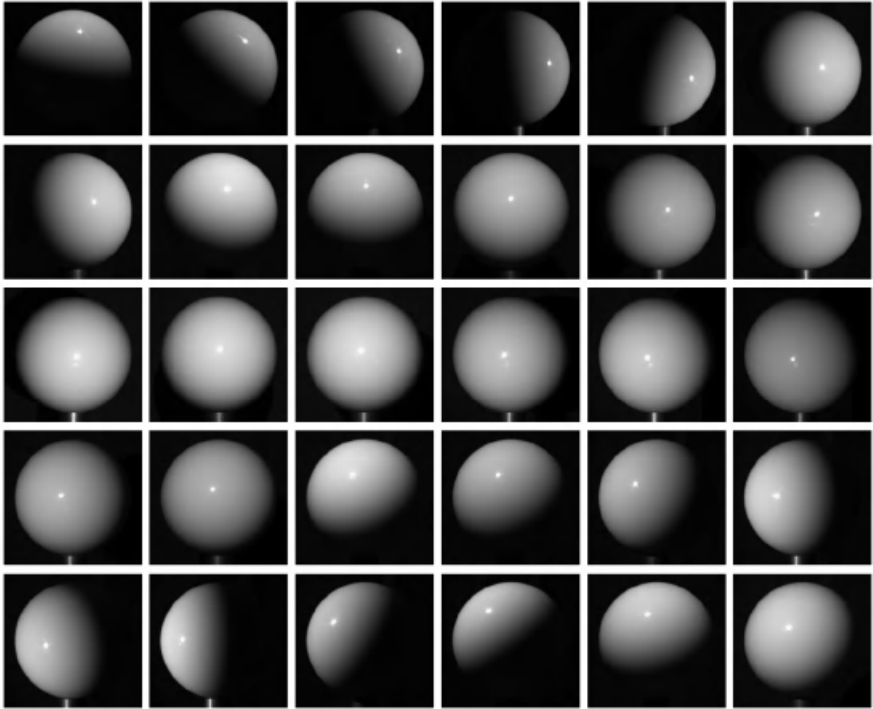
in all cases. Analysis of which sets of four specular pairs give well-conditioned solution is a topic for future research.

### 3 Experiment

In this experiment we show normal and albedo reconstruction for two objects:

1. *WhiteBall* which is a highly polished billiard ball of uniform albedo,
2. *ChinaPot* which is a glazed china tea pot with painted motif.

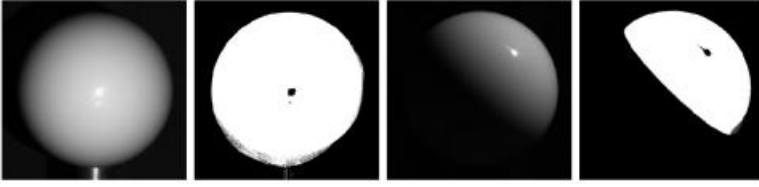
Images were acquired by 12 bit cooled camera (COOL-1300 by Vosskühler, standard Computar 75mm lens) under tungsten illumination (150W, stabilized direct current). The light was moved by hand around the object. The distance between object and light was not kept constant. No information about lights has been measured nor recorded. Input images for the *WhiteBall* object are shown in Fig. 3.



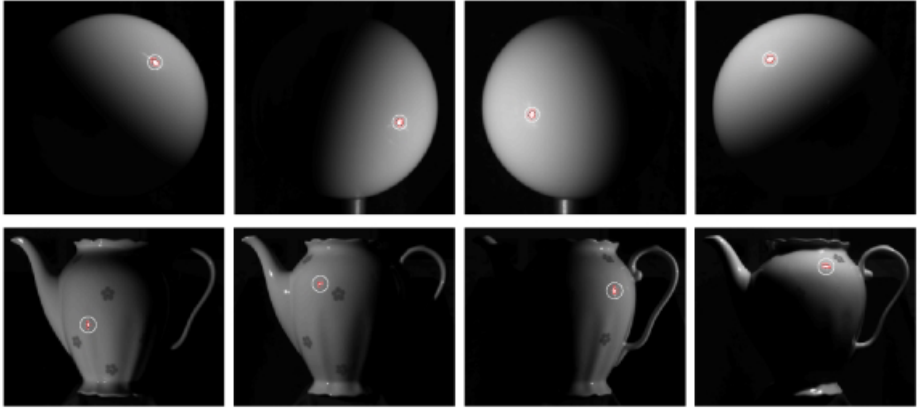
**Fig. 3.** Input data for the *WhiteBall* object.

Data was processed in 9 consecutive steps:

1. The mean of 10 dark frames was subtracted from each of the input images.

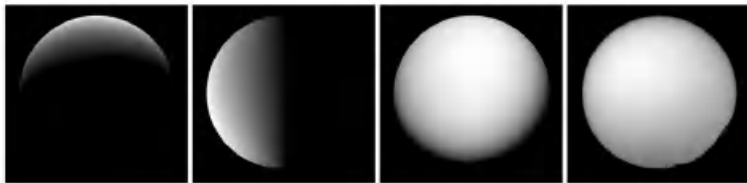


**Fig. 4.** Two examples of Lambertian behaviour masks for the WhiteBall object.

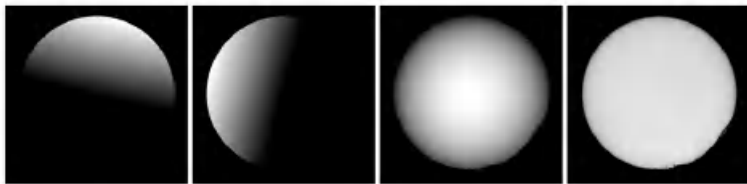


**Fig. 5.** Selected specular regions (marked with red contour and rounded by small white circle for better reading) used to apply the consistent viewpoint constraint, for both objects.

2. Image points whose intensity fell below or over respective thresholds were labeled as non-Lambertian, the other ones as *candidates* for Lambertian points.
3. Candidates for Lambertian points were confirmed to be Lambertian in the case that in four randomly selected images, they belonged to sufficiently large image pixel sets with Lambertian behavior (i.e., if intensities of the pixel set in any of four images could be sufficiently well expressed as a linear combination of intensities of the pixel set in the other three images). Only such quadruples of images were involved whose any three corresponding light directions were sufficiently far from being coplanar (automatic check of this condition was done using simple conditioning number tests). Two selected Lambertian-consistent masks resulting from this step are shown in Fig. 4.
4. Lambertian portion of data was factorized by Jacobs algorithm [7]. From the factorization pseudolights  $\tilde{\mathbf{S}}$  were obtained.
5. Pseudonormals  $\tilde{\mathbf{B}}$  were computed using Lambertian image regions and the pseudolights obtained in the previous step. Each normal was fit individually by using least-square fit. After that, pseudolights were re-evaluated by an analogous procedure, and this iterative process (alternating between re-



(a) Normals illuminated from the coordinate axis directions (left, top, viewpoint) and albedo (last image) reconstructed by Jacobs method from segmented input data assuming Lambertian reflectance. Note that normals look as illuminated from directions other than specified. The albedo map does not correspond to expectations either, since in reality it is uniform over the object.



(b) Normals (illuminated as above) and albedo conforming to the consistent viewpoint constraint. Note that images already appear as illuminated from three mutually perpendicular directions and that the albedo map is uniform, as expected.

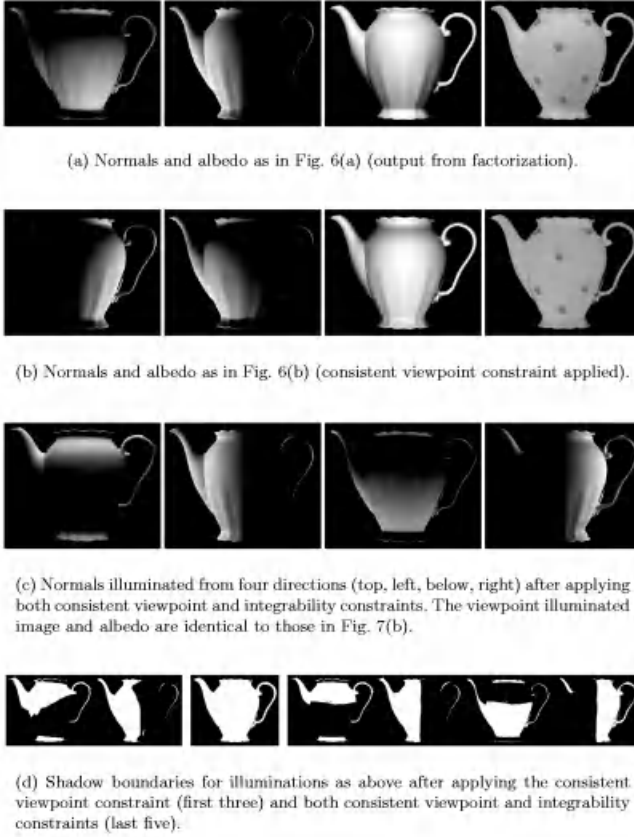


(c) Normals illuminated from the left and top directions conforming to both the consistent viewpoint and integrability constraints. The viewpoint-illuminated image and albedo are identical to those in Fig. 6(b). Note that the original ambiguity (cf. Fig. 6(a)) is already canceled.

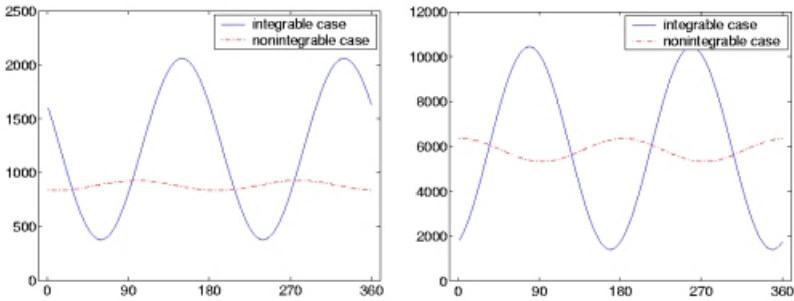


(d) Shadow boundaries for illuminations as above after applying the consistent viewpoint constraint (first three) and both consistent viewpoint and integrability constraints (last three).

**Fig. 6.** Results on the WhiteBall object.



**Fig. 7.** Results on the ChinaPot object.



**Fig. 8.** Integrability violation measure as a function of rotation of normals around the viewing direction ( $\xi \in (0, 360)$  [deg]). The WhiteBall object (left) and the ChinaPot object (right). The two plots in each graph (shown in red and blue) correspond to two coordinate frame handednesses. Normals are integrable in only one of them.

computation of pseudonormals and pseudolights as in [5]) was repeated 10 times. The residual (the sum of squared differences between the predicted and observed intensities over the valid image regions) converged to about 1/3 of its initial value. The result of this step is shown in Fig. 6(a) for the WhiteBall object and in Fig. 7(a) for the ChinaPot object. Note neither the illuminated normals nor albedo do correspond to our expectations.

6. Specularly reflecting normals in four images were determined. Specular regions were selected by hand from the set of segmented non-Lambertian regions available from Step 3. The selections are shown in Fig. 5. Pseudonormals from the previous step were averaged over the whole extension of the respective specularity.
7. A transformation  $\mathbf{A}$  was found that maps specular pseudonormals and corresponding pseudolights onto those which fulfill the consistent viewpoint constraint. The idea of the algorithm was to design ‘ideal’ specular pairs (which follow the consistent viewpoint constraint *exactly*), and look for transformation  $\mathbf{A}$  that maps experimentally obtained specular pseudonormals and pseudolights *closest* (in a least-square sense) to these ideal specular pairs. The algorithm was essentially of the same type as the well-known bundle-adjustment method and we acknowledge the article of Triggs et al. [12] that helped us to design it. The transformation itself was parametrized as  $\mathbf{A} = \text{diag}[1, \lambda_2, \lambda_3] \mathbf{V}^{\top 1}$ . A unique solution existed in both objects.
8. This transformation was applied to pseudonormals and pseudolights output from Step 5. The consistent viewpoint direction resulting from the previous step was rotated to  $[0, 0, 1]^{\top}$ .

The results of this step are shown in Figs. 6(b) and 7(b). Note that the resulting albedo is uniform, as expected. It is already the disambiguated albedo of the object. Note also the viewpoint-illuminated normal fields are already symmetric in both objects, as expected.

9. Integrability constraint was applied to resolve the rotation angle  $\xi$  and the sign of the first coordinate in transformation (6). The integrability constraint violation measure was constructed as a sum of squared height differences computed over elementary loops in the normal field. Note that for integrable surface the change in height over a closed loop vanishes. The measure was computed for  $\xi \in (0, 2\pi)$  on pseudonormals output from Step 8 as well as on these pseudonormals with the  $x$ -component reflected (blue and red plots in Fig. 8). The coordinate frame handedness was selected according

---

<sup>1</sup> This corresponds to the SVD decomposition of  $\mathbf{A}$  (11) with  $\mathbf{U}$  set to the identity and  $\lambda_1$  set to 1. Specifically, the optimized parameters were: viewpoint direction  $\mathbf{v}$ , four ideal specular normals  $\mathbf{n}^S$ , and the parameters of transformation  $\mathbf{A}$ . Ideal specular lights were computed by reflecting the (optimized) viewing direction  $\mathbf{v}$  around (optimized) ideal specular normals  $\mathbf{n}^S$ . Initial parameter values were:  $\lambda_2 = \lambda_3 = 1$ ,  $\mathbf{V}$  initiated randomly (random rotation, solution was observed not to be dependent on this initial choice), ideal specular normals were set to normalized specular pseudonormals, and  $\mathbf{v}$  to normalized average of these. The iterative algorithm converged quickly (in about 15 iterations) into a unique solution.

to which of the two plots gave lower minima. The two minima in such plot correspond to the convex-concave ambiguity which was resolved by hand. The result of this step is shown in Figs. 6(c) (illuminated from left and top directions) and 7(c) (illuminated from four directions as indicated). Note the normals look as expected. This is confirmed by self-shadow boundaries as shown in Fig. 6(d) and 7(d).

## 4 Implications

In this section we show that there exist problems in computer vision that are formally very similar to the discussed problem of uncalibrated photometric stereo with consistent viewpoint constraint.

In Section 2, we analyzed the problem of uncalibrated photometric stereo, and the role of consistent viewpoint direction in constraining essentially affine ambiguity of uncalibrated photometric stereo. Here we show the validity of the following statement:

Let affine structure of an object be evaluated by affine geometric stereo with uncalibrated cameras. If four cameras in a general configuration observe specularities being reflected by the object surface from one distant point light source, then the original affine ambiguity reduces into similarity (composition of rotation, isotropic scaling, and change in coordinate frame handedness).

This statement follows from the analysis given in Section 2, where all the results apply if we make substitution *light direction*  $\leftrightarrow$  *viewing direction*.

To check the validity of this observation in detail, let there be a surface  $\mathbf{X}$  that is parametrized by  $u$  and  $v$ ,  $\mathbf{X} = \mathbf{X}(u, v)$ . Geometrical stereo with uncalibrated affine projection matrices evaluates the shape up to an affine transformation because the projections  $\mathbf{x}^j(u, v)$  of point  $\mathbf{X}(u, v)$  in the  $j$ -th affine camera  $\mathbf{C}^j$  are<sup>2</sup>

$$\mathbf{x}^j(u, v) = \begin{bmatrix} C_{1,1}^j & C_{1,2}^j & C_{1,3}^j \\ C_{2,1}^j & C_{2,2}^j & C_{2,3}^j \end{bmatrix} \begin{pmatrix} X_1(u, v) \\ X_2(u, v) \\ X_3(u, v) \end{pmatrix} = [\mathbf{C}_1^j, \mathbf{C}_2^j]^\top \mathbf{X}(u, v) = \mathbf{C}^{jT} \mathbf{X}(u, v) \quad (7)$$

and thus  $\mathbf{x}^j(u, v)$ 's are invariant under transformation  $\mathbf{C}^j \mapsto -\mathbf{A}\mathbf{C}^j$ ,  $\mathbf{X} \mapsto -\mathbf{A}^{-\top}\mathbf{X}$ , where  $\mathbf{A} \in GL(3)$ . It is known (see Yuille et al. [15]) that under these affine transformations the camera viewing vectors  $\mathbf{v}_j \sim \mathbf{C}_1^j \times \mathbf{C}_2^j$  are transformed covariantly and the surface normals  $\mathbf{n} \sim \frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v}$  are transformed contravariantly:  $\mathbf{v} \sim \mathbf{A}^{-\top} \mathbf{v}$ , and  $\mathbf{n} \sim \mathbf{A} \mathbf{n}$  ( $\sim$  means “up to a scaling factor”). Thus the normals in affine geometrical stereo transform like in photometric stereo, and the camera viewing vectors behave just like the illumination directions. But, in addition,

<sup>2</sup> Origins of image frames in all cameras are aligned.

the specular geometry condition (see Fig. 2) is also symmetrical with respect to the change  $\mathbf{l} \leftrightarrow \mathbf{v}$ . After we formulate the equivalent of the consistent viewpoint constraint (in this case, it could be called *consistent specular illumination constraint*), the mathematics of the problem is the same. The affine ambiguity is therefore reduced into composition of scaling, rotation around (unknown) illumination direction, and change in coordinate frame handedness; but this is the similarity ambiguity.

## 5 Conclusions

As a basic result of this paper we have shown that if object reflectance is a sum of Lambertian and specular terms, the uncalibrated photometric stereo ambiguity is reduced into effectively 2dof group of transformations (compositions of rotation around the viewing vector, isotropic scaling and change in coordinate frame handedness). For that, identification of specularities in images corresponding to four different distant point lights in general configuration is sufficient. We expect a similar result will hold if the specular spike is blurred by isotropic surface roughness. This result brings us closer to the practical situation when ‘one waves a torch in front of an object and Euclidean structure is revealed.’ The good applicability of the approach was verified experimentally on two real objects made of different material.

Note that albedo is obtained without imposing the integrability constraint. The integrability is used to fix only 1dof of the normal field. Since integrability must be computed on normal derivatives, any reduction of the number of parameters to be found significantly improves the accuracy of the resulting normals.

As we noted, lights and cameras play a symmetric role in the consistent viewpoint constraint. Hence, by interchanging lights and cameras, the constraint may also be applied to the case of uncalibrated geometric stereo with four affine cameras in a general configuration observing specularities from a single distant point light source.

**Acknowledgement.** This work was supported by the Grant Agency of the Czech Republic under Project GACR 102/01/1371 and in part by the Czech Ministry of Education under Research Programme MSM 212300013.

## References

1. P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, 1999.
2. E. N. Coleman and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. *Computer Graphics and Image Processing*, 18(4):309–328, 1982.
3. O. Drbohlav and R. Šára. Unambiguous determination of shape from photometric stereo with unknown light sources. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 581–586, 2001.

4. J. Fan and L. B. Wolff. Surface curvature and shape reconstruction from unknown multiple illumination and integrability. *Computer Vision and Image Understanding*, 65(2):347–359, 1997.
5. A. S. Georgiades, D. J. Kriegman, and P. N. Belhumeur. Illumination cones for recognition under variable lighting: Faces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 52–58, 1998.
6. H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *Journal of Optical Society of America A*, 11(11):3079–3089, 1994.
7. D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–212, 1997.
8. J. J. Koenderink and A. J. van Doorn. The generic bilinear calibration-estimation problem. *International Journal of Computer Vision*, 23(3):217–234, 1997.
9. S. Lin and S. W. Lee. Estimation of diffuse and specular appearance. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 855–860, 1999.
10. S. K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Journal of Robotics and Automation*, 6(4):418–431, 1990.
11. F. Solomon and K. Ikeuchi. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):449–454, 1996.
12. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. F. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Proc. Vision Algorithms: Theory and Practice. International Workshop on Vision Algorithms.*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372, 1999.
13. R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.
14. A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 158–164, 1997.
15. A.L. Yuille, J.M. Coughlan, and S. Konishi. The KGBR viewpoint-lighting ambiguity and its resolution by generic constraints. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 376–382, 2001.

## Appendix

Let  $\mathcal{S}$  be a unit sphere in  $\mathbb{R}^3$ . Let  $\mathbf{v}$  be a vector from  $\mathcal{S}$  and let  $\mathcal{S}^{1/2}$  be a set of unit vectors  $\mathbf{b}^S \in \mathcal{S}$  for which it holds that  $\mathbf{v} \cdot \mathbf{b}^S \geq 0$  (thus  $\mathcal{S}^{1/2}$  is a half-sphere, with  $\mathbf{v}$  being its axis). Vector  $\mathbf{v}$  represents viewing direction and vectors  $\mathbf{b}^S$  represent visible normals. For given normal  $\mathbf{b}^S$ , let  $\mathbf{s}$  denotes the light direction under which  $\mathbf{b}^S$  is specular (so that  $\mathbf{b}^S$  and  $\mathbf{s}$  is a specular pair).

We are asking the following question: if the normals are transformed as  $\mathbf{b}^S \mapsto \mathbf{A}\mathbf{b}^S$  and lights as  $\mathbf{s} \mapsto -\mathbf{A}^{-\top}\mathbf{s}$ , what are the only transformations that preserve the consistent viewpoint condition (4)?

First, we showed in Section 2 that this question is equivalent to asking what symmetric positively definite matrices  $\mathbf{P} = \mathbf{A}^\top \mathbf{A}$  exist such that

$$\alpha \mathbf{w} = 2(\mathbf{s} \cdot \mathbf{b}^S) \mathbf{P} \mathbf{b}^S - (\mathbf{b}^S \cdot \mathbf{P} \mathbf{b}^S) \mathbf{s}, \quad (8)$$



where  $\mathbf{w} \in \mathcal{S}$  is some vector which is fixed for all  $\mathbf{b}^S \in \mathcal{S}^{1/2}$ , and  $\alpha$  is a scaling constant (different for each specular pair).

Obviously, the necessary condition for the validity of (8) is that  $\forall \mathbf{b}^S \in \mathcal{S}^{1/2}$ :

$$\mathbf{w} \in \text{span}(\mathbf{P}\mathbf{b}^S, \mathbf{s}), \quad (9)$$

or, equivalently,

$$\mathbf{P}\mathbf{b}^S \in \text{span}(\mathbf{w}, \mathbf{s}). \quad (10)$$

But  $\mathbf{P}$  is symmetric and positively definite and thus its effect on  $\mathbf{b}^S$  represents anisotropic scaling in arbitrary three orthogonal directions. To see that, let us write the SVD decomposition of  $\mathbf{A}$  and  $\mathbf{P}$ :

$$\mathbf{A} = \mathbf{U}\text{diag}(\lambda_1, \lambda_2, \lambda_3)\mathbf{V}^\top \quad \lambda_1, \lambda_2, \lambda_3 > 0; \quad \mathbf{U}, \mathbf{V} \in O(3), \quad (11)$$

$$\mathbf{P} = \mathbf{V}\text{diag}(\lambda_1^2, \lambda_2^2, \lambda_3^2)\mathbf{V}^\top. \quad (12)$$

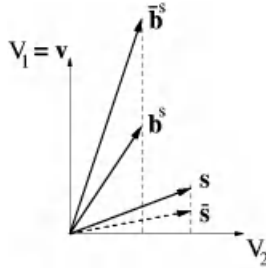
Thus  $\mathbf{P}$  scales along the direction of eigenvector  $\mathbf{V}_i$  ( $i$ -th column of  $\mathbf{V}$ ) by the respective  $\lambda_i$ .

Then normals  $\mathbf{b}_i^S = \pm \mathbf{V}_i$  (where  $\pm$  is properly selected according to whether  $\mathbf{V}_i$  is or is not in  $\mathcal{S}^{1/2}$ ) that are specular under corresponding lights  $\mathbf{s}_i$  are mapped onto themselves (up to a scale), and consequently  $\mathbf{w} \in \text{span}(\mathbf{b}_i^S, \mathbf{s}_i)$ ,  $i = 1 \dots 3$ . That implies that  $\mathbf{w} = \text{span}(\mathbf{b}_1^S, \mathbf{s}_1) \cap \text{span}(\mathbf{b}_2^S, \mathbf{s}_2) \cap \text{span}(\mathbf{b}_3^S, \mathbf{s}_3) = \mathbf{v}$ . But (10) must hold for all  $\mathbf{b}^S$  and we must therefore require  $\mathbf{P}\mathbf{b}^S \in \text{span}(\mathbf{v}, \mathbf{s}) = \text{span}(\mathbf{v}, \mathbf{b}^S)$ . The only way to arrange it is to align one of the scaling directions (say,  $\mathbf{V}_1$ ) with  $\mathbf{v}$ , and to set the scalings along the other two directions equal ( $\lambda_2 = \lambda_3$ ). Next, we show that all  $\lambda_i$ 's must be equal.

Let us complete (11) and (12) by writing decomposition of  $\mathbf{A}^{-\top}$  as

$$\mathbf{A}^{-\top} = \mathbf{U}\text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}\right)\mathbf{V}^\top. \quad (13)$$

Observe that a particular choice of the matrix  $\mathbf{U}$  has no effect on the validity of the consistent viewpoint constraint, since it only transforms both  $\mathbf{s}$  and  $\mathbf{b}^S$



**Fig. 9.** Scaling the  $\mathbf{V}_1$  component of normals by  $\lambda_1$  while scaling the same component of lights by  $1/\lambda_1$  results in that the new specular pair violates consistent viewpoint constraint if  $\lambda_1 \neq 1$ .

by a global orthogonal transformation (cf. (11)). It therefore suffices to consider the effect of transformations (11) and (13) on  $\mathbf{b}^S$  and  $\mathbf{s}$ , respectively, with  $\mathbf{U}$  set to identity.

Fig. 9 shows one pair of specular normal and corresponding light source before  $(\mathbf{b}^S, \mathbf{s})$  and after  $(\bar{\mathbf{b}}^S, \bar{\mathbf{s}})$  the photometric stereo transformation. Without the loss of generality we set  $\lambda_2 (= \lambda_3) = 1$ . The figure illustrates the fact that when, for example,  $\lambda_1 > 1$ , then the transformed normal makes a smaller angle with  $\mathbf{v}$  than the original normal, while the transformed light makes a greater angle with  $\mathbf{v}$  as compared with the original light. From that it follows that for transformed normals and lights  $\mathbf{v}$  is not consistent with the specular geometry condition unless it holds that  $\lambda_1 = \lambda_2 = \lambda_3$ .

Thus we have the result that  $\mathbf{P}$  may be only the scaled identity  $\mathbf{P} = \lambda^2 \mathbf{I}$ .

# Pairwise Clustering with Matrix Factorisation and the EM Algorithm

Antonio Robles-Kelly <sup>\*</sup> and Edwin R. Hancock

Department of Computer Science  
University of York, York YO10 5DD, UK  
{arobkell,erh}@cs.york.ac.uk

**Abstract.** In this paper we provide a direct link between the EM algorithm and matrix factorisation methods for grouping via pairwise clustering. We commence by placing the pairwise clustering process in the setting of the EM algorithm. We represent the clustering process using two sets of variables which need to be estimated. The first of these are cluster-membership indicators. The second are revised link-weights between pairs of nodes. We work with a model of the grouping process in which both sets of variables are drawn from a Bernoulli distribution. The main contribution in this paper is to show how the cluster-memberships may be estimated using the leading eigenvector of the revised link-weight matrices. We also establish convergence conditions for the resulting pairwise clustering process. The method is demonstrated on the problem of multiple moving object segmentation.

## 1 Introduction

Many problems in computer vision can be posed as ones of pairwise clustering. That is to say they involve grouping objects together based on their mutual similarity rather than their closeness to a cluster prototype. Such problems naturally lend themselves to a graph-theoretic treatment in which the objects to be clustered are represented using a weighted graph. Here the nodes represent the objects to be clustered and the edge-weights represent the strength of pairwise similarity relations between objects. The search for pairwise clusters can then be posed as the search for subgraphs. There are several ways in which such clusters can be located. These include classical search algorithms and optimization methods such as mean-field annealing. However, one of the most elegant solutions to the pairwise clustering problem comes from spectral graph theory. This is a branch of mathematics concerned with the characterising structural properties of graphs using the eigenvalues and eigenvectors of their adjacency matrices. The result which is of pivotal importance to the pairwise clustering problem is that the degree of bijectivity of a graph can be measured by the difference between the first and second eigenvalues of the adjacency matrix. This allows clusters to be removed sequentially from the set of nodes. Hence pairwise clustering can be recast in a matrix factorisation setting.

---

<sup>\*</sup> Supported by CONACYT, under grant No. 146475/151752.

Broadly speaking there are two distinct approaches to the pairwise clustering problem. The first of these involves the use of matrix factorisation methods. Here the pattern of pairwise distances are encoded in an affinity weight matrix. Several pairwise clustering algorithms have been developed to perform perceptual grouping and image segmentation tasks [15,13,14]. These draw their inspiration from spectral graph theory [5,8]. Perhaps the best known of these is the normalised cut method of Shi and Malik [15], which aims to iteratively optimise a graph-theoretic measure of pairwise cluster separation. Recently, Shi and Meilă [2] have analysed the convergence properties of the method using Markov chains. In related work, Tishby and Slonim [16] have developed a graph-theoretic method which exploits the stationarity and ergodicity properties of Markov chains defined on the affinity weights to locate clusters. Finally, Sarkar and Boyer [14] have a non-iterative method which use the same-sign positive eigenvectors of the affinity matrix to define object clusters.

The second approach to the pairwise clustering problem adopts an information theoretic framework. One of the best known methods here is that of Hofmann and Buhmann [9], which uses mean-field theory to update a set of pairwise cluster indicators. In related work, Gdalyahu, Weinshall and Werman [19] use a stochastic sampling method.

In a recent paper, we have described a new pairwise clustering method which combines ideas from spectral graph theory and information theory [13]. Specifically, we have developed an EM-like method for maximum likelihood pairwise clustering. The development of this method commences by assuming that the elements of the affinity matrix, or link weights, follow a Bernoulli distribution. The novelty of the method resides in the fact that the link-weights are updated in addition to the cluster membership variables.

The aims in this paper are three-fold. First, we place the method in a more rigorous EM setting. Second, we show how cluster membership variables can be computed using the coefficients of the leading eigenvector of the affinity matrix. Thirdly, we establish the convergence conditions when the cluster memberships are updated in this way.

## 2 Maximum Likelihood Framework

We pose the problem of perceptual grouping as that of finding the pairwise clusters which exist within a set of objects segmented from raw image data. These objects may be point-features such as corners, lines, curves or regions. However, in this paper we focus on the specific problem independent moving object segmentation from image sequences.

We abstract the problem in the following way. The raw perceptual entities are indexed using the set  $V$ . Our aim is to assign each node to one of a set of pairwise clusters which are indexed by the set  $\Omega$ . To represent the state of organisation of the perceptual relation graph, we introduce some indicator variables. First, we introduce a cluster membership indicator which is unity if the node  $i$  belongs

to the perceptual cluster  $\omega \in \Omega$  and is zero otherwise, i.e.

$$s_{i\omega} = \begin{cases} 1 & \text{if } i \in \omega \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The second model ingredient is the link-weight  $A_{i,j}$  between distinct pairs of nodes  $(i, j) \in V \times V - \{(i, i) | i \in V\}$ . When the link-weights become binary in nature, they convey the following meaning

$$A_{ij} = \begin{cases} 1 & \text{if } i \in \omega \text{ and } j \in \omega \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

When the link-weights satisfy the above condition, then the different clusters represent disjoint subgraphs.

Our aim is to find the cluster membership variables and the link weights which partition the set of raw perceptual entities into disjoint pairwise clusters. We commence by assuming that there are putative edges between each distinct pair of nodes  $(i, j)$  belonging to the Cartesian self-product  $\Phi = V \times V - \{(i, i) | i \in V\}$ . Further suppose that  $p(A_{ij})$  is the probability density for the link weight appearing on the pair of nodes  $(i, j) \in \Phi$ . Our aim is to locate disjoint subgraphs by updating the link weights until they are either zero or unity. Under the assumption that the link-weights on different pairs of nodes are independent of one-another, then the likelihood function for the observed arrangement of perceptual entities can be factorised over the set of putative edges as

$$P(A) = \prod_{(i,j) \in \Phi} P(A_{ij}) \quad (3)$$

We are interested in partitioning the set of perceptual entities into pairwise clusters using the link weights between them. We must therefore entertain the possibility that each of the Cartesian pairs appearing under the above product, which represent putative perceptual relations, may belong to each of the pairwise clusters indexed by the set  $\Omega$ . To make this uncertainty of association explicit, we construct a mixture model over the perceptual clusters and write

$$P(A_{ij}) = \sum_{\omega \in \Omega} P(A_{ij}|\omega)P(\omega) \quad (4)$$

According to this mixture model,  $P(A_{ij}|\omega)$  is the probability that the nodes  $i$  and  $j$  are connected by an edge with link weight  $A_{ij}$  which falls within the perceptual cluster indexed  $\omega$ . The total probability mass associated with the cluster indexed  $\omega$  is  $P(\omega)$ . In most of our experiments, we will assume that there are only two such sets of nodes; those that represent a foreground arrangement, and those that represent background clutter. However, for generality we proceed under the assumption that there are an arbitrary number of perceptual clusters. As a result, the probability of the observed set of perceptual entities is

$$P(A) = \prod_{(i,j) \in \Phi} \sum_{\omega \in \Omega} P(A_{ij}|\omega)P(\omega) \quad (5)$$

To proceed, we require a model of probability distribution for the link-weights. Here we adopt a model in which the observed link structure of the pairwise clusters arises through a Bernoulli distribution. The parameter of this distribution is the link-probability  $A_{i,j}$ . The idea behind this model is that any pair of nodes  $i$  and  $j$  may connect to each with a link. This link is treated as a Bernoulli variable. The probability that this link is the correct is  $A_{i,j}$  while the probability that it is in error is  $1 - A_{i,j}$ . To gauge the correctness of the link, we check whether the nodes  $i$  and  $j$  belong to the same pairwise cluster. To test for cluster-consistency we make use of the quantity  $s_{i\omega}s_{j\omega}$ . This is unity if both nodes belong to the same cluster and is zero otherwise. Using this switching property, the Bernoulli distribution becomes

$$p(A_{ij}|\omega) = A_{i,j}^{s_{i\omega}s_{j\omega}} (1 - A_{i,j})^{1-s_{i\omega}s_{j\omega}} \quad (6)$$

This distribution takes on its largest values when either the link weight  $A_{ij}$  is unity and  $s_{i\omega} = s_{j\omega} = 1$ , or if the link weight  $A_{i,j} = 0$  and  $s_{i\omega} = s_{j\omega} = 0$ .

### 3 Expectation-Maximisation

Our aim is to find the cluster-membership weights and the link-weights which maximize the likelihood function appearing in Equation (5). One way to locate the maximum likelihood perceptual relation graph is to update the binary cluster and edge indicators. This could be effected using a number of optimisation methods including simulated annealing and Markov Chain Monte Carlo. However, here we use the apparatus of the EM algorithm originally developed by Dempster, Laird and Rubin [1]. Our reason for doing this is that the cluster-membership variables  $s_{i\omega}$  must be regarded as quantities whose distribution is governed by the link weights  $A_{ij}$ . Since at the outset we know neither the associations between nodes and clusters nor the strength of the link weights within clusters, this information must be treated as hidden data. In other words, we must use the EM algorithm to estimate them.

The idea underpinning the EM algorithm is to recover maximum likelihood solutions to problems involving missing or hidden data by iterating between two computational steps. In the E (or expectation) step we estimate the a posteriori probabilities of the hidden data using maximum likelihood parameters recovered in the preceding maximisation (M) step. The M-step in-turn aims to recover the parameters which maximise the expected value of the log-likelihood function. It is the available a posteriori probabilities from the E-step which allows the weighting of log-likelihood required in the maximisation-step.

#### 3.1 Expected log-Likelihood Function

For the likelihood function appearing in Equation (5), the expected log-likelihood function is defined to be

$$Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega \in \Omega} \sum_{(i,j) \in \Phi} P(w|A_{ij}^{(n)}) \ln p(A_{ij}^{(n+1)}|\omega) \quad (7)$$

where  $p(A_{ij}^{(n+1)}|\omega)$  is the probability distribution for the link-weights at iteration  $n + 1$  and  $P(w|A_{ij}^{(n)})$  is the a posteriori probability that the pair of nodes with link weight  $A_{ij}^{(n)}$  belong to the cluster indexed  $\omega$  at iteration  $n$  of the algorithm. When the probability distribution function from Equation (6) is substituted, then the expected log-likelihood function becomes

$$Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega \in \Omega} \sum_{(i,j) \in \Phi} \zeta_{i,j,\omega}^{(n)} \left\{ s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)} \ln A_{ij}^{(n+1)} + (1 - s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)}) \ln(1 - A_{ij}^{(n+1)}) \right\} \quad (8)$$

where we have used the shorthand  $\zeta_{i,j,\omega}^{(n)} = P(w|A_{ij}^{(n)})$  for the a posteriori cluster membership probabilities. After some algebra to collect terms, the expected log-likelihood function simplifies to

$$Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega \in \Omega} \sum_{(i,j) \in \Phi} \zeta_{i,j,\omega}^{(n)} \left\{ s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)} \ln \frac{A_{ij}^{(n+1)}}{1 - A_{ij}^{(n+1)}} + \ln(1 - A_{ij}^{(n+1)}) \right\} \quad (9)$$

### 3.2 Maximisation

There are two sets of variables that need to be updated in the M-step of the algorithm. These are the link-weights and the cluster membership indicators.

**Link-Weights.** In the maximisation step of the algorithm we update the link-weight matrix. The updated link-weights are found by computing the derivatives of the expected log-likelihood function

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial A_{ij}^{(n+1)}} = \sum_{\omega \in \Omega} \zeta_{i,j,\omega}^{(n)} \left\{ s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)} \frac{1}{A_{ij}^{(n+1)}(1 - A_{ij}^{(n+1)})} - \frac{1}{1 - A_{ij}^{(n+1)}} \right\} \quad (10)$$

and solving the saddle-point equations  $\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial A_{ij}^{(n+1)}} = 0$ . As a result the updated link-weights are given by

$$A_{ij}^{(n+1)} = \frac{\sum_{\omega \in \Omega} \zeta_{i,j,\omega}^{(n)} s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)}}{\sum_{\omega \in \Omega} \zeta_{i,j,\omega}^{(n)}} \quad (11)$$

In other words, the link-weight for the pair of nodes  $(i, j)$  is simply the average of the product of individual node cluster memberships over the different perceptual clusters. Since each node is associated with a unique cluster, this means that the updated affinity matrix is composed of non-overlapping blocks. Moreover, the link-weights are guaranteed to be in the interval  $[0, 1]$ . Finally, it is important to note that the updating of the link-weights is a unique feature of our algorithm which distinguishes it from the pairwise clustering methods of Hoffman and Buhmann and Shi and Malik.

**Cluster memberships.** To update the cluster-membership variables we have used a gradient-based method. We commence by computing the derivatives of the expected log-likelihood function with respect to the cluster-membership variable

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}} = \sum_{j \in V_\omega} \zeta_{i,j,\omega} s_{j\omega}^{(n)} \ln \frac{A_{ij}^{(n+1)}}{1 - A_{ij}^{(n+1)}} \quad (12)$$

Since the associated saddle-point equations are not tractable in closed form, we use the soft-assign ansatz of Bridle [4] to update the cluster membership assignment variables. This involves exponentiating the partial derivatives of the expected log-likelihood function in the following manner

$$s_{i\omega}^{(n+1)} = \frac{\exp \left[ \frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n)}} \right]}{\sum_{i \in V_\omega} \exp \left[ \frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n)}} \right]} \quad (13)$$

As a result the update equation for the cluster membership indicator variables is

$$s_{i\omega}^{(n+1)} = \frac{\exp \left[ \sum_{j \in V_\omega} \zeta_{i,j,\omega} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} \right]}{\sum_{i \in V_\omega} \exp \left[ \sum_{j \in V_\omega} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} \right]} \quad (14)$$

After simplifying the argument of the exponential, the update formula reduces to

$$s_{i\omega}^{(n+1)} = \frac{\prod_{j \in V_\omega} \left\{ \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} \right\}^{\zeta_{i,j,\omega} s_{j\omega}^{(n)}}}{\sum_{i \in V_\omega} \prod_{j \in V_\omega} \left\{ \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} \right\}^{\zeta_{i,j,\omega} s_{j\omega}^{(n)}}} \quad (15)$$

It is worth pausing to consider the structure of this update equation. First, the updated link weights are an exponential function of the current ones. Second, the exponential constant is greater than unity, i.e. there is re-enforcement of the cluster memberships, provided that  $A_{i,j}^{(n+1)} > \frac{1}{2}$ .

We can take this analysis one step further and establish a link with the eigenvectors of the updated adjacency matrix. To this end we introduce the matrix  $T^{(n+1)}$  whose element with row  $i$  and column  $j$  is

$$T_{i,j}^{(n+1)} = \zeta_{i,j,\omega} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}}$$

We also define the vector  $\underline{s}_\omega^{(n)} = (s_{1\omega}^{(n)}, s_{2\omega}^{(n)}, \dots)^T$ . With this notation we can turn our attention to the argument of the exponential appearing in Equation (14)



and write

$$\sum_{j \in V_\omega} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} = \left( T^{(n+1)} \underline{s}_\omega^{(n)} \right)_i \quad (16)$$

In other words, the argument is simply the  $i$ th component of the vector obtained by the matrix multiplication  $T^{(n+1)} \underline{s}_\omega^{(n)}$ .

Next, consider the case when the vector  $\underline{s}_\omega^{(n)}$  is an eigenvector of the matrix  $T^{(n+1)}$ . The eigenvector equation for the matrix  $T^{(n+1)}$  is  $T^{(n+1)} \underline{z}_\omega^{(n)} = \lambda_\omega \underline{z}_\omega^{(n)}$ , where  $\lambda_\omega$  is the  $\omega^{th}$  eigenvalue and  $\underline{z}_\omega$  is the corresponding eigenvector. Hence when the vector of cluster memberships  $\underline{s}_\omega^{(n)}$  is an eigenvector of  $T^{(n+1)}$ , then we can write  $(T^{(n+1)} \underline{s}_\omega^{(n)})_i = \underline{z}_\omega(i)$ , where  $\underline{z}_\omega(i)$  is the  $i$ th component of the vector  $\underline{z}_\omega$ . If this is the case, then we can identify the pairwise clusters with the eigenmodes of  $\ln T^{(n+1)}$ . Using the shorthand  $\hat{\lambda}_\omega = \ln \lambda_\omega$ , the update equation becomes

$$s_{i\omega}^{(n+1)} = \frac{\exp \left[ \lambda_\omega z_\omega^{(n)}(i) \right]}{\sum_{i \in V_\omega} \exp \left[ \lambda_\omega z_\omega^{(n)}(i) \right]} = \frac{\hat{\lambda}_\omega^{z_\omega^{(n)}(i)}}{\sum_{i \in V_\omega} \hat{\lambda}_\omega^{z_\omega^{(n)}(i)}} \quad (17)$$

Since the directions of the eigenvectors of the matrix  $\ln T$  and  $T$  are identical [7], the updated cluster membership variables can be computed directly from the matrix  $A^{(n+1)}$ . Moreover, by inspection is easy to see that the updated cluster membership variables are the coefficients of the eigenvectors of  $\ln T^{(n+1)}$ . In this way, by computing the eigenmodes of the current link-weight matrix, we can update the individual cluster membership indicators.

### 3.3 Convergence Analysis

In this section we provide some analysis of the convergence properties of the grouping algorithm when the cluster memberships are updated using the eigenvector method outlined in the previous section. We are interested in the relationship between this modal analysis and the updated cluster membership variables. Using the update formulae for the link-weight matrix and the cluster membership indicators given in Equations (12) and (18), it is a straightforward matter to show that the corresponding updated log-likelihood function is given by

$$\begin{aligned} Q(A^{(n+1)} | A^{(n)}) = & \sum_{\omega \in \Omega} \sum_{(i,j) \in |V_\omega \times V_\omega|} \left\{ T_{ij}^{(n+1)} \lambda_\omega^{(n+1) z_\omega(i)^{(n+1)} + z_\omega(j)^{(n+1)}} \right. \\ & \left( \frac{1}{\sum_{i' \in V_\omega} \lambda_\omega^{(n+1) z_\omega(i')^{(n+1)}} \sum_{j' \in V_\omega} \lambda_\omega^{(n+1) z_\omega(j')^{(n+1)}}} \right) \\ & \left. + \ln(1 - A_{ij}^{(n+1)}) \right\} \end{aligned} \quad (18)$$

We would like to understand the conditions under which the likelihood is maximised by the update process. We hence compute the partial derivative of  $Q$  with respect to  $\lambda_\omega^{(n+1)}$ . After collecting terms and some algebra we find

$$\frac{\partial \hat{Q}(A^{(n+1)}|A^{(n)})}{\partial \lambda_\omega^{(n+1)}} = \frac{\lambda_\omega^{(n+1)z_\omega(i)^{(n)}+z_\omega(j)^{(n)}}}{\lambda_\omega^{(n+1)} \left( \sum_{i' \in V_\omega} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}} \right)^2} \left\{ z_\omega(i)^{(n)} + z_\omega(j)^{(n)} - 2 \frac{\sum_{i' \in V_\omega} z_\omega(i')^{(n)} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}}}{\sum_{i' \in V_\omega} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}}} \right\} \quad (19)$$

The maximum value of the log-likelihood occurs where the partial derivative is zero. This condition is satisfied when

$$z_\omega(i)^{(n)} + z_\omega(j)^{(n)} = 2 \frac{\sum_{i' \in V_\omega} z_\omega(i')^{(n)} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}}}{\sum_{i' \in V_\omega} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}}} \quad (20)$$

Unfortunately, this condition is not always guaranteed to be satisfied. However, from Equation 20 we can conclude that the following will always be the best approximation

$$\lambda_\omega^{(n+1)z_i^{(n)}+z_j^{(n)}} << \lambda_\omega^{(n+1)} \left( \sum_{i' \in V_\omega} \lambda_\omega^{(n+1)z_{i'}^{(n)}} \right)^2 \quad (21)$$

If  $T^{(n+1)}$  is a non-negative irreducible symmetric matrix  $T^{(n+1)}$ , then the coefficients of the eigenvector  $\underline{z}_*^{(n)}$  associated with the largest eigenvalue  $\lambda^*$  are each positive [17]. As a result, the quantity  $\sum_{i' \in V_\omega} \lambda_\omega^{(n+1)z_\omega(i')^{(n)}}$  will be maximized when  $\lambda_\omega^{(n+1)}$  is maximum. Hence,  $Q(A^{(n+1)}|A^{(n)})$  will be maximized by the first (maximum) eigenvalue of  $T^{(n+1)}$ .

### 3.4 Expectation

The a posteriori probabilities are updated in the expectation step of the algorithm. The current estimates of the parameters  $s_{i\omega}^{(n)}$  and  $A_{ij}^{(n)}$  are used to compute the probability densities  $p(A_{ij}^{(n)}|\omega)$  and the a posteriori probabilities are updated using the formula

$$P(\omega|A_{ij}^{(n)}) = \frac{p(A_{ij}^{(n)}|\omega)\alpha^{(n)}(\omega)}{\sum_{\omega \in \Omega} p(A_{ij}^{(n)}|\omega)\alpha^{(n)}(\omega)} \quad (22)$$

where  $\alpha^{(n)}(\omega)$  is the available estimate of the class-prior  $P(\omega)$ . This is computed using the formula

$$\alpha^{(n)}(\omega) = \frac{1}{|V|^2} \sum_{(i,j) \in \Phi} P(\omega|A_{ij}^{(n)}) \quad (23)$$

Upon substituting for the probability density from Equation (6), the updated a posteriori probabilities are given by

$$P(\omega|A_{ij}^{(n)}) = \zeta_{i,j,\omega}^{(n+1)} = \frac{A_{i,j}^{(n)} s_{i\omega}^{(n)} s_{j\omega}^{(n)} (1 - A_{i,j}^{(n)})^{1-s_{i\omega}^{(n)} s_{j\omega}^{(n)}} \alpha^{(n)}(\omega)}{\sum_{(i,j) \in \Phi} A_{i,j}^{(n)} s_{i\omega}^{(n)} s_{j\omega}^{(n)} (1 - A_{i,j}^{(n)})^{1-s_{i\omega}^{(n)} s_{j\omega}^{(n)}} \alpha^{(n)}(\omega)} \quad (24)$$

### 3.5 Selecting the Number of Modes

One of the practical difficulties associated with using a mixture model is that of selecting the number of components. There are several well established ways of addressing this problem. These frequently involve removing or splitting components so as to optimise a measure of model-order complexity [11,18,10]. There are several possible choices of utility measures including the minimum description length [12] and the Akaike criterion.

However, here we take a different route and use the modal structure of the initial affinity matrix to set the number of mixing components. Here we use a result due to Sarkar and Boyer [14] who have shown how matrix factorisation methods can be used to locate the set of edges which partition the nodes into distinct perceptual clusters. One way of viewing this is as the search for the permutation matrix which re-orders the elements of  $A$  into non-overlapping blocks. Unfortunately, when the elements of the matrix  $A$  are not binary in nature, then this is not a straightforward task. However, Sarkar and Boyer [14] have shown how the positive eigenvectors of the matrix of link-weights can be used to assign nodes to perceptual clusters. Using the Rayleigh-Ritz theorem, they observe that the scalar quantity  $\underline{x}^t A^{(0)} \underline{x}$ , where  $A^{(0)}$  is the initial weighted adjacency matrix, is maximised when  $\underline{x}$  is the leading eigenvector of  $A$ . Moreover, each of the subdominant eigenvectors corresponds to a disjoint cluster. They confine their attention to the same-sign positive eigenvectors. If a component of a positive eigenvector is non-zero, then the corresponding node belongs to the cluster associated with the eigenmodes of the weighted adjacency matrix. The eigenvalues  $\lambda_1, \lambda_2, \dots$  of  $A^{(0)}$  are the solutions of the equation  $|A^{(0)} - \lambda I| = 0$  where  $I$  is the  $|V| \times |V|$  identity matrix. The corresponding eigenvectors  $\underline{x}_{\lambda_1}, \underline{x}_{\lambda_2}, \dots$  are found by solving the equation  $A^{(0)} \underline{x}_{\lambda_i} = \lambda_i \underline{x}_{\lambda_i}$ . Let the set of positive same-sign eigenvectors be represented by  $\Omega = \{\omega | \lambda_\omega > 0 \wedge [(\underline{x}_\omega^*(i) > 0 \forall i) \vee \underline{x}_\omega^*(i) < 0 \forall i]\}$ . Since the positive eigenvectors are orthogonal, this means that there is only one value of  $\omega$  for which  $\underline{x}_\omega^*(i) \neq 0$ . We denote the set of nodes assigned to the cluster with modal index  $\omega$  as  $V_\omega = \{i | \underline{x}_\omega^*(i) \neq 0\}$ . Hence each positive same-sign eigenvector is associated with a distinct mixing component. This is done using the magnitudes of the modal co-efficients and we set

$$s_{i\omega}^{(0)} = \frac{|\underline{x}_\omega^*(i)|}{\sum_{i \in V_\omega} |\underline{x}_\omega^*(i)|} \quad (25)$$

We use the eigenvectors of the initial affinity matrix to initialise the cluster membership variables.

## 4 Motion Segmentation

The application of our pairwise clustering method focusses on the segmentation of independently moving objects from image sequences. The motion vectors used in our analysis have been computed using a single resolution block matching algorithm [6]. The method measures the similarity of motion blocks using spatial correlation and uses predictive search to efficiently compute block-correspondences in different frames. The block matching algorithm assumes that the translational motion from frame to frame is constant. The current frame is divided into blocks that will be compared with the next frame in order to find the displaced coordinates of the corresponding block within the search area of the reference frame. Since the computational complexity is much lower than the optical flow equation and the pel-recursive methods, block matching has been widely adopted as a standard for video coding and hence it provides a good starting point.

However, the drawback of the single resolution block-matching scheme is that while the high resolution field of motion vectors obtained with small block sizes captures fine detail, it is susceptible to noise. At low resolution, i.e. for large block sizes, the field of motion vectors is less noisy but the fine structure is lost. To strike a compromise between low-resolution noise suppression and high resolution recovery of fine detail, there have been several attempts to develop multi-resolution block matching algorithms. These methods have provided good predictive performance and also improvements in speed. However, one of the major problems with the multi-resolution block matching method is that random motions can have a significant degradational effect on the estimated motion field. For these reasons, we have used a single high-resolution block matching algorithm to estimate the raw motion field. This potentially noisy information is refined in the motion segmentation step, where we exploit hierarchical information.

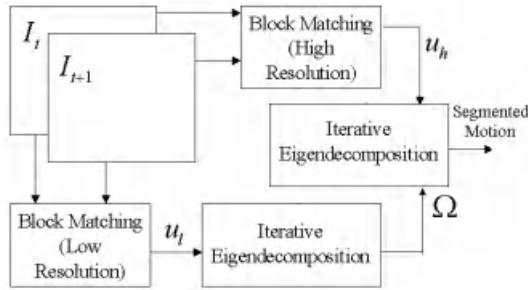
We pose the problem of grouping motion blocks into coherent moving objects as that of finding pairwise clusters. The 2D velocity vectors for the extracted motion blocks are characterised using a matrix of pairwise similarity weights. Suppose that  $\hat{\mathbf{n}}_i$  and  $\hat{\mathbf{n}}_j$  are the unit motion vectors for the blocks indexed  $i$  and  $j$ . The elements of this weight matrix are given by

$$A_{i,j}^{(0)} = \begin{cases} \frac{1}{2}(1 + \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

### 4.1 Hierarchical Motion Segmentation

As mentioned earlier, we use a single-level high-resolution block-matching method to estimate the motion field. The resulting field of motion vectors is therefore likely to be noisy. To control the effects of motion-vector noise, we have developed a multi-resolution extension to the clustering approach described above.

The adopted approach is as follows.



**Fig. 1.** Motion segmentation system.

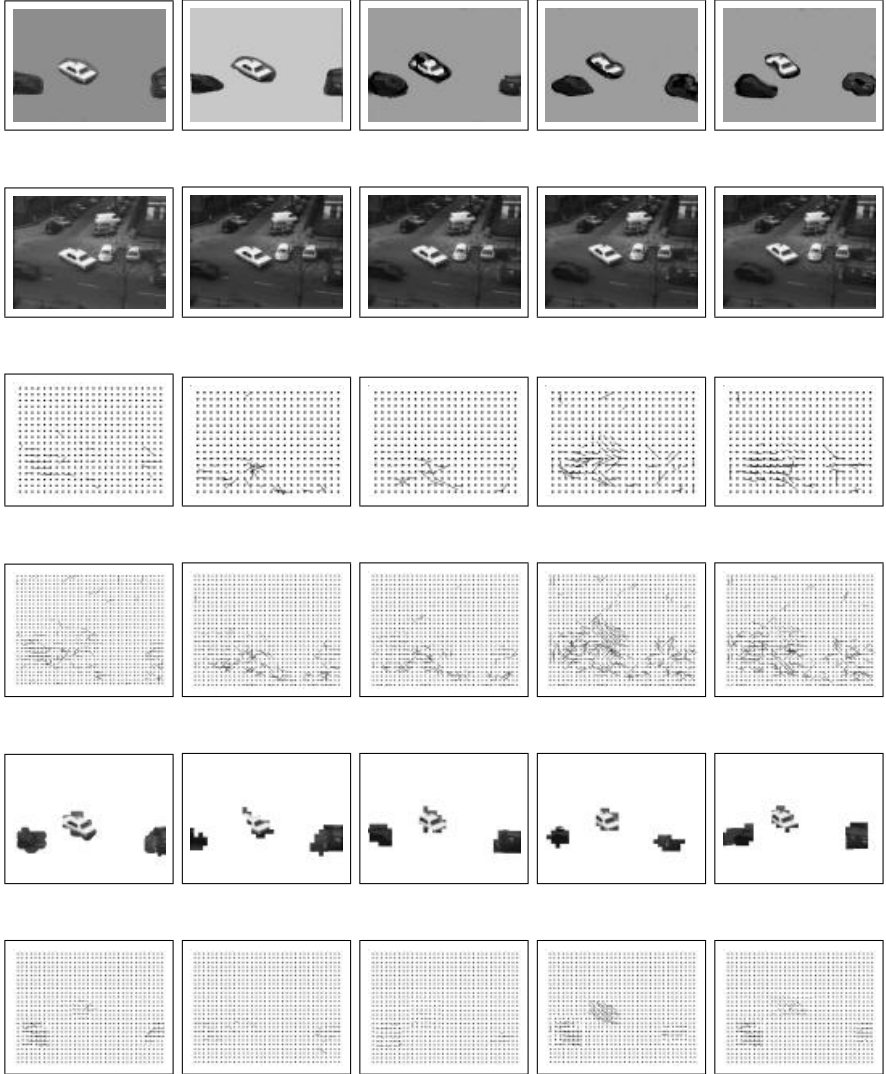
- We obtain the a high resolution field of motion vectors  $U_H$  using blocks of size  $k$ -pixels and a low-resolution motion field  $U_L$  using blocks of size  $2k$  pixels.
- We apply our clustering algorithm to the low resolution motion field  $U_L$  numbering the  $N_L$  clusters detected.
- We make a second application of our clustering algorithm to the high-resolution motion field  $U_H$ . Here we select only the first  $N_L$  eigenvalues of the motion-vector similarity matrix as cluster centres.

In this way we successively perform the motion estimation at low and high resolution. The number of clusters detected at low resolution is used to constrain the number of permissible high resolution clusters. This allows the high-resolution clustering process to deal with fine detail motion fields without succumbing to noise. There is scope to extend the method and develop a pyramidal segmentation strategy. The structure of the hierarchical system can be seen in Figure 1.

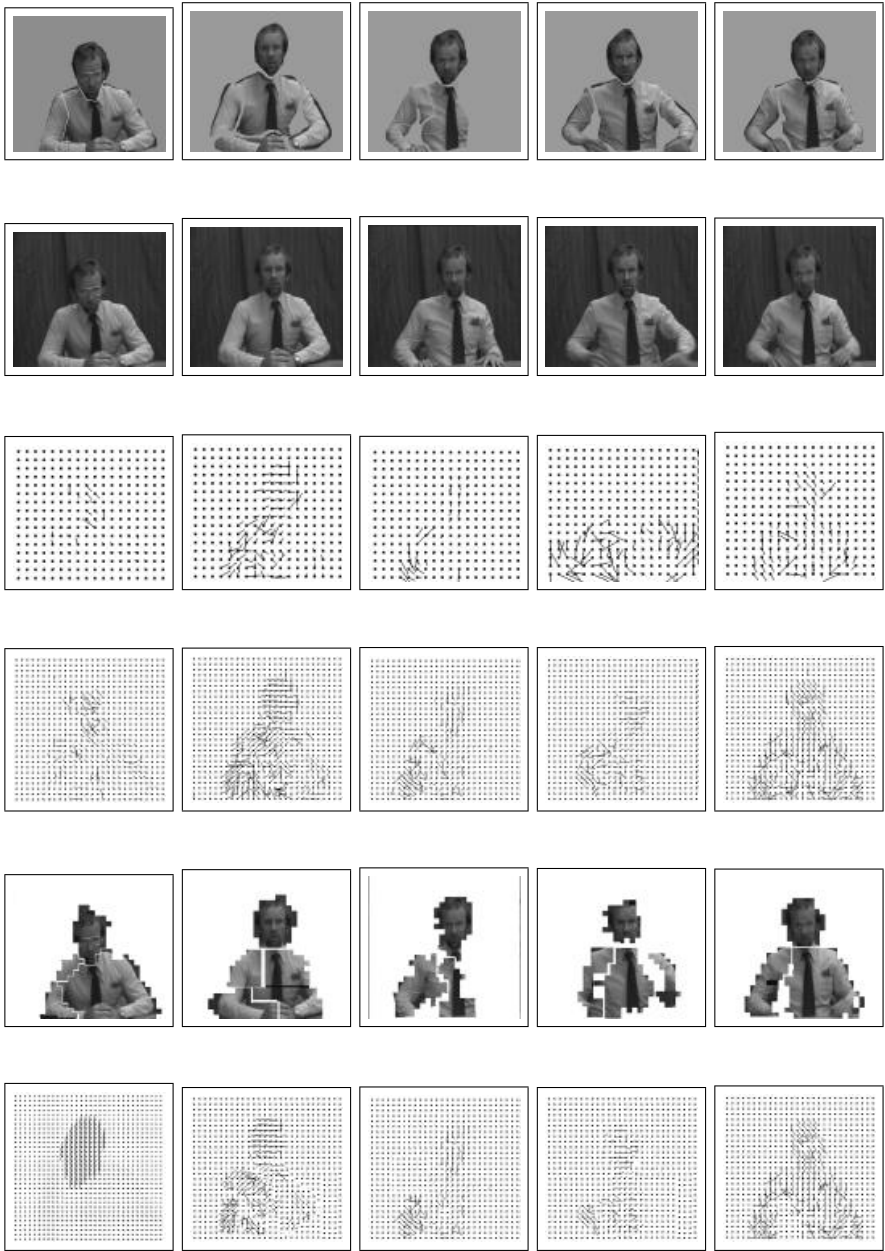
## 4.2 Motion Experiments

We have conducted experiments on motion sequences with known ground truth. In Figure 2 we show some results obtained with five frames of the well-known “Hamburg Taxi” sequence. The top row shows the hand-labelled ground-truth segmentation for the motion sequence. The second row shows the corresponding image frames from the motion sequence. In the third and fourth rows we respectively show the low resolution and high resolution block motion vectors. The low-resolution uses  $16 \times 16$  pixel blocks to perform motion correspondence and compute the motion vectors; for the high resolution motion field the block size is  $8 \times 8$  pixels. The fifth row shows the moving objects segmented from the motion field using pairwise clustering. In each frame there are 3 clusters which match closely to the ground truth data shown. It is interesting to note that the

results are comparable to those reported in [3] where a 5 dimensional feature vector and a neural network was used. The proposed algorithm converges in an average of four iterations.



**Fig. 2.** Top row: ground truth for the 1st, 4th, 8th, 12th and 16th frame of the "Hamburg Taxi" sequence; Second row: original frames ; Third and fourth row: resolution motion maps; Fifth row: Final motion segmentation; Bottom row: smoothed motion maps.



**Fig. 3.** Top row: ground truth for the 1st, 5th, 10th, 15th and 20th frame of the "Trevor White" sequence; Second row: original frames ; Third and fourth row: motion maps; Fifth row: Final motion segmentation; Bottom row: smoothed motion maps.

**Table 1.** Error percentage for the two image sequences.

Sequence	Cluster	% of Error	Sequence	Cluster	% of Error
Trevor White	Right Arm	8 %	Ham. Taxi	Taxi	4 %
Trevor White	Chest	6 %	Ham. Taxi	Left Car	3 %
Trevor White	Head	12 %	Ham. Taxi	Right Car	10 %

Figure 3 repeats these experiments for the “Trevor White” sequence. The sequence of rows is the same as in in Figure 3. Here the block sizes are respectively  $24 \times 24$  and  $12 \times 12$  pixels. There are three motion clusters which correspond to the head, the right arm, and the chest plus left arm. These clusters again match closely to the ground-truth data.

In Table 1 we provide a more quantitative analysis of these results. The table lists the fraction of the pixels in each region of the ground truth data which are misassigned by the clustering algorithm. The best results are obtained for the chest-region, the taxi and the far-left car, where the error rate is a few percent. For the far-right car and the head of the Trevor White, the error rates are about 10%. The problems with the far-right car probably relate to the fact that it is close to the periphery of the image.

## 5 Conclusions

In this paper, we have developed an Expectation-maximization framework for pairwise clustering. The method commences from a specification of the pairwise clustering problem in terms of a matrix of link-weights and a set of cluster membership indicators. The Expected log-likelihood function underpinning our method is developed under the assumption that the cluster membership indicators are random variables which are drawn from a Bernoulli distributions. The parameter of the Benoulli distributions are the link-weights. Based on this model, we develop an iterative process for updating the link-weights and the cluster membership indicators in interleaved steps.

To understand the relationships between our probabilistic method and the graph-spectral approach to pairwise clustering, we present a variational analysis for the expected log-likelihood function. This reveals two important results. First, we demonstrate a relationship between the updated cluster membership variables resulting from our variational analysis and the eigenvectors of the link-weight matrix. Second, we show that the log-likelihood function is maximised by the leading eigenvector of the link-weight matrix.

There are a number of ways in which the work presented in this paper can be extended and improved. First, we intend to investigate alternatives to the Bernoulli model of the clustering process. Second, our present method does not facilitate data-closeness between the final arrangement of clusters and the raw data. Our future work will therefore focus on developing a clustering process which minimizes the Kullback-Leibler divergence between the initial matrix of link-weights and the final arrangement of pairwise clusters.



## References

1. A. Dempster, N. Laird and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977.
2. Marina Meilă and Jianbo Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems 13*, pages 873–879. MIT Press, 2001.
3. A. G. Bors and I. Pitas. Optical flow estimation and moving object segmentation based on median radial basis function network. *IEEE Trans. on Image Processing*, 7(5):693–702, 1998.
4. J. S. Bridle. Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters. In *NIPS 2*, pages 211–217, 1990.
5. Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
6. J. S. Shyn C. H. Hsieh, P. C. Lu and E. H. Lu. Motion estimation algorithm using inter-block correlation. *IEE Electron. Lett.*, 26(5):276–277, 1990.
7. L. Dieci. Considerations on computing real logarithms of matrices, hamiltonian logarithms and skew-symmetric logarithms. *Linear Algebra and its Applications*, 244:35–54, 1996.
8. M. Doob D. Cvetković and H. Sachs. *Spectra of Graphs: Theory and Application*. Academic Press, 1980.
9. T. Hofmann and M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.
10. J. M. Leitao M. Figueiredo and A. K. Jain. On fitting mixture models. In *Proceedings of the Second Int. Workshop on Energy Minimization Methods in Comp. Vis. and Pattern Recognition*, number 1654 in Lecture Notes in Comp. Science, pages 54–67, 1999.
11. Z. Ghahramani N. Ueda, R. Nakano and G. E. Hinton. Smem algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.
12. J. Risanen. *Stochastic Complexity in Statistical Enquiry*. World Scientific, 1989.
13. A. Robles-Kelly and E. R. Hancock. A maximum likelihood framework for iterative eigendecomposition. In *Proc. of the IEEE International Conference on Computer Vision*, 2001.
14. S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998.
15. J. Shi and J. Malik. Normalized cuts and image segmentations. In *Proc. IEEE CVPR*, pages 731–737, 1997.
16. Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 640–646. MIT Press, 2001.
17. R. S. Varga. *Matrix Iterative Analysis*. Springer, second edition, 2000.
18. N. Vlassis and A. Likas. A kurtosis-based dynamic approach to gaussian mixture modeling. *IEEE Trans. in Systems, Man and Cybernetics*, 29(4):393 – 399, 1999.
19. D. Weinshall Y. Gdalyahu and M. Werman. A randomized algorithm for pairwise clustering. In *Advances in Neural Information Processing Systems 11*, pages 424–430. MIT Press, 1999.

# Shape Priors for Level Set Representations

Mikael Rousson<sup>1\*</sup> and Nikos Paragios<sup>2</sup>

<sup>1</sup> I.N.R.I.A.

B.P. 93, 2004 Route de Lucioles,  
06902 Sophia Antipolis Cedex, France  
[Mikael.Rousson@sophia.inria.fr](mailto:Mikael.Rousson@sophia.inria.fr)

<sup>2</sup> Imaging and Visualization Department  
Siemens Corporate Research  
755 College Road East  
Princeton, NJ 08540, USA  
[nikos@scr.siemens.com](mailto:nikos@scr.siemens.com)

**Abstract.** Level Set Representations, the pioneering framework introduced by Osher and Sethian [14] is the most common choice for the implementation of variational frameworks in Computer Vision since it is implicit, intrinsic, parameter and topology free. However, many Computer vision applications refer to entities with physical meanings that follow a shape form with a certain degree of variability. In this paper, we propose a novel energetic form to introduce shape constraints to level set representations. This formulation exploits all advantages of these representations resulting on a very elegant approach that can deal with a large number of parametric as well as continuous transformations. Furthermore, it can be combined with existing well known level set-based segmentation approaches leading to paradigms that can deal with noisy, occluded and missing or physically corrupted data. Encouraging experimental results are obtained using synthetic and real images.

## 1 Introduction

Level Set [14,13,19] and variational methods are increasingly considered by the vision community [17]. The application domain is wide and not restricted to image segmentation, restoration, inpainting, tracking, shape from shading, 3D reconstruction [7], medical image segmentation [11], etc. These techniques have been exhaustively studied and also applied to other scientific domains like geometry, robotics, fluids, semiconductors designing, etc. [19].

Most of the mentioned applications share a common concern, tracking moving interfaces. Level Set representations are well suited computational methods to perform this task. They can be used to any dimension (curves, surfaces, hyper-surfaces, ...), are parameter free and can change naturally the topology. Moreover, they provide a natural way to estimate the geometric properties of the evolving interface.

Furthermore, they can deal with non-rigid objects and motions, since they refer to very local characteristics and can deform an interface pixel-wise. Opposite to that, they

---

\* The authors is currently with the Computer Vision and Robotics Group (RobotVis) at INRIA Sophia Antipolis, France.

have a poor performance compared with parametric models when solid/rigid motions and objects are considered mainly because local propagations are quite sensitive to noise and fail to take fully advantage of *a priori* physical constraints like solid shape models. It is clear that evolving interfaces using level set representations is a powerful tool with certain strengths and some limitations. For example, this property (locality) is not helpful when the considered task refers to the extraction of solid objects, while it is a vital element when non-rigid motions and objects are considered.

Our visual space consists of objects from both categories. For example most of the active human organs cannot be considered solid, but at the same time their forms are well constrained within a family of shapes. This family cannot be fully characterized using parametric models. The use of level set-based methods is suitable for this kind of applications due to their ability of dealing with local deformations.

In this paper we consider a challenging application: constrain the level set representations to follow a shape global consistency while preserving the ability to capture local deformations. The most closely related work with our approach can be found in [4,6,10] and more recently in [21]. In [10] a two stage approach was proposed that integrates prior shape knowledge and visual information. During the first step, a segmentation solution is obtained according to a data-driven term, while during the second step a correction of the result is performed using a level set shape prior model that is obtained through a Principal Component Analysis. The same modeling technique is used in [21]. These two steps alternate until convergence is reached. In [4,21] a different technique was considered that refers to an optimization criterion with objective to recover a transformation that better maps the evolving interface to a shape prior term. This criterion is shape driven and in [4] aims at minimizing the Euclidean distance between the model and the prior while in [21] a region-driven criterion is considered that aims at minimizing a metric defined on the level set space. In [4], the shape prior was obtained by averaging the registered training examples and refers to a collection of points. Theoretical comparison of our approach with the ones proposed up to now in can be found in Section 5.

A novel mathematical functional is proposed in this paper that can account for global/local shape properties of the object to be recovered. This functional can be combined with any level set objective function under the assumption that a shape model with a certain degree of variability is available. Our approach consists of two stages. During the first stage a shape model is built directly on the level set space using a collection of samples. This model is constructed using a variational framework that creates a non-stationary pixel-wise model that accounts for shape variabilities. Then, this model is used as basis to introduce the shape prior in an energetic form. This prior aims at minimizing the non-stationary distance between the evolving interface and the shape model in terms of their level set representations. In order to demonstrate the performance of the proposed module, it is integrated with a data-driven variational method to perform image segmentation for physically corrupted and incomplete data.

The remainder of this paper is organized as follows: in section 2, we briefly present the level set representations. Section 3 is dedicated to the construction of the shape prior model using a certain number of examples. In Section 4, we introduce our shape-prior energetic functional that is integrated with a data-driven variational framework. Finally, discussion and conclusions are part of Section 5.

## 2 Level Set Representations

Let us consider a parameterized closed evolving interface in a Euclidean plane  $[C : [0, 1] \rightarrow \mathcal{R}^2, p \rightarrow C(p)]$  and let  $C(p, t)$  the family of interfaces generated by the propagation of the initial one  $C_0(p)$  in the direction of the inward normal  $\mathcal{N}$ . Under the assumption that the propagation is guided by a scalar function  $[F]$  of the geometric properties of the curve (i.e. curvature  $\mathcal{K}$ ), we can have the following motion equation:

$$\begin{cases} C(p, 0) = C_0(p) \\ C_t(p) = F(\mathcal{K}(p)) \mathcal{N}(p) \end{cases} \quad (1)$$

The implementation of this evolution can be done using a Lagrangian approach. In that case we produce the associated equations of motion for the position vector  $(x, y) = C(p)$  and we update them according to a difference approximation scheme. As a consequence, the evolving interface cannot change its topology (with the exception of [12]).

To overcome this limitations, *Osher and Sethian* [14,13,19] have proposed to represent the evolving interface  $C(p)$  with a zero-level set ( $\phi = 0$ ) function of a surface  $z$

$$[z = (x, y, \phi(x, y, t)) \in \mathcal{R}^3] \quad (2)$$

Deriving  $\phi(x, y, t) = 0$  with respect to time and space (given [eq. (1)]) we obtain the following motion for the embedding surface  $\phi()$ :

$$\begin{cases} \phi(C_0(p), 0) = 0 \\ \phi_t(p) = F(\mathcal{K}(p)) |\nabla \phi(p)| \end{cases} \quad (3)$$

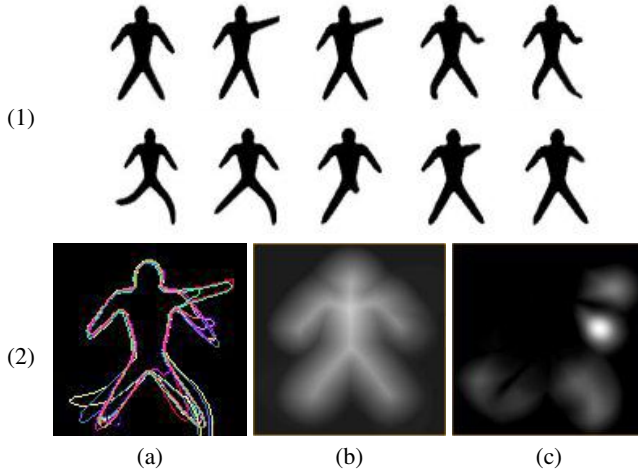
where  $[|\nabla \phi|]$  is the norm of gradient and  $[\mathcal{N} = -\frac{\nabla \phi}{|\nabla \phi|}]$ .

Thus, we have established a connection between the family  $C(p, t)$  and the family of one parameter surfaces  $\phi(x, y, t)$  where the zero iso-surface of the function  $\phi$  yields always to the evolving interface.

The embedding surface  $\phi(p)$  remains a function as long as  $F$  is smooth and the evolving interface  $C(p)$  can change topology. Additionally, numerical simulations on  $\phi(p)$  may be developed trivially and intrinsic geometric properties of the evolving interface can be estimated directly from the level set function. Finally, the method can be easily extended to deal with problems in higher dimensions. A very common selection for the embedding function refers to the use of (Euclidean) distance transforms.

## 3 Shape Prior Model Construction

A vital component for most of the approaches that aimed at creating shape representations is the alignment of the training samples. Matching geometric shapes is an open as well as complex issue in computer vision that has been exhaustively studied. A complete review of the literature in shape matching can be found in [23].



**Fig. 1.** (1) Training Samples, (2) Shape Prior Model, (a) Aligned Shapes, (b) Shape Prior Representation, (c) Model Variability.

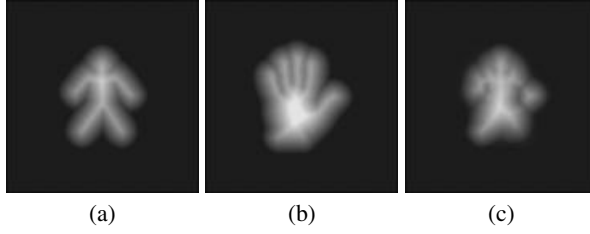
We consider a variational approach that is based on a shape-to-area principle for the alignment [16]. This framework exploits maximally the information of the level set representations. The central idea behind this approach is to perform the registration using the level set representations of the training examples. Therefore, we seek for shape-to-area transformations that best match the level set representations of the shapes of the training set. Thus, given a source shape  $\mathcal{S}$  and a target shape  $\mathcal{D}$  as well as their level set representations using distance transforms, registration is obtained by seeking a global transformation  $[A]$  with a scale component  $[s]$  that minimizes the following dissimilarity measure<sup>1</sup>:

$$E(\Phi_D, \Phi_S, A) = \iint_{\Omega} (s\Phi_S(x, y) - \Phi_D(A(x, y)))^2 dx dy$$

A detailed description and the algorithm and the extension to deal with local deformations can be found in [16]. The output of this procedure is a set of  $N$  level set representations (one for each training sample)  $[\hat{\Phi}_i]$  registered to an arbitrary selected (from the family of samples) reference shape  $[\hat{\Phi}_0]$ . The efficiency of this framework is shown in [fig. (1)].

The next step is the construction of the shape model, using the aligned contours. Point-based snake models [8], deformable models/templates [2], active shapes [5], level set representations [3], etc. are common selections. Although these representations are powerful enough to capture a certain number of local deformations, they require a large number of parameters to deal with important shape deformations. Moreover, they cannot deal with changes of topology (with the exception of level set methods [3] and the approach presented in [12]). Finally, their extension to describe structures of higher dimension than curves and surfaces is not trivial. In the Level Set literature, two models

<sup>1</sup> One can easily prove that level set representation are invariant to translation and rotation but not to scale variations. However, as it will be explained later in the paper, one can predict the effect of scale variations and the appearance of the scale parameter in the energy function.



**Fig. 2.** Construction of a Shape Prior Model using two very different aligned training samples. (a) Training Sample 1, (b) Training Sample 2, (c) Shape Prior Model.

have been presented up to now. In [10,21] the use of a global statistical representation on the level set space was proposed while a different approach was considered in [4] where shapes are represented using a collection of points.

We consider a more challenging approach where the objective is to generate a shape model that accounts for local variations as well. In order to do that, we consider a stochastic framework with two unknown variables:

- The shape image,  $\Phi_M(x, y)$ ,
- The local degrees (variability) of shape deformations  $\sigma_M(x, y)$ .

where each grid location can be described in the shape model using a Gaussian density function

$$p_{x,y}^M(\phi) = \frac{1}{\sqrt{2\pi}\sigma_M(x, y)} e^{-\frac{(\phi - \Phi_M(x, y))^2}{2\sigma_M^2(x, y)}}$$

A similar framework for a different purpose was proposed in [24].

The mean of this probability density function corresponds to the level set function, while the variance refers to the variation of the aligned samples in this location. On top of these assumptions, we impose the constraint that mean values of the shape model refer to a signed distance function (level set representation).

Thus given  $N$  aligned training samples (level set representations) where  $\hat{\Phi}_i$  is the aligned transformation of  $\Phi_i$ , we can construct a variational framework for the estimation of the BEST shape by seeking for the maximum likelihood of the local densities with respect to  $(\Phi_M, \sigma_M)$ :

$$E(\Phi_M, \sigma_M) = - \sum_{i=1}^n \iint_{x,y} \log [p_{x,y}^M(\hat{\Phi}_i(x, y))] dx dy$$

SUBJECT TO THE CONSTRAINT :  $|\nabla \Phi_M(x, y)|^2 = 1, \quad \forall (x, y) \in \Omega$

Additionally, we can enforce spatial coherence on the variability estimates by adding a smoothness

term. Since the constant term ( $\sqrt{2\pi}$ ) does not affect the minimization procedure, the following functional is used:

$$E(\Phi_M, \sigma_M) = (1 - \alpha) \iint_{\Omega} \left( \left( \frac{d}{dx} \sigma_M(x, y) \right)^2 + \left( \frac{d}{dy} \sigma_M(x, y) \right)^2 \right) dx dy \\ + \alpha \iint_{\Omega} \sum_{i=1}^n \left( \log [\sigma_M(x, y)] + \frac{(\hat{\Phi}_i(x, y) - \Phi_M(x, y))^2}{2\sigma_M^2(x, y)} \right) dx dy \\ \text{SUBJECT TO THE CONSTRAINT : } |\nabla \Phi_M(x, y)|^2 = 1, \quad \forall (x, y) \in \Omega$$

where  $[a]$  is a blending parameter between the two energy terms. The interpretation of the objective function is rather simple. The first component is data driven and aims at recovering a level set representation that best accounts for the training samples. The second term is a smoothness constraint on the representation variability. Neighborhood pixels for all registered examples of the training set have to exhibit similar variability properties.

The constrained optimization of this functional can be done using Lagrange multipliers and a gradient descent method. However, given the form of constraints (involvement of first and second order derivatives), we cannot obtain a closed form solution and prove that the conditions which guarantee the validity of Lagrange theorem are satisfied. Moreover, the number of unknown variables of the system is too high  $O(N^2)$  and the system is quite unstable especially when there is large variability among training samples. A possible way to overcome this limitation that is currently investigated refers to the use of an augmented Lagrangian function, but even in that case the proof of validity and the initial conditions are open issues.

An alternative selection refers to a two step optimization process, that separates the two conditions. During the first step, we obtain the "optimal" solution according to the data driven terms, while during the second step we find the "optimal" projection of this solution to the manifold of acceptable solutions (distance functions).

Thus, the unknown variables are obtained by minimizing the previously defined data-driven objective function that preserves some regularity conditions:

$$\begin{cases} \frac{d}{dt} \Phi_M = \alpha \sum_{i=1}^n \frac{(\hat{\Phi}_i - \Phi_M)}{2\sigma_M^2} \\ \frac{d}{dt} \sigma_M = \alpha \sum_{i=1}^n \left[ -\frac{1}{\sigma_M} + \frac{(\Phi - \Phi_M)^2}{\sigma_M^3} \right] + (1 - \alpha) \left[ \frac{\partial^2}{\partial x \partial x} \sigma_M + \frac{\partial^2}{\partial y \partial y} \sigma_M \right] \end{cases}$$

while the projection/correction to the manifold space of accepted solutions (Level Set Representations)<sup>2</sup> is done using a heavily considered Partial Differential Equation [20]:

$$\left\{ \frac{d}{dt} \Phi_M = (1 - \text{sgn}(\Phi_M^0)) (1 - |\nabla \Phi_M|) \right.$$

where  $\Phi_M^0$  is the initial representation (data driven).

<sup>2</sup> The use of the data driven term will modify the evolving representation without respecting the constraint of being a distance function.

These two steps alternate until the system reaches a steady-state solution. Upon convergence of the system, we will obtain a level set representation model, that optimally expresses the properties of the training set using degrees of variability that are constrained to be locally smooth. As far as the initial conditions of the system are concerned, we use the level set representation of the reference sample while the variability estimates are set equal to one for the whole image plane. In order to avoid stability problems, one can replace the variability factors with

$$\sigma_M = 1 + \hat{\sigma}_M$$

and then seek for the estimates of  $\hat{\sigma}_M$  that are constrained to be strictly positive. The performance of our method is demonstrated in [fig. (2)] where two very different training samples are used to generate a model that integrates information from both shapes. An example using training samples from the same family is shown in [fig. (1)].

## 4 Level Set Shape Priors

Let us now consider an image where an object with a shape form similar to the one of the training samples is present. Then, the objective is to recover the image area that corresponds to this object. At the very beginning, we will introduce our approach without using any data-driven term.

### 4.1 Shape-Driven Propagation

Let  $\Phi : \Omega \times \mathcal{R}^+ \rightarrow \mathcal{R}^+$  be a Lipschitz function that refers to level set representation that is evolving over time  $[t]$  given by,

$$\Phi(x, y; t) = \begin{cases} 0 & , (x, y) \in \partial\mathcal{R}(t) \\ + \mathcal{D}((x, y), \partial\mathcal{R}(t)) > 0 & , (x, y) \in \mathcal{R}(t) \\ - \mathcal{D}((x, y), \partial\mathcal{R}(t)) < 0 & , (x, y) \in [\Omega - \mathcal{R}(t)] \end{cases}$$

where  $\partial\mathcal{R}(t)$  refers to the interface (boundaries) of  $\mathcal{R}(t)$ ,  $\mathcal{D}((x, y), \partial\mathcal{R}(t))$  **the minimum Euclidean distance between the pixel  $(x, y)$  and the interface  $\mathcal{R}(t)$**  and  $t$  at time. Let us also define the approximations of Dirac and Heaviside [26] distributions as:

$$\delta_a(\phi) = \begin{cases} 0 & , |\phi| > \alpha \\ \frac{1}{2\alpha} (1 + \cos(\frac{\pi\phi}{\alpha})) & , |\phi| < \alpha \end{cases}$$

$$H_\alpha(\phi) = \begin{cases} 1 & , \phi > \alpha \\ 0 & , \phi < -\alpha \\ \frac{1}{2} (1 + \frac{\phi}{\alpha} + \frac{1}{\pi} \sin(\frac{\pi\phi}{\alpha})) & , |\phi| < \alpha \end{cases}$$

Then it can be shown easily that

$$\begin{aligned} \{(x, y) \in \Omega : \lim_{\alpha \rightarrow 0^+} [H_\alpha(\Phi((x, y); t))] = 1\} &= \mathcal{R} \\ \{(x, y) \in \Omega : \lim_{\alpha \rightarrow 0^+} [\delta_\alpha(\Phi((x, y); t))] = 1\} &= \partial\mathcal{R} \end{aligned}$$

Now, given an interface and (consequently) its level set representation, we would like to evolve it to recover a structure that respects some known shape properties  $\Phi_M(x, y)$ .



We assume that all instances of the evolving representation belong to the family shapes that is generated by applying all possible global transformations (according to a predefined model) to the prior shape model. This assumption is valid mainly for rigid objects.

Thus, given the current level set representation  $\Phi$ , we can assume that there is an ideal transformation  $A = (A_x, A_y)$  between the shape prior and the evolving representation. If we consider that noise does not effect our measures and the absence scale variations, then the optimal transformation will satisfy the following conditions,

$$\begin{cases} (x, y) \rightarrow A(x, y) \\ \Phi(x, y) \approx \Phi_M(A(x, y)), \quad \forall (x, y) : H_\alpha(\Phi(x, y)) \geq 0 \end{cases}$$

In that case, by considering a very simple optimization criterion like the sum of squared differences, the optimal transformation  $A$  should minimize the following functional:

$$E(\Phi, A) = \iint_{\Omega} H_\alpha(\Phi(x, y)) (\Phi(x, y) - \Phi_M(A(x, y)))^2 dx dy$$

In order to account for scale variations, we can assume a scale component  $[s]$  for the transformation  $A$ . It is straightforward to show that the level set representations are invariant to translation and rotation but not to scale variations. Given the properties of distance transforms from an interface, one can predict how scale changes will affect the information space: the level set representation values will be also scaled up/down according to the scale variable, resulting in the following scale/rotation/translation invariant criterion:

$$E(\Phi, A) = \iint_{\Omega} H_\alpha(\Phi(x, y)) (s\Phi(x, y) - \Phi_M(A(x, y)))^2 dx dy$$

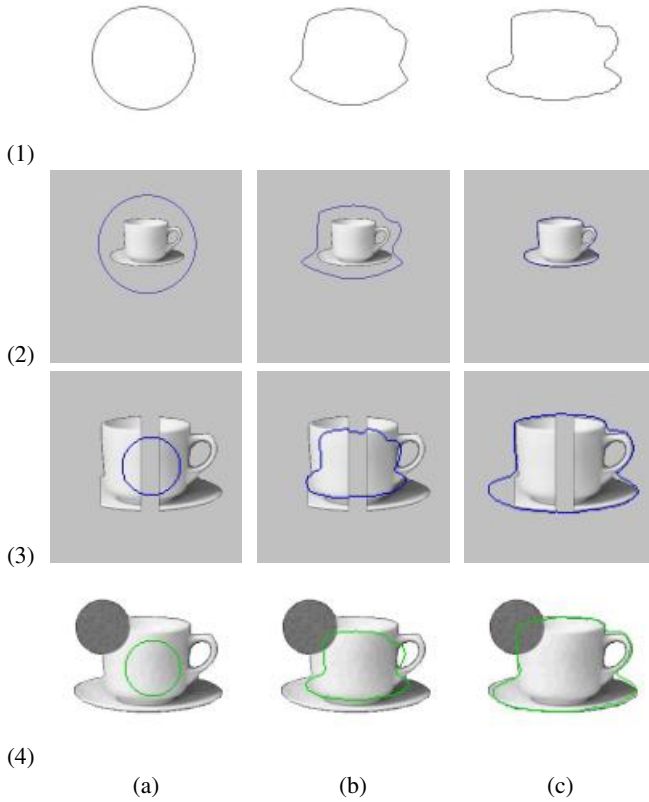
Thus, we are seeking for a transformation that provides pixel-wise level set values correspondences between the evolving interface and the shape prior level set representation.

In order to minimize the above functional with respect to the evolving level set representation and the global linear transformation, we will assume (without loss of generality) that it is composed of  $N + 1$  motion parameters  $A = [s, a_1, a_2, \dots, a_N]$ . Then, using the calculus of variations we obtain the following system of coupled equations:

$$\begin{cases} \frac{d}{dt} \Phi = -2s H_\alpha(\Phi) (s\Phi - \Phi_M(A)) + \delta_\alpha(\Phi) (s\Phi - \Phi_M(A))^2 \\ \frac{d}{dt} s = -2 \iint_{\Omega} \left[ H_\alpha(\Phi) (s\Phi - \Phi_M(A)) \left( \Phi - \nabla \Phi_M(A) \cdot \frac{\partial}{\partial s} (A_x, A_y) \right) \right] \\ \forall j \in [1, N], \\ \frac{d}{dt} a_j = 2 \iint_{\Omega} \left[ H_\alpha(\Phi) (s\Phi - \Phi_M(A)) \left( \nabla \Phi_M(A) \cdot \frac{\partial}{\partial a_j} (A_x, A_y) \right) \right] \end{cases}$$

Let us now try to interpret the obtained motion equation for the time evolving level set representation term by term. We will consider  $\Phi \rightarrow 0$  to facilitate the interpretation:

- The first term  $[-2sH_\alpha(\Phi)(s\Phi - \Phi_M(A)) = \Phi_M(A)]$  is positive when  $\Phi_M(A)$  is positive. The physical meaning of this condition is that the projection of the considered pixel is interior to the shape prior interface. Therefore, the evolving interface has to expand locally resulting on a better fit with the model.



**Fig. 3.** (1) *Shape Prior Term*, (2) *Shape-Constrained Geodesic Active Contours*, (3-4) *Shape-Constrained Geodesic Active Regions*, (a) *Initial Contour*, (b) *Mid-Contour*, (c) *Final Contour*. Different scales are used.

- The second force aims at decreasing the length and the area defined by the evolving interface and consequently the value of the cost function. Therefore, one can ignore this component.

This simple/static model can have encouraging performance [fig. (3)]. However, it does not take into account the local shape variations and is constrained by a rigid transformation between the evolving representation and the shape prior model.

During the model construction, we have considered that the shape model can have some local degrees of variability. In that case the ideal transformation will map each value of current representation at the most probable value on the model:

$$\left\{ \begin{array}{l} (x, y) \rightarrow A(x, y) \\ \max_{x, y} \left\{ p_{A(x, y)}^M (s\Phi(x, y)) \right\} \forall (x, y) : H_a(\Phi(x, y)) \geq 0 \end{array} \right.$$

The most probable transformation is the one obtained through the maximum likelihood for all pixels. Under the assumption that densities are independent across pixels, the

minimization of the  $-\log$  function of the maximum likelihood can be considered as global optimization criterion. This criterion refers to two set of unknown variables. The linear transformation  $A$ , and the level set function  $\Phi$ :

$$\begin{aligned} E(\Phi, A) &= - \iint_{\Omega} H_{\alpha}(\Phi(x, y)) \log \left[ p_{A(x, y)}^M(s\Phi(x, y)) \right] dx dy \\ &= \iint_{\Omega} H_{\alpha}(\Phi(x, y)) \left[ \log(\sigma_M(A(x, y))) + \frac{(s\Phi(x, y) - \Phi_M(A(x, y)))^2}{2\sigma_M^2(A(x, y))} \right] dx dy \end{aligned}$$

The interpretation of this functional is rather simple; we seek a transformation and a level set representation that maximizes the posterior probability pixel-wise given the shape prior model. This model refers to a non-stationary measurement where pixels are considered according to the confidence of their projections in the shape prior model (variance term).

The minimization of this functional can be done using a gradient descent method:

$$\begin{cases} \frac{d}{dt}\Phi = -s H_{\alpha}(\Phi) \left[ \frac{(s\Phi - \Phi_M(A))}{\sigma_M^2(A)} \right] - \delta_{\alpha}(\Phi) \left[ \log(\sigma_M(A)) + \frac{(s\Phi - \Phi_M(A))^2}{2\sigma_M^2(A)} \right] \\ \frac{d}{dt}s = -2 \iint_{\Omega} H_{\alpha}(\Phi) \left[ \frac{1}{2\sigma_M(A)} \nabla \sigma_M(A) \cdot \frac{\partial}{\partial s}(A_x, A_y) \right. \\ \quad \left. + \frac{(s\Phi - \Phi_M(A))^2 [\nabla \sigma_M(A) \cdot \frac{\partial}{\partial s}(A_x, A_y)]}{\sigma_M^3(A)} \right. \\ \quad \left. - \frac{(s\Phi - \Phi_M(A)) (\Phi - \nabla \Phi_M(A) \cdot \frac{\partial}{\partial s}(A_x, A_y))}{\sigma_M^2(A)} \right] \end{cases}$$

$$\begin{cases} \forall j \in [1, N], \\ \frac{d}{dt}a_j = -2 \iint_{\Omega} H_{\alpha}(\Phi) \left[ \frac{1}{2\sigma_M(A)} \nabla \sigma_M(A) \cdot \frac{\partial}{\partial a_j}(A_x, A_y) - \right. \\ \quad \left. \frac{(s\Phi - \Phi_M(A)) [\nabla \Phi_M(A) \cdot \frac{\partial}{\partial s}(A_x, A_y)]}{\sigma_M^2(A)} - \frac{(s\Phi - \Phi_M(A))^2 [\nabla \sigma_M(A) \cdot \frac{\partial}{\partial a_j}(A_x, A_y)]}{\sigma_M^3(A)} \right] \end{cases}$$

We recall that the second term on the evolution of the level set representation is ignored. Also due to the fact that the scale  $[s]$  parameter appears in the cost function, a different motion equation is obtained in comparison with the other parameters of the registration model.

The obtained motion equations have the same interpretation with the ones presented earlier without the local variability factor. In the absence of data driven term, they will have the same behavior with the ones that do not account for local variability. On the other hand, the integration of the shape prior with data-driven terms can provide a soft-to-hard constraint. In order to demonstrate the efficiency of the proposed functional, we will integrate it with an existing well known data-driven variational framework for image segmentation.

## 4.2 Self-Constrained Geodesic Active Region

The geodesic active region model was originally proposed in [15], and aimed at combining boundary (in the form of Geodesic Active Contours) with some regional/global properties of the object to be recovered. This model makes the assumption that some *a priori* knowledge regarding the global region/statistical properties are available (intensities, optical flow, texture information, etc.)

The original model was defined on the image plane, and the obtained motion equation were implemented using level set methods. Here, we will introduce a self-constrained version of this model, directly on the level set representation space. Thus, if we consider that some region-based image-based descriptor functions  $p_i$  are available that capture the intensity properties of each region, then the following objective functional

$$E(\Phi_i, A_i) = a \sum_{i=1}^N \iint_{\Omega} \delta_{\alpha}(\Phi_i(x, y)) g(|\nabla I(x, y)|) |\nabla \Phi_i(x, y)| + \\ b \sum_{i=1}^N \iint_{\Omega} [H_{\alpha}(\Phi_i(x, y)) g(p_i(I(x, y))) + (1 - H_{\alpha}(\Phi_i(x, y))) g(p_0(I(x, y)))] + \\ c \sum_{i=1}^N \iint_{\Omega} H_{\alpha}(\Phi_i(x, y)) \left[ \log(\sigma_{M_i}(A_i(x, y))) + \frac{(s \Phi_i(x, y) - \Phi_{M,i}(A_i(x, y)))^2}{2\sigma_{M,i}^2(A_i)}(x, y) \right]$$

where  $p_0$  is the descriptor function that captures the background properties.

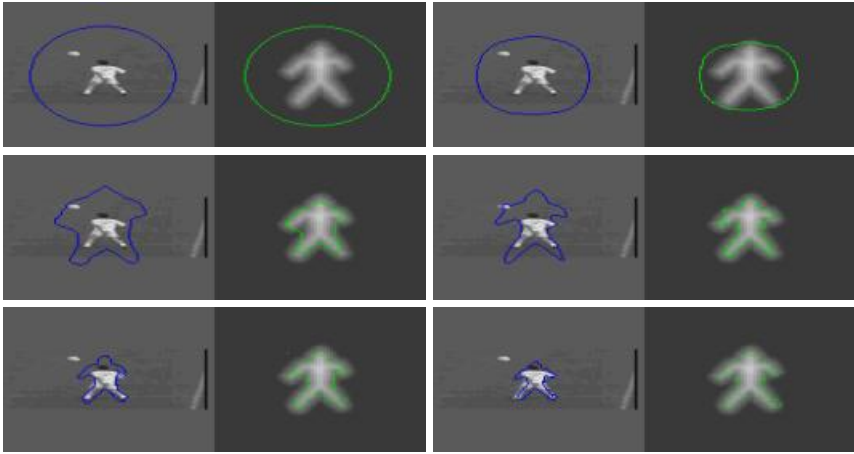
The minimization of this function with respect to the time evolving level set representations  $\Phi_i$  can be done using the calculus of variations and the following equations are obtained:

$$\begin{cases} \forall i \in [1, N], \\ \frac{d}{dt}(\Phi_i) = a \delta_{\alpha}(\Phi_i) [g(I)\mathcal{K} - \nabla g(I)\nabla \Phi_i] + b \delta_{\alpha}(\Phi_i) [g(p_i(I)) - g(p_0(I))] - \\ c H_{\alpha}(\Phi_i) s \frac{(s \Phi_i - \Phi_{M,i}(A_i))}{\sigma_{M,i}^2(A)} - c \delta_{\alpha}(\Phi_i) \left[ \log(\sigma_{M,i}(A_i)) + \frac{(s \Phi_i - \Phi_{M,i}(A_i))^2}{2\sigma_{M,i}^2(A_i)} \right] \end{cases}$$

that consist of three forces acting locally on the evolving interface all in the direction of the normal:

- An image-driven boundary force that shrinks the evolving interface (constrained by the curvature effect) towards the object boundaries,
- An image-driven region/statistical force that shrinks or expands the evolving interface towards the direction that optimizes the separation between the background pixels and the object pixels according to some predefined global statistical properties,
- A shape-driven force that shrinks or expands the evolving interface towards the direction that produces a segmentation result which satisfies some predefined shape constraints.

In the absence of regional information, we can consider the geodesic active contour model [3,9] that only makes use of boundary information.



**Fig. 4.** *Self-Constrained Geodesic Active Regions (raster-scan format). The shape prior model was created using synthetic samples [fig. 3]). The projection of the evolving interface to the shape prior model is also shown.*

### 4.3 Implementation Issues

The last issue to be dealt with, is the numerical implementation of the proposed framework. The level set implementation is performed using the Narrow Band Method [1]. The essence of this method is to perform the level set propagation only within a limited zone ( $\alpha$  parameter of the DIRAC and HEAVISIDE distributions) that is located around the latest position of the propagating contours (in the inward and outward direction). Thus, the working area is reduced significantly resulting on a significant decrease of the computational complexity per iteration. However, this method requires a frequent re-initialization of the level set functions that is performed using the Fast Marching algorithm [19]. A similar algorithm within the area of automatic control was proposed in [22].

## 5 Discussion and Summary

In this paper, we have proposed a novel approach for introducing shape priors into level set representations targeting 2D closed structures. Encouraging experimental results were obtained for real and synthetic images. Two key contributions are presented in this paper.

- The first, refers to new way of defining global-to-local shape prior models in the level set representations space according to probabilistic principles. This is obtained through a constrained variational framework that exploits maximally the information of the level set representations and can account for local degrees of variability.
- The second, refers to a novel energetic term that can account for shape priors in level set representations. This term is defined directly on the level set space and can deal with global transformations. Moreover, it can account for local variations due to the shape prior model.

Furthermore, this paper deals with registration and segmentation simultaneously. The objective is to recover a segmentation map that is in accordance with the shape prior model as well as a rigid registration between this map and the model. Due to the use of distance transforms [16] in the registration process, the method is robust to local deformations. This is due to the nature of this transformation that scales down local deformations when considered in a certain distance from the contour.

These two components have been integrated to an existing well known level set segmentation framework, the Geodesic Active Region model. The resulting functional refers to a joint optimization approach that can deal with important shape deformations, as well as with noisy physically corrupted and occluded data.

The proposed framework, to our understanding is favorably compared with the existing level set shape prior methods [4,10,21]. One can claim that our alignment method compared with the ones proposed up to now within this problem can have a better performance due to the information space that is used. Moreover, the proposed shape prior model can naturally account for local degrees of variability which is not the case for [4] and performs better than the model employed in [21]. Also, we claim that the construction of this model does not require a significant number of samples as the one proposed in [10,21]. Then, the shape prior term can account for a large variety of global transformations (opposite to [10]) as well as scale variations (opposite to [10,21]) and can deal with important local shape variations (opposite to [4]). Finally, we have proposed a robust method to estimate this transformation where a shape-to-area approach is considered that maximally exploits the information of the level set representations (opposite to [4] where pixel-wise shape correspondences are considered). Moreover, the extension of the proposed framework to deal with objects of any arbitrary dimension is trivial (opposite to [4]).

Regarding the computational cost of our approach, it is comparable (lower bound) with the one of the geodesic active region model. The most expensive part of the algorithm is the implementation of the level set propagation, and this part is common in both methods.

As far the future directions of this approach are considered, several issues remain open. A key characteristic of the Level Set Representations is the ability of changing the topology of the evolving interface. Although modeling and extracting separately multi-seeds objects can be naturally handled, we cannot detect multiple objects with different shape prior models by considering a single level set representation. At the same time, the behavior of the method is questionable when multiple initial seeds are used to recover a single object due to the assumption that there is a global transformation between the evolving representation and the shape prior model. This is a challenging perspective that will be explored in the near future. Also, the mathematical justification of the model is a step forward to be done. The use of this framework to deal simultaneously with segmentation and registration of medical volumes is a challenging application. In order to do that, investigating the use of faster numerical approximation techniques [25] to implement the proposed framework is a step to be done. Finally, the validation of the method is an open issue. Towards, this end the task of the segmentation of the left ventricle is considered where the shape prior holds a principle role during the segmentation process.

**Acknowledgments.** The authors would like to thank M.-P. Jolly, B. Bascle, C. Xu and V. Ramesh for fruitful discussions, the anonymous reviewers for their suggestions and Alok Gupta, the director of the IM Department at Siemens Corporate Research for encouraging the exploration of the topic and providing financial support to Mikael Rousson. We are also thankful to T. Sebastian and B. Kimia from Brown University for providing their shape database [18].

## References

1. D. Adalsteinsson and J. Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118:269–277, 1995.
2. A. Blake and M. Isard. *Active Contours*. Springer-Verlag Press, 1997.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IEEE ICCV*, pages 694–699, Boston, USA, 1995.
4. Y. Chen, H. Thiruvankadam, H. Tagare, F. Huang, and D. Wilson. On the Incorporation of Shape Priors into Geometric Active Contours. In *IEEE VLSP*, pages 145–152, 2001.
5. T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active Shape Models - their training and applications. *CVGIP: Image Understanding*, 61, 1995.
6. D. Cremers, C. Schnorr, and J. Weickert. Diffusion-Snakes: Combining Statistical Shape Knowledge and Image Information in a Variational Framework. In *IEEE VLSP*, pages 137–144, 2001.
7. O. Faugeras and R. Keriven. Variational principles, Surface Evolution, PDE's, level set methods and the Stereo Problem. *IEEE TIP*, 7:336–344, 1998.
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *IEEE ICCV*, pages 261–268, 1987.
9. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *IEEE ICCV*, pages 810–815, Boston, USA, 1995.
10. M. Leventon, E. Grimson, and O. Faugeras. Statistical Shape Influence in Geodesic Active Contours. In *IEEE CVPR*, pages 1:316–322, 2000.
11. R. Malladi and J. Sethian. A Real-Time Algorithm for Medical Shape Recovery. In *IEEE ICCV*, pages 304–310, Bombay, India, 1998.
12. T. McIrerney and D. Terzopoulos. Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation. *IEEE TMI*, 18:840–850, 1999.
13. S. Osher and R. Fedkiw. Level Set Methods. Technical report, Mathematics Department, UCLA, 2000.
14. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : algorithms based on the hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
15. N. Paragios and R. Deriche. Geodesic Active regions for Supervised Texture Segmentation. In *IEEE ICCV*, pages 926–932, Corfu, Greece, 1999. Previous: INRIA Research Report, RR 3440, June 1998, <http://www.inria.fr/RRRT/RR-3440.html>.
16. N. Paragios, M. Rousson, and V. Ramesh. Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration. Copenhagen, Denmark, 2002.
17. G. Sapiro. *Geometric Partial Differential Equations in Image Processing*. Cambridge University Press, Jan. 2001.
18. T. Sebastian, P. Klein, and B. Kimia. Recognition of Shapes by Editing Shock Graphs. In *IEEE ICCV*, pages 755–762, Vancouver, Canada, 2001.
19. J. Sethian. *Level Set Methods*. Cambridge University Press, 1996.

20. M. Sussman, P. Smereka, and S. Osher. A Level Set Method for Computing Solutions to incompressible Two-Phase Flow. *Journal of Computational Physics*, 114:146–159, 1994.
21. A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, A. Grimson, and A. Willsky. Model-based Curve Evolution Technique for Image Segmentation. In *IEEE CVPR*, volume I, pages 463–468, 2001.
22. J. Tsitsiklis. Efficient algorithms for Globally Optimal Trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.
23. R. Velkamp and M. Hagedoorn. State-of-the-art in Shape Matching. Technical Report UU-CS-1999-27, Utrecht University, Sept. 1999.
24. Y. Wang and L. Staib. Elastic Model Based Non-rigid Registration Incorporating Statistical Shape Information. In *MICCAI*, pages 1162–1173, 1998.
25. J. Weickert, B. M. t. Haar Romeny, and M. Viergener. Efficient and Reliable Scheme for Non-Linear Diffusion and Filtering. *IEEE TIP*, 7:398–410, 1998.
26. H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127:179–195, 1996.



# Nonlinear Shape Statistics in Mumford–Shah Based Segmentation

Daniel Cremers, Timo Kohlberger, and Christoph Schnörr

Computer Vision, Graphics and Pattern Recognition Group  
Department of Mathematics and Computer Science  
University of Mannheim, D-68131 Mannheim, Germany  
{cremers,tiko,schnoerr}@uni-mannheim.de  
<http://www.cvgpr.uni-mannheim.de>

**Abstract.** We present a variational integration of nonlinear shape statistics into a Mumford–Shah based segmentation process. The nonlinear statistics are derived from a set of training silhouettes by a novel method of density estimation which can be considered as an extension of kernel PCA to a stochastic framework.

The idea is to assume that the training data forms a Gaussian distribution after a nonlinear mapping to a potentially higher-dimensional feature space. Due to the strong nonlinearity, the corresponding density estimate in the original space is highly non-Gaussian. It can capture essentially arbitrary data distributions (e.g. multiple clusters, ring- or banana-shaped manifolds).

Applications of the nonlinear shape statistics in segmentation and tracking of 2D and 3D objects demonstrate that the segmentation process can incorporate knowledge on a large variety of complex real-world shapes. It makes the segmentation process robust against misleading information due to noise, clutter and occlusion.

**Keywords:** Segmentation, shape learning, nonlinear statistics, density estimation, Mercer kernels, variational methods, probabilistic kernel PCA

## 1 Introduction

One of the challenges in the field of image segmentation is the incorporation of prior knowledge on the shape of the segmenting contour. The general idea is to learn the possible shape deformations of an object statistically from a set of training shapes, and to then restrict the contour deformation to the subspace of familiar shapes during the segmentation process. For the problem of segmenting a known object — such as an anatomical structure in a medical image — this approach has been shown to drastically improve segmentation results [15,8].

Although the shape prior can be quite powerful in compensating for misleading information due to noise, clutter and occlusion in the input image, most approaches are limited in their applicability to more complicated shape variations of real-world objects. The permissible shapes are assumed to form a multivariate

Gaussian distribution, which essentially means that all possible shape deformations correspond to linear combinations of a set of eigenmodes, such as those given by principal component analysis (cf. [14,4,15]). In particular, this means that for any two permissible shapes, the entire sequence of shapes obtained by a linear morphing of the two shapes is permissible as well.

Once the set of training shapes exhibits highly nonlinear shape deformations — such as different 2D views of a 3D object — one finds distinct clusters in shape space corresponding to the stable views of an object. Moreover, each of the clusters may by itself be quite non-Gaussian. The Gaussian hypothesis will then result in a mixing of the different views, and the space of accepted shapes will be far too large for the prior to sensibly restrict the contour deformation.

A number of models have been proposed to deal with nonlinear shape variation. However, they often suffer from certain drawbacks. Some involve a complicated model construction procedure [3]. Some are supervised in the sense that they assume prior knowledge on the structure of the nonlinearity [12]. Others require prior classification with the number of classes to be estimated or specified beforehand and each class being assumed Gaussian [13,5]. And some cannot be easily extended to shape spaces of higher dimension [11].

In the present paper we present a density estimation approach which is based on Mercer kernels [6] and which does not suffer from any of the mentioned drawbacks. In Section 2 we review the variational integration of a linear shape prior into Mumford–Shah based segmentation. In Section 3 we present the nonlinear density estimate which was first introduced in [7]. We discuss its relation to kernel PCA and to the classical Parzen estimator, give estimates of the involved parameters and illustrate its application to artificial 2D data and to silhouettes of real objects. In Section 4 this nonlinear shape prior is integrated into segmentation. We propose a variational integration of similarity invariance. Numerous examples of segmentation with and without shape prior on static images and tracking sequences finally confirm the properties of the nonlinear shape prior: it can encode very different shapes and generalizes to novel views without blurring or mixing different views. Furthermore, it improves segmentation by reducing the dimension of the search space, by stabilizing with respect to clutter and noise and by reconstructing the contour in areas of occlusion.

## 2 Statistical Shape Prior in Mumford–Shah Segmentation

In [8] we presented a variational integration of statistical shape knowledge in a Mumford–Shah based segmentation. We suggested modifications of the Mumford–Shah functional and its cartoon limit [17] which facilitate the implementation of the segmenting contour as a parameterized spline curve:

$$C_{\mathbf{z}} : [0, 1] \rightarrow \Omega \subset \mathbb{R}^2, \quad C_{\mathbf{z}}(s) = \sum_{n=1}^N \begin{pmatrix} x_n \\ y_n \end{pmatrix} B_n(s), \quad (1)$$

where  $B_n$  are quadratic B-spline basis functions [10], and  $\mathbf{z} = (x_1, y_1, \dots, x_N, y_N)^t$  denotes the control points. Shape statistics can then be ob-

tained by estimating the distribution of the control point vectors corresponding to a set of contours which were extracted from binary training images.

In the present paper we focus on significantly improving the shape statistics. We will therefore restrict ourselves to the somewhat simpler cartoon limit of the modified Mumford–Shah functional. The segmentation of a given grey value input image  $f : \Omega \rightarrow [0, 255]$  is obtained by minimizing the energy functional

$$E_{MS}(C, u_o, u_i) = \frac{1}{2} \int_{\Omega_i} (f - u_i)^2 dx + \frac{1}{2} \int_{\Omega_o} (f - u_o)^2 dx + \nu \mathcal{L}(C) \quad (2)$$

with respect to  $u_o$ ,  $u_i$  and the segmenting contour  $C$ . This enforces a segmentation into an inside region  $\Omega_i$  and an outside region  $\Omega_o$  with piecewise constant grey values  $u_i$  and  $u_o$ , such that the variation of the grey value is minimal within each region.<sup>1</sup>

In [8] we proposed to measure the length of the contour by the squared  $\mathcal{L}_2$ -norm  $\mathcal{L}(C) = \int_0^1 \left(\frac{dC}{ds}\right)^2 ds$ , which is more adapted to the implementation of the contour as a closed spline curve than the usual  $\mathcal{L}_1$ -norm, because it enforces an equidistant spacing of control points. Beyond just minimizing the length of the contour, one can minimize a shape energy  $E_{shape}(C)$ , which measures the dissimilarity of the given contour with respect to a set of training contours. Minimizing the total energy

$$E(C, u_o, u_i) = E_{MS}(C, u_o, u_i) + \alpha E_{shape}(C) \quad (3)$$

will enforce a segmentation which is based on both the input image and the similarity to a set of training shapes.

In order to study the interaction between statistical shape knowledge and image grey value information we restricted the shape statistics in [8] to a common model by assuming the training shapes to form a multivariate Gaussian distribution in shape space. This corresponds to a quadratic shape energy on the spline control point vector  $\mathbf{z}$ :

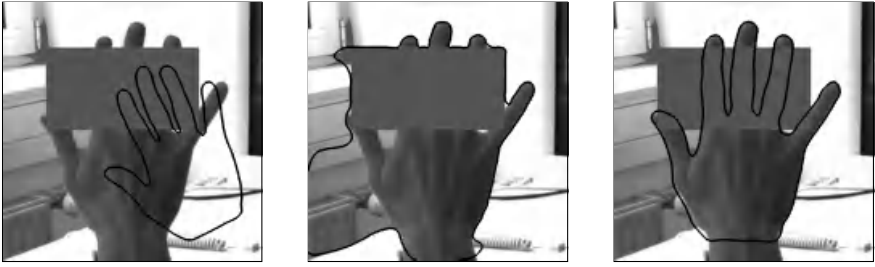
$$E_{shape}(C_{\mathbf{z}}) = (\mathbf{z} - \mathbf{z}_0)^t \Sigma^{-1} (\mathbf{z} - \mathbf{z}_0), \quad (4)$$

where  $\mathbf{z}_0$  denotes the mean control point vector and  $\Sigma$  the covariance matrix after appropriate regularization [8]. The effect of this shape energy in dealing with clutter and occlusion is exemplified in Figure 1. For the input image  $f$  of a partially occluded hand, we performed a gradient descent to minimize the total energy (3) without ( $\alpha = 0$ ) and with ( $\alpha > 0$ ) shape prior.

### 3 Density Estimation in Feature Space

Unfortunately, the linear shape statistics (4) are limited in their applicability to more complicated shape deformations. As soon as the training shapes form

<sup>1</sup> The underlying piecewise-constant image model can easily be generalized to incorporate higher-order grey value statistics [27] or edge information [18]. In this paper, however, we focus on modeling shape statistics and therefore do not consider these possibilities.



**Fig. 1.** Segmentation with **linear** shape prior on an image of a partially occluded hand: initial contour (left), segmentation without shape prior (center), and segmentation with shape prior (right). The statistical shape prior compensates for misleading information due to noise, clutter and occlusion. Integration into the variational framework effectively reduces the dimension of the search space and enlarges the region of convergence.

distinct clusters in shape space — such as those corresponding to the stable views of a 3D object — or the shapes of a given cluster are no longer distributed according to a hyperellipsoid, the Gaussian shape prior tends to mix classes and blur details of the shape information in such a way that the resulting shape prior is no longer able to effectively restrict the contour evolution to the space of familiar shapes.

In the following we present an extension of the above method which incorporates a strong nonlinearity at almost no additional effort. Essentially we propose to perform a density estimation not in the original space but in the feature space of nonlinearly transformed data. The nonlinearity enters in terms of Mercer kernels [6], which have been extensively used in the classification and support vector community [1,2], but which have apparently been studied far less in the field of density estimation. In the present section we present the method of density estimation, discuss its relation to kernel principal component analysis (kernel PCA) [23] and to the Parzen estimator [20,19], and propose estimates of the involved parameters. Finally we illustrate the density estimate in applications to artificial 2D data and to 200-dimensional data corresponding to silhouettes of real-world training shapes.

### 3.1 Gaussian Density in Kernel Space

Let  $\mathbf{z}_1, \dots, \mathbf{z}_m \in \mathbb{R}^n$  be a given set of training data. We propose to map the data by a nonlinear function  $\phi$  to a potentially higher-dimensional space  $Y$ . Denote a mapped point after centering with respect to the training points by

$$\tilde{\phi}(\mathbf{z}) := \phi(\mathbf{z}) - \phi_0 = \phi(\mathbf{z}) - \frac{1}{m} \sum_{i=1}^m \phi(\mathbf{z}_i), \quad (5)$$

and let the Mercer kernel [6]  $k(\mathbf{x}, \mathbf{y}) := (\phi(\mathbf{x}), \phi(\mathbf{y}))$  denote the corresponding scalar product for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . Denote the centered kernels by

$$\tilde{k}(\mathbf{x}, \mathbf{y}) := (\tilde{\phi}(\mathbf{x}), \tilde{\phi}(\mathbf{y})) = k(\mathbf{x}, \mathbf{y}) - \frac{1}{m} \sum_{k=1}^m (k(\mathbf{x}, \mathbf{z}_k) + k(\mathbf{y}, \mathbf{z}_k)) + \frac{1}{m^2} \sum_{k,l=1}^m k(\mathbf{z}_k, \mathbf{z}_l). \quad (6)$$

We estimate the distribution of the *mapped* training data by a Gaussian probability density in the space  $Y$  — see Figure 2. The corresponding energy is given by the negative logarithm of the probability, and can be considered as a measure of the dissimilarity between a point  $\mathbf{z}$  and the training data:

$$E_\phi(\mathbf{z}) = \tilde{\phi}(\mathbf{z})^t \Sigma_\phi^{-1} \tilde{\phi}(\mathbf{z}). \quad (7)$$

In general the covariance matrix  $\Sigma_\phi$  is not invertible. We therefore regularize it by replacing the zero eigenvalues by a constant  $\lambda_\perp$ :

$$\Sigma_\phi = V \Lambda V^t + \lambda_\perp (I - V V^t), \quad (8)$$

where  $\Lambda$  denotes the diagonal matrix of nonzero eigenvalues  $\lambda_1 \leq \dots \leq \lambda_r$  and  $V$  is the matrix of the corresponding eigenvectors  $V_1, \dots, V_r$ . By definition of  $\Sigma_\phi$ , these eigenvectors lie in the span of the mapped training data:

$$V_k = \sum_{i=1}^m \alpha_i^k \tilde{\phi}(\mathbf{z}_i), \quad 1 \leq k \leq r. \quad (9)$$

In [23] it is shown that the eigenvalues  $\lambda_k$  of the covariance matrix correspond (up to the factor  $m$ ) to the nonzero eigenvalues of the  $m \times m$ -matrix  $K$  with entries  $K_{ij} = \tilde{k}(\mathbf{z}_i, \mathbf{z}_j)$ , and that the expansion coefficients  $\{\alpha_i^k\}_{i=1, \dots, m}$  in (9) form the components of the  $k$ -th eigenvector of  $K$ .

Inserting (8) splits energy (7) into two terms:

$$E_\phi(\mathbf{z}) = \sum_{k=1}^r \lambda_k^{-1} \left( V_k, \tilde{\phi}(\mathbf{z}) \right)^2 + \lambda_\perp^{-1} \left( |\tilde{\phi}(\mathbf{z})|^2 - \sum_{k=1}^r \left( V_k, \tilde{\phi}(\mathbf{z}) \right)^2 \right). \quad (10)$$

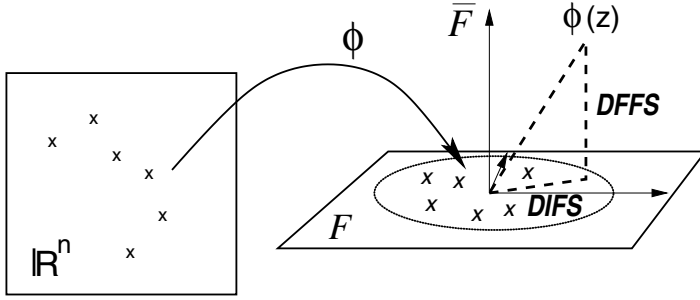
With expansion (9), we obtain the final expression for our energy:

$$E_\phi(\mathbf{z}) = \sum_{k=1}^r \left( \sum_{i=1}^m \alpha_i^k \tilde{k}(\mathbf{z}_i, \mathbf{z}) \right)^2 \cdot (\lambda_k^{-1} - \lambda_\perp^{-1}) + \lambda_\perp^{-1} \cdot \tilde{k}(\mathbf{z}, \mathbf{z}). \quad (11)$$

As in the case of kernel PCA, the nonlinearity  $\phi$  only appears in terms of the kernel function. This allows to specify an entire family of possible nonlinearities by the choice of the associated kernel. For all our experiments we used the Gaussian kernel:

$$k(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp \left( -\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2} \right), \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \quad (12)$$

We refer to Section 3.4 for a justification of this choice.



**Fig. 2.** Nonlinear mapping into  $Y = F \oplus \bar{F}$  and the distances DIFS and DFFS.

### 3.2 Relation to Kernel PCA

Just as in the linear case (cf. [16]), the regularization (8) of the covariance matrix causes a splitting of the energy into two terms (10), which can be considered as a *distance in feature space* (DIFS) and a *distance from feature space* (DFFS) — see Figure 2. For the purpose of pattern reconstruction in the framework of kernel PCA, it was suggested to minimize a reconstruction error [22], which is identical with the DFFS. This procedure is based on the assumption that the entire plane spanned by the mapped training data corresponds to acceptable patterns. However, this is not a valid assumption: already in the linear case, moving too far along an eigenmode will produce patterns which have almost no similarity to the training data, although they are still accepted by the hypothesis. Moreover, the distance DFFS is not based on a probabilistic model. In contrast, energy (11) is derived from a Gaussian probability distribution. It minimizes both the DFFS and the DIFS; the latter can be considered a Mahalanobis distance in feature space.

### 3.3 On the Regularization of the Covariance Matrix

A regularization of the covariance matrix in the case of kernel PCA — as done in (8) — was first proposed in [7] and has also been suggested more recently in [24]. The choice of the parameter  $\lambda_{\perp}$  is not a trivial issue. For the linear case, such regularizations of the covariance matrix have also been proposed [4,16,21,25,9]. There [16,25], the constant  $\lambda_{\perp}$  is estimated as the mean of the replaced eigenvalues by minimizing the Kullback–Leibler distance of the corresponding densities. However, we believe that this is not the appropriate regularization of the covariance matrix. The Kullback–Leibler distance is supposed to measure the error with respect to the correct density, which means that the covariance matrix calculated from the training data is assumed to be the correct one. But this is not the case because the number of training points is limited. For essentially the same reason this approach does not extend to the nonlinear case considered here: depending on the type of nonlinearity  $\phi$ , the covariance matrix is potentially infinite-dimensional such that the mean over all replaced eigenvalues will be

zero. As in the linear case [9], we therefore propose to choose  $0 < \lambda_{\perp} < \lambda_r$ , which means that unfamiliar variations from the mean are less probable than the smallest variation observed on the training set. In practice we fix  $\lambda_{\perp} = \lambda_r/2$ .

### 3.4 Relation to Classical Density Estimation

Why should the training data after a nonlinear mapping corresponding to the kernel (12) be distributed according to a Gaussian density? The final expression of the density estimate (11) resembles the well-known Parzen estimator [20,19], which estimates the density of a distribution of training data by summing up the data points after convolution with a Gaussian (or some other kernel function).

In fact, the energy associated with an *isotropic* (spherical) Gaussian distribution in feature space is (up to normalization) equivalent to a Parzen estimator in the original space. In the notations of (5) and (6), this energy is given by the Euclidean feature space distance

$$E_{sphere}(z) = |\tilde{\phi}(z)|^2 = \tilde{k}(z, z) = -\frac{2}{m} \sum_{i=1}^m k(\mathbf{z}, \mathbf{z}_i) + \text{const.}$$

Up to scaling and a constant, this is the Parzen estimator.

Due to the regularization of the covariance matrix in (8), the energy associated with the more general anisotropic feature space Gaussian (7) contains a (dominant) isotropic component given by the last term in (11). We believe that this connection to the Parzen estimator justifies the assumption of a Gaussian in feature space and the choice of localized kernels such as (12).

Numerical simulations show that the remaining anisotropic component in (11) has an important influence. However, a further investigation of this influence is beyond the scope of this paper.

### 3.5 On the Choice of the Hyperparameter $\sigma$

The last parameter to be fixed in the proposed density estimate is the hyperparameter  $\sigma$  in (12). Let  $\mu$  be the average distance between two neighboring data points:

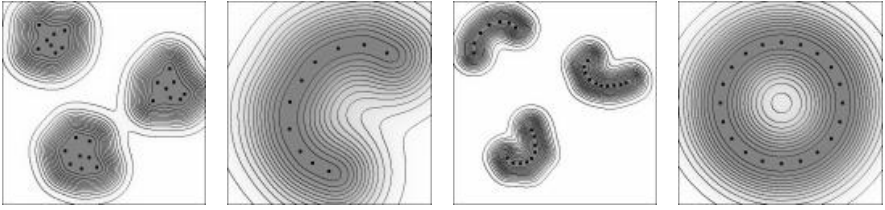
$$\mu^2 := \frac{1}{m} \sum_{i=1}^m \min_{j \neq i} |\mathbf{z}_i - \mathbf{z}_j|^2. \quad (13)$$

In order to get a smooth energy landscape, we propose to choose  $\sigma$  in the order of  $\mu$ . In practice we used

$$\sigma = 1.5 \mu \quad (14)$$

for most of our experiments. We chose this somewhat heuristic measure  $\mu$  for the following favorable properties:  $\mu$  is insensitive to the distance of clusters as long as each cluster contains more than one data point,  $\mu$  scales linearly with the data points, and  $\mu$  is robust with respect to the individual data points.

Given outliers in the training set, i.e. clusters with only one sample, one could refer to the more robust  $\mathcal{L}_1$ -norm or more elaborate robust estimators in (13). Since this is not the focus of our contribution, it will not be pursued here.



**Fig. 3. Density estimate (7) for artificial 2D data.** Distributions of variable shape are well estimated by the Gaussian hypothesis in feature space. We used the kernel (12) with  $\sigma = 1.5\mu$  — see definition (13).

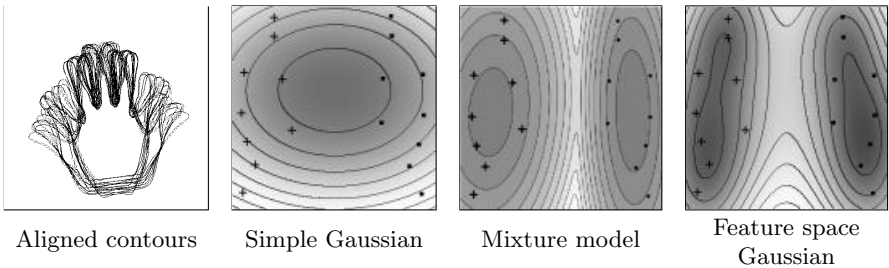
### 3.6 Density Estimate for Silhouettes of 2D and 3D Objects

Although energy (7) is quadratic in the space  $Y$  of mapped points, it is generally not convex in the original space, showing several minima and level lines of essentially arbitrary shape. Figure 3 shows artificial 2D data and the corresponding lines of constant energy  $E_\phi(\mathbf{z})$  in the original space.

For a set of binarized views of objects we automatically fit a closed quadratic spline curve around each object. All spline curves have  $N=100$  control points, set equidistantly. The polygons of control points  $\mathbf{z} = (x_1, y_1, x_2, y_2, \dots, x_N, y_N)$  are aligned with respect to translation, rotation, scaling and cyclic permutation. This data was used to determine the density estimate  $E_\phi(\mathbf{z})$  in (11).

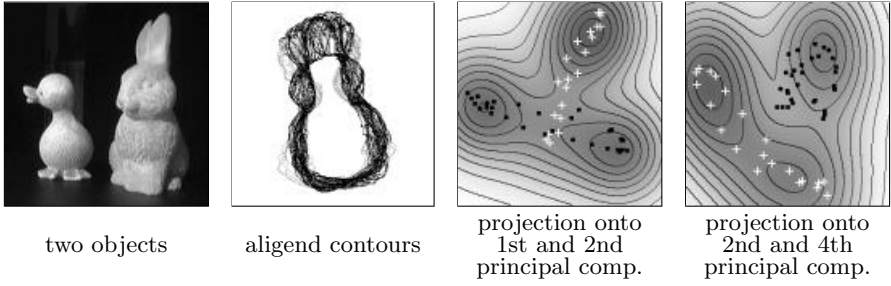
For the visualization of the density estimate and the training shapes, all data was projected onto two of the principal components of a linear PCA. Note that due to the projection, this visualization only gives a very rough sketch of the true distribution in the 200-dimensional shape space.

Figure 4 shows density estimates for a set of right hands and left hands. The estimates correspond to the hypotheses of a simple Gaussian in the original space, a mixture of Gaussians and a Gaussian in feature space. Although both



**Fig. 4. Model comparison:** density estimates for a set of left (+) and right (•) hands, projected onto the first two principal components. **From left to right:** aligned contours, simple Gaussian, mixture of Gaussians, Gaussian in feature space (7). Both the mixture model and the Gaussian in feature space capture the two-class structure of the data. However, the estimate in feature space is unsupervised and produces level lines which are not necessarily ellipses.





**Fig. 5. Density estimate for views of two 3D objects:** the training shapes of the duck (white +) and the rabbit (black •) form distinct clusters in shape space which are well captured by the energy level lines shown in appropriate 2D projections.

the mixture model and our estimate in feature space capture the two distinct clusters, there are several differences: firstly the mixture model is supervised — the number of classes and the class membership must be known — and secondly it only allows level lines of elliptical shape, corresponding to the hypothesis that each cluster by itself is a Gaussian distribution. The model of a Gaussian density in feature space does not assume any prior knowledge and produces level lines which capture the true distribution of the data even in the case that it does not correspond to a sum of hyperellipsoids.

This is demonstrated on a set of training shapes which correspond to different views of two 3D objects. Figure 5 shows the two objects, their contours after alignment and the level lines corresponding to the estimated energy density (7) in appropriate 2D projections.

## 4 Nonlinear Shape Statistics in Mumford–Shah Based Segmentation

### 4.1 Minimization by Gradient Descent

Energy (7) measures the similarity of a shape  $C(\mathbf{z})$  parameterized by a control point vector  $\mathbf{z}$  with respect to a set of training shapes. For the purpose of segmentation, we combine this energy as a shape energy  $E_{shape}$  with the Mumford–Shah energy (2) in the variational approach (3).

The total energy (3) must be simultaneously minimized with respect to the control points defining the contour and with respect to the segmenting grey values  $u_i$  and  $u_o$ . Minimizing the modified Mumford–Shah functional (2) with respect to the contour  $C$  (for fixed  $u_i$  and  $u_o$ ) results in the evolution equation

$$\frac{\partial C(s, t)}{\partial t} = -\frac{dE_{MS}}{dC} = -(e_s^+ - e_s^-) \cdot \mathbf{n}_s + \nu \frac{d^2 C}{ds^2}, \quad (15)$$

where the terms  $e_s^+$  and  $e_s^-$  denote the energy density  $e_s^{+/-} = (f - u_{i/o})^2$ , inside and outside the contour  $C(s)$ , respectively, and  $\mathbf{n}_s$  denotes the outer normal

vector on the contour. The two constants  $u_i$  and  $u_o$  are updated in alternation with the contour evolution to be the mean grey value of the adjoining regions  $\Omega_i$  and  $\Omega_o$ . The contour evolution equation (15) is transformed into an evolution equation for the control points  $\mathbf{z}$  by introducing definition (1) of the contour as a spline curve. By discretizing on a set of nodes  $s_j$  along the contour we obtain a set of coupled linear differential equations. Solving for the  $x$ -coordinate of the  $i$ -th control point and including the term induced by the shape energy we obtain:

$$\frac{dx_i(t)}{dt} = (\mathbf{B}^{-1})_{ij} [(e_{s_j}^+ - e_{s_j}^-)n_x - \nu(x_{j-1} - 2x_j + x_{j+1})] - \alpha \left[ \frac{dE_{shape}(\mathbf{z})}{d\mathbf{z}} \right]_{2i-1}, \quad (16)$$

where summation over  $j$  is assumed. The cyclic tridiagonal matrix  $\mathbf{B}$  contains the spline basis functions evaluated at these nodes, and  $n_x$  denotes the  $x$ -component of the normal vector on the contour. An expression similar to (16) holds for the  $y$ -coordinate of the  $i$ -th control point.

The three terms in the evolution equation (16) can be interpreted as follows: the first term pulls the contour towards the object in the image, thus minimizing the grey value variance in the adjoining regions. The second term pulls each control point towards its respective neighbors, thus minimizing the length of the contour. And the third term pulls the control point vector towards the nearest cluster of probable shapes, which minimizes the shape energy.

## 4.2 Invariance in the Variational Framework

By construction, the density estimate (7) is not invariant with respect to translation, scaling and rotation of the shape  $C(\mathbf{z})$ . We therefore propose to eliminate these degrees of freedom in the following way: since the training shapes were aligned to their mean shape  $\mathbf{z}_0$  with respect to translation, rotation and scaling and then normalized to unit size, we shall do the same to the argument  $\mathbf{z}$  of the shape energy before applying our density estimate  $E_\phi$ .

We therefore define the shape energy by

$$E_{shape}(\mathbf{z}) = E_\phi(\tilde{\mathbf{z}}), \quad \text{with } \tilde{\mathbf{z}} = \frac{R_\theta \mathbf{z}_c}{|R_\theta \mathbf{z}_c|}, \quad (17)$$

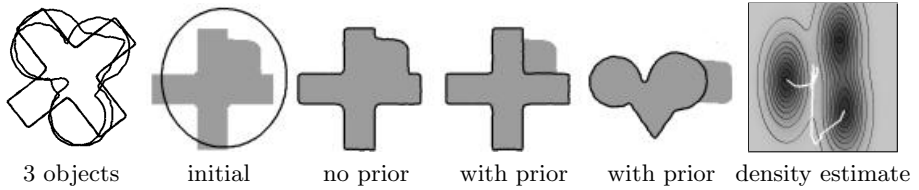
where  $\mathbf{z}_c$  denotes the control point vector after centering, and  $R_\theta$  denotes the optimal rotation of the control point polygon  $\mathbf{z}_c$  with respect to the mean shape  $\mathbf{z}_0$ . We will not go into details about the derivation of  $R_\theta$ . A similar derivation can be found in [26]. The final result is given by the formula:

$$\tilde{\mathbf{z}} = \frac{M \mathbf{z}_c}{|M \mathbf{z}_c|}, \quad \text{with } M = I_n \otimes \begin{pmatrix} \mathbf{z}_0^t \mathbf{z}_c & -\mathbf{z}_0 \times \mathbf{z}_c \\ \mathbf{z}_0 \times \mathbf{z}_c & \mathbf{z}_0^t \mathbf{z}_c \end{pmatrix},$$

where  $\otimes$  denotes the Kronecker product and  $\mathbf{z}_0 \times \mathbf{z}_c = \mathbf{z}_0^t R_{\pi/2} \mathbf{z}_c$ .

The last term in the contour evolution equation (16) is now calculated by applying the chain rule:

$$\frac{dE_{shape}(\mathbf{z})}{d\mathbf{z}} = \frac{dE_\phi(\tilde{\mathbf{z}})}{d\tilde{\mathbf{z}}} \cdot \frac{d\tilde{\mathbf{z}}}{d\mathbf{z}}.$$



**Fig. 6. Segmentation of artificial objects (left) with nonlinear shape prior:** the *same* prior can encode very different shapes. Introduction of the shape prior upon stationarity of the contour causes the contour to evolve normal to the level lines of constant energy into the nearest local minimum, as indicated by the white curves in the projected density estimate (right).

Since this derivative can be calculated analytically, no additional parameters enter the above evolution equation to account for scale, rotation and translation.

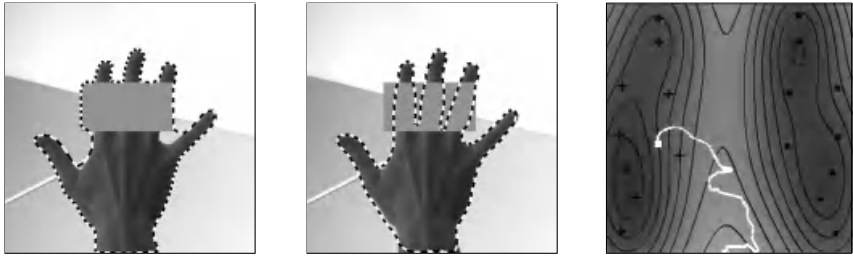
Other authors often propose to explicitly model a translation, an angle and a scale and minimize with respect to these quantities (e.g. by gradient descent). In our opinion this has several drawbacks: firstly it introduces four additional parameters, which makes numerical minimization more complicated — parameters to balance the gradient descent must be chosen. Secondly this approach mixes the degrees of freedom corresponding to scale, rotation and shape deformation. And thirdly potential local minima may be introduced by the additional parameters. On several segmentation tasks we were able to confirm these effects by comparing the two approaches.

Since there exists a similar closed form solution for the optimal alignment of two polygons with respect to the affine group [26], the above approach could be extended to define a shape prior which is invariant with respect to affine transformations. However, we do not elaborate this for the time being.

### 4.3 Coping with Multiple Objects and Occlusion

Compared to the linear case (4), the nonlinear shape energy (7) is no longer convex. In general it has several minima corresponding to different clusters of familiar contours. Minimization by gradient descent will end up in the nearest local minimum. In order to obtain a certain independence of the shape prior from the initial contour, we propose to first minimize the image energy  $E_{MS}$  by itself until stationarity and to then include the shape prior  $E_{shape}$ , after performing the cyclic permutation of control points which — given the optimal similarity transformation — best aligns the current contour with the mean of the training shapes. This approach guarantees that we will extract as much information as possible from the image before “deciding” which of the different clusters of accepted shapes the obtained contour resembles most.

Figure 6 shows a simple example of three artificial objects. The shape prior (17) was constructed on the three aligned silhouettes shown on the left. The next images show the initial contour for the segmentation of a partially occluded image of object 1, the final segmentation without prior knowledge, the final



Segmentation without prior    Segmentation with prior    Projected density estimate

**Fig. 7. Segmentation with a nonlinear shape prior containing right (+) and left (•) hands** — shown in the projected energy plot on the right. The input image is a right hand with an occlusion. After the Mumford–Shah segmentation becomes stationary (left image), the nonlinear shape prior is introduced, and the contour converges towards the final segmentation (center image). The contour evolution in its projection is visualized by the white curve in the energy density plot (right). Note that the final segmentation (white box) does not correspond to any of the training silhouettes, nor to the minimum (i.e. the most probable shape) of the respective cluster.

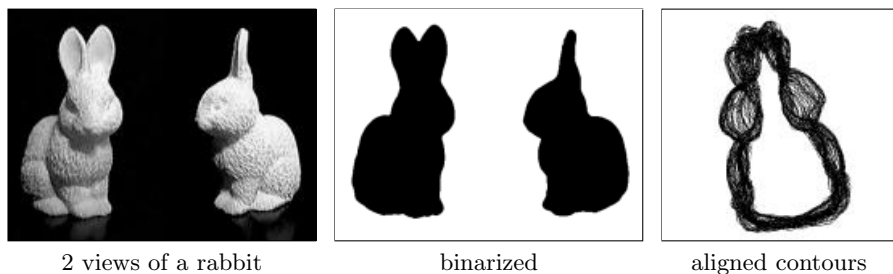
segmentation after introducing the prior, and a segmentation with *the same* prior for an occluded version of object 2.

The final image (Figure 6, right) shows the training shapes and the density estimate in a projection onto the first two axes of a PCA. The white curves correspond to the path of the segmenting contour from its initialization to its converged state for the two segmentation processes respectively. Note that upon introducing the shape prior the corresponding contour descends the energy landscape in direction of the negative gradient to end up in one of the minima. The example shows that the nonlinear shape prior can well separate different objects without mixing them as in the simple Gaussian hypothesis. Since each cluster in this example contains only one view for the purpose of illustration, the estimate (14) for the kernel width  $\sigma$  does not apply; instead we chose a smaller granularity of  $\sigma = \mu/4$ .

#### 4.4 Segmentation of Real Objects

The following example is an application of the nonlinear shape statistics to silhouettes of real objects. The training set consisted of nine right and nine left hands, shown together with the estimated energy density in a projection onto the first two principal components in Figure 7, right side.

Rather than mixing the two classes of right and left hands, the shape prior clearly separates several clusters in shape space. The final segmentations without (left) and with (center) prior shape knowledge show that the shape prior compensates for occlusion by filling up information where it is missing. Moreover, the statistical nature of the prior is demonstrated by the fact that the hand in the image is not part of the training set. This can be seen in the projection (Figure 7, right side), where the final segmentation (white box) does not correspond to any of the training contours (black crosses).



**Fig. 8.** Example views and binarization used for estimating the shape density.

#### 4.5 Tracking 3D Objects with Changing Viewpoint

In the following we present results of applying the nonlinear shape statistics for an example of tracking an object in 3D with a prior constructed from a large set of 2D views. We binarized 100 views of a rabbit — two of them and the respective binarizations are shown in Figure 8. For each of the 100 views we automatically extracted the contours and aligned them with respect to translation, rotation, scaling and cyclic reparameterization of the control points. We calculated the density estimate (7) and the induced shape energy (17).

In a film sequence we moved and rotated the rabbit in front of a cluttered background. Moreover, we artificially introduced an occlusion afterwards. We segmented the first image by the modified Mumford–Shah model until convergence before the shape prior was introduced. The initial contour and the segmentations without and with prior are shown in Figure 9. Afterwards we iterated 15 steps in the gradient descent on the full energy for each frame in the sequence.<sup>2</sup>

Some sample screen shots of the sequence are shown in Figure 10. Note that the viewpoint changes continuously.

The training silhouettes are shown in 2D projections with the estimated shape energy in Figure 11. The path of the evolving contour during the entire sequence corresponds to the white curve. The curve follows the distribution of training data well, interpolating in areas where there are no training silhouettes. Note that the intersections of the curve and of the training data in the center (Figure 11, left side) are only due to the projection on 2D. The results show that — given sufficient training data — the shape prior is able to capture fine details such as the ear positions of the rabbit in the various views. Moreover, it generalizes well to novel views not included in the training set and permits a reconstruction of the occluded section throughout the entire sequence.

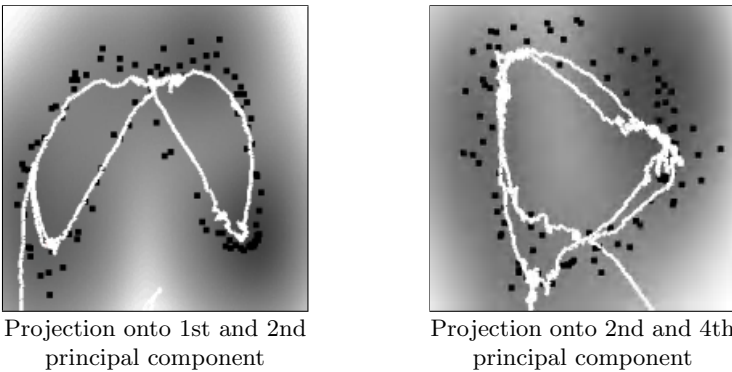
<sup>2</sup> The gradient of the shape prior in (16) has a complexity of  $O(rmn)$ , where  $n$  is the number of control points,  $m$  is the number of training silhouettes and  $r$  is the eigenvalue cutoff. For input images of 83 kpixels and  $m=100$ , we measured an average runtime per iteration step of 96ms for the prior, and 11ms for the cartoon motion on a 1.2 GHz AMD Athlon. This permitted to do 6 iterations per second. Note, however, that the relative weight of the cartoon motion increases with the size of the image: for an image of 307 kpixels the cartoon motion took 100ms per step.



**Fig. 9. Begin of the tracking sequence:** initial contour, segmentation without prior, segmentation upon introducing the nonlinear prior on the contour.



**Fig. 10. Sample screen shots from the tracking sequence.**



**Fig. 11. Tracking sequence visualized:** Training data ( $\bullet$ ), estimated energy density and the contour evolution (white curve) in appropriate 2D projections. The contour evolution is restricted to the valleys of low energy induced by the training data.

## 5 Conclusion

We presented a variational integration of nonlinear shape statistics into a Mumford–Shah based segmentation process. The statistics are derived from a novel method of density estimation which can be considered as an extension of the

kernel PCA approach to a probabilistic framework. The original training data is nonlinearly transformed to a feature space. In this higher dimensional space the distribution of the mapped data is estimated by a Gaussian density. Due to the strong nonlinearity, the corresponding density estimate in the original space is highly non-Gaussian, allowing several shape clusters and banana- or ring-shaped data distributions.

We integrated the nonlinear statistics as a shape prior in a variational approach to segmentation. We gave details on appropriate estimations of the involved parameters. Based on the explicit representation of the contour, we proposed a closed-form, parameter-free solution for the integration of invariance with respect to similarity transformations in the variational framework.

Applications to the segmentation of static images and image sequences show, that the nonlinear prior can capture even small details of shape variation without mixing different views. It copes for misleading information due to noise and clutter, and it enables the reconstruction of occluded parts of the object silhouette. Due to the statistical nature of the prior, a generalization to novel views not included in the training set is possible. Finally we showed examples where the 3D structure of an object is encoded through a training set of 2D projections.

By projecting onto the first principal components of the data, we managed to visualize the training data and the estimated shape density. The evolution of the contour during the segmentation of static images and image sequences can be visualized by a projection into this density plot and by animations. In this way we verified that the shape prior effectively restricts the contour evolution to the submanifold of familiar shapes.

**Acknowledgments.** We thank P. Bouthemy and his group, C. Kervrann and A. Trubuil for stimulating discussions and hospitality.

## References

1. M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
2. B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of the 5th Annual ACM Workshop on Comput. Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.
3. B. Chalmond and S. C. Girard. Nonlinear modeling of scattered multivariate data and its application to shape change. *IEEE PAMI*, 21(5):422–432, 1999.
4. T. Cootes and C. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. In J. Illingworth, editor, *BMVC*, pages 639–648, 1993.
5. T.F. Cootes and C.J. Taylor. A mixture model for representing shape variation. *Image and Vis. Comp.*, 17(8):567–574, 1999.
6. R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, Inc., New York, 1953.
7. D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics via kernel spaces. In B. Radig and S. Florczyk, editors, *Pattern Recognition*, volume 2191 of *LNCS*, pages 269–276, Munich, Germany, Sept. 2001. Springer.

8. D. Cremers, C. Schnörr, and J. Weickert. Diffusion–snakes: Combining statistical shape knowledge and image information in a variational framework. In *IEEE First Workshop on Variational and Level Set Methods*, pages 137–144, Vancouver, 2001.
9. D. Cremers, C. Schnörr, J. Weickert, and C. Schellewald. Diffusion–snakes using statistical shape knowledge. In G. Sommer and Y.Y. Zeevi, editors, *Algebraic Frames for the Perception-Action Cycle*, volume 1888 of *LNCS*, pages 164–174, Kiel, Germany, Sept. 10–11, 2000. Springer.
10. G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, San Diego, 1997.
11. D. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.
12. T. Heap and D. Hogg. Automated pivot location for the cartesian-polar hybrid point distribution model. In *BMVC*, pages 97–106, Edinburgh, UK, Sept. 1996.
13. T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, Colchester, UK, 1997.
14. C. Kervrann and F. Heitz. A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60:173–195, 5 1998.
15. M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. Conf. Computer Vis. and Pattern Recog.*, volume 1, pages 316–323, Hilton Head Island, SC, June 13–15, 2000.
16. B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proc. IEEE Internat. Conf. on Comp. Vis.*, pages 786–793, 1995.
17. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.
18. N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: a level set approach. In D. Vernon, editor, *ECCV*, volume 1843 of *LNCS*, pages 224–240. Springer, 2000.
19. E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.
20. F. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.
21. S. Roweis. Em algorithms for PCA and SPCA. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 626–632, Cambridge, MA, 1998. MIT Press.
22. B. Schölkopf, S. Mika, Smola A., G. Rätsch, and Müller K.-R. Kernel PCA pattern reconstruction via approximate pre-images. In L. Niklasson, M. Boden, and T. Ziemke, editors, *ICANN*, pages 147–152, Berlin, Germany, 1998. Springer.
23. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
24. M. Tipping. Sparse kernel principal component analysis. In *Advances in Neural Information Processing Systems 13*, Vancouver, Dec. 2001.
25. M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. Technical Report Woe-19, Neural Computing Research Group, Aston University, 1997.
26. M. Werman and D. Weinshall. Similarity and affine invariant distances between 2d point sets. *IEEE PAMI*, 17(8):810–814, 1995.
27. S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE PAMI*, 18(9):884–900, 1996.



# Class-Specific, Top-Down Segmentation

Eran Borenstein and Shimon Ullman\*

Dept. of Computer Science and Applied Math  
The Weizmann Institute of Science  
Rehovot 76100, Israel  
{boren, shimon}@wisdom.weizmann.ac.il

**Abstract.** In this paper we present a novel class-based segmentation method, which is guided by a stored representation of the shape of objects within a general class (such as horse images). The approach is different from bottom-up segmentation methods that primarily use the continuity of grey-level, texture, and bounding contours. We show that the method leads to markedly improved segmentation results and can deal with significant variation in shape and varying backgrounds. We discuss the relative merits of class-specific and general image-based segmentation methods and suggest how they can be usefully combined.

**Keywords:** Grouping and segmentation; Figure-ground; Top-down processing; Object classification

## 1 Introduction

A major goal of image segmentation is to identify structures in the image that are likely to correspond to scene objects. Current approaches to segmentation mainly rely on image-based criteria, such as the grey level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours. In this work we describe a segmentation method that is guided primarily by high-level information and the use of class-specific criteria. The motivation for using such class-based criteria to supplement the traditional use of image-based criteria in segmentation has two parts. First, it stems from the fact that although recent image-based segmentation algorithms provide impressive results, they still often fail to capture meaningful and at times crucial parts. Second, evidence from human vision indicates that high-level, class-based criteria play a crucial role in the ability to segment images in a meaningful manner (e.g. [11],[10],[9],[8]), suggesting that the incorporation of such methods will help improve the results of computer vision segmentation algorithms.

Figure 1 demonstrates some of the major difficulties encountered by image-based segmentation algorithms. An appropriate segmentation algorithm should group together the dark and light regions of the horse (left), and separate the man from the horse, despite the grey level similarity (right). Figure 2 shows

---

\* This research was supported by the Israel Ministry of Science under the Scene Teleportation Research Project and by the Moross Laboratory at the Weizmann Institute of Science.

the results obtained when applying a state of the art image-based segmentation algorithm [15] to these and other images. The segmentations demonstrate some of the inherent difficulties of an image-based approach, including the splitting of object regions and the merging of object parts with background regions. These shortcomings are due to unavoidable ambiguities that cannot be solved without prior knowledge about the object class at hand. This paper presents an approach that addresses these difficulties, in which a simple representation of object classes in memory is used to guide the segmentation process, leading to markedly improved segmentation of images containing familiar objects (Fig. 3).

The overall structure of the paper is as follows. Section 2 briefly reviews past approaches. Section 3 provides an overview of our approach, and how information about object shapes is represented in memory and used for segmentation. Section 4 describes the approach in detail: 4.1-4.3 describe the segmentation criteria used by our method. 4.4 describes the algorithm that segments class-images according to these criteria. Section 5 shows results, Sect.6 contains a final discussion and conclusions.

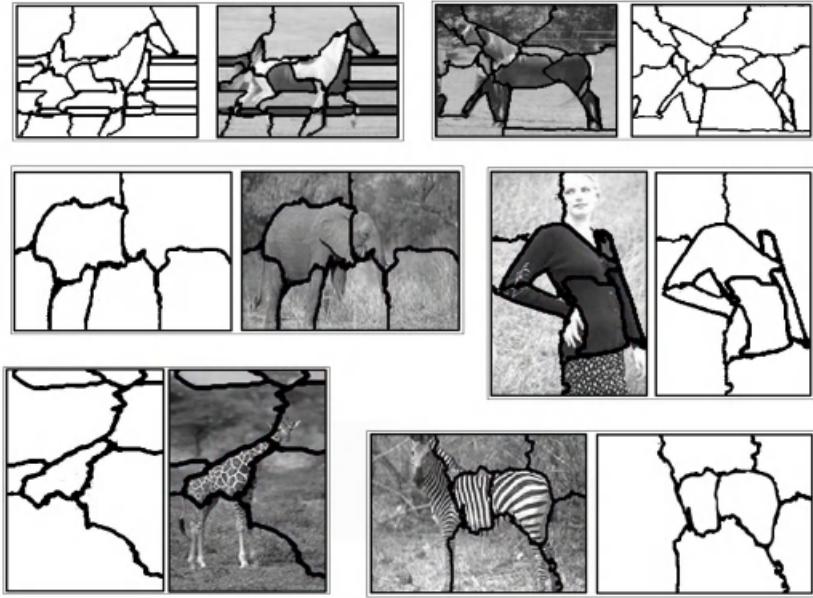


**Fig. 1.** Segmentation difficulties: the same object can contain markedly different regions (left), while neighboring objects may contain regions that are similar in color and texture (right).

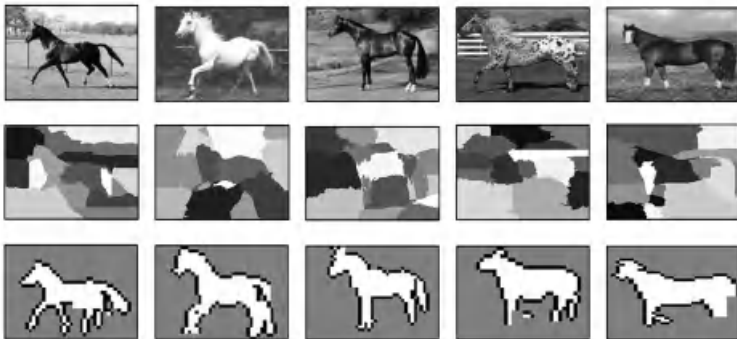
## 2 Brief Review of Segmentation Approaches

Most of the current approaches to segmentation rely primarily on image-based criteria, such as color, grey level, or texture uniformity of image regions (e.g. [16],[4],[2]); the smoothness and continuity of their bounding contours (e.g. [7]); or a combination of these (e.g. [6]). The region-based approaches merge and split image regions according to specific criteria. Merging approaches recursively merge similar regions (e.g. [14],[1]). “Divide & Conquer” approaches recursively split regions into distinct sub-regions (e.g. [12],[15]). Contour-based approaches emphasize the properties of region boundaries, such as continuity, smoothness, length, curvature, and shape.

Somewhat closer to our goal is the work on deformable templates (e.g. [18]), where the template parameters are used to match a model with an object image. This approach usually assumes approximate initial correspondence between



**Fig. 2.** Typical results of low-level segmentation. Objects are broken into sub-regions, and some of these sub-regions are merged with the background. (More results can be seen at <http://www.cs.berkeley.edu/~doron/software/ncuts/results/>)



**Fig. 3.** Low-level vs. class-specific segmentation of horse images. Top: input images (170x120 pixels). Middle: low-level segmentation into sub-regions, as given by the normalized cuts algorithm [15]. Bottom: figure-ground segmentation map given by the algorithm described in this paper (input resolution reduced to 40x30).

the model and its image and therefore it can be used as a final stage in our segmentation but not as the main segmentation process.

Recent psychological studies of image segmentation by humans indicate strong effects of prior familiarity with specific shapes on image segmentation. For example, a number of behavioral studies have shown that subjects are more likely to regard a familiar region as “figure” than a less familiar region, indicating that object recognition facilitates segmentation [11],[10]. Developmental studies [9],[8] show that figure-ground segregation in four month-old infants is also affected by the presence of already familiar shapes.

### 3 Overview of the Approach

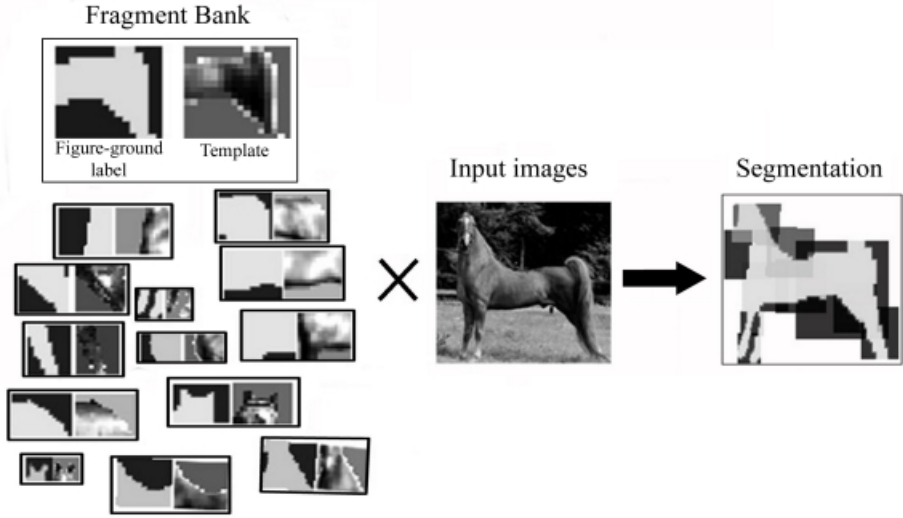
The general thrust behind our class-based segmentation approach is to use known shape characteristics of objects within a given class to guide the segmentation process. The main difficulty in this approach stems from the large variability of shapes within a given class of objects. We would like to somehow capture the common characteristics of a set of shapes within a class, for example horse images, and then use this information to segment novel images. To address this problem our approach uses a fragment-based representation of object classes. Similar fragments have been used in the past for object classification [13][17] but not for segmentation. Given an image containing a certain object, we use fragments previously extracted from images of the same object class to produce a consistent cover of the novel object. This cover defines a figure-ground map that associates each pixel in the input image with the likelihood of belonging to an object or background. (Fig. 4).

The construction of an object by fragments is somewhat similar to the assembly of a jigsaw puzzle, where we try to put together a set of pieces such that their templates form an image similar to a given example. A common strategy is to start with the easiest pieces (e.g. corners) and proceed by connecting additional pieces that match in shape, color, edges, texture, etc. In some cases, as information accumulates along this process, pieces must be replaced: locally these pieces provide good matches, but the global structure adds constraints that reject the local matches.

The next several sections describe our segmentation algorithm in detail. We first describe the fragment representation — how the class fragments are represented in memory, and how they are extracted from sample images.

#### 3.1 Fragment Representation in Memory

In this section we describe the fragment-based representation used for segmentation. The goal of this representation is to cover as closely as possible the images of different objects from a given class, using a set of more primitive shapes. We therefore need to identify useful “building blocks,” a collection of components that can be used to identify and delineate the boundaries of objects in the class. To find such common components we look for image fragments that are strongly correlated with images containing the desired object class — they show a high



**Fig. 4.** Overview of the approach. Given an input image containing an object from a given class (in this case, a horse) we use class fragments stored in memory to produce a cover of the target image. Each fragment consists of a template and a figure-ground label. The cover defines the figure-ground segmentation.

similarity measure with regions from images containing this desired object class but not with others. Based on the Neyman-Pearson decision theory, optimal fragments can be defined as fragments with maximal frequency (hit rate) within the class, subject to the constraint that the frequency of false detection in non-class images (false alarms) does not exceed a fixed limit (e.g. [3]).

Our search for optimal fragments therefore proceeds in three stages. Stage 1 starts from a set of training images divided into class images (C) and non-class images (NC) and then generates a large number of candidate fragments. We simply extract from the images in C a large number of rectangular sub-images, these sub-images can vary in size and range from  $\frac{1}{50}$  to  $\frac{1}{7}$  of the object size. In stage 2, which is the crucial step, we compare the distribution of each fragment in the class and non-class training images. For a given fragment  $F_i$ , we measure the strength of the response  $S_i$  in C and NC.  $S_i$  is defined in a standard way: we correlate  $F_i$  with each image  $I$  in C and NC (normalized correlation) and take the maximum value over  $I$ . To reach a fixed level of false alarms  $\alpha$  in non-class images we determine a threshold  $\theta_i$  for  $F_i$  by the criterion:

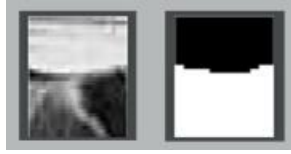
$$p(S_i > \theta_i | NC) \leq \alpha \quad (1)$$

This has the advantage of automatically fixing an optimal detection threshold for each fragment. In stage 3 we order the fragments by their hit rate  $p(S_i > \theta_i | C)$  and select the  $K$  best ones where  $K$  determines the size of the fragment set.

To be used for segmentation, we add two factors to each fragment: a *figure-ground label* and a *reliability value*. The figure-ground label marks each pixel

in the fragment as figure or ground. The figure label is learned at present by comparing the fragment to the source image in the data base from which it was derived. We assume that in this limited set the figure-ground information has been estimated, for example by relative motion of the figure with respect to background. An alternative is to label pixels in the fragment according to their grey level variability in the class database. Figure pixels are similar across images and have low-variability, while background pixels show high variability.

The reliability of a fragment measures the extent to which the fragment is class-specific, measured by its hit rate  $p(S_i > \theta_i | C)$ . These two factors are essential to our segmentation process, as will be later demonstrated.



**Fig. 5.** Fragment representation in memory: Each fragment is represented by its grey level template (left) and figure-ground label (right).

## 4 Segmentation by Optimal Cover

The main stage of the class-based segmentation algorithm consists of covering an image with class-based fragments and using the cover to delineate the figure boundaries. To accomplish this we seek an optimal *cover* of the image in terms of the fragments. A cover is an assignment of fragments to positions in the image  $I$ , with each fragment being defined as either “present” in the image along with its designated position  $p_i$  in the cover or “absent” from the image:

$$I \xrightarrow{i=1 \dots K} \text{cover} \begin{cases} f_i = 1, p_i = (x_i, y_i) \\ f_i = 0, p_i = \phi \end{cases} \quad (2)$$

Given a cover, we can compute the quality of the cover, which is a function of the individual match of fragments with the image, the consistency of the cover, and the reliability of the participating fragments. The following sections describe each of these factors in turn and an algorithm to find a locally optimal cover according to these criteria.

### 4.1 Individual Match

The individual match measures the similarity between fragments and the image regions that they cover. We use a similarity measure that combines region correlation with edge detection. This combination which is not commonly used,

is designed to meet the special requirements of segmentation as illustrated in Fig. 7.

A key feature of our fragment representation is the elimination of background noise. Using the figure-ground label it is possible to exclude background pixels from the similarity measure, thereby reducing background noise. Since this exclusion results in the loss of the contour features, an edge detector is added to capture the transition from figure to ground (Fig. 7). The edge presence is computed by using a normalized correlation between the figure-ground boundary and edges in the image (Fig. 6). The similarity measure  $s_i(p, I)$  between fragment  $F_i$  at image position  $p = (x, y)$  and an image  $I$  is defined in (3) and consists of two factors. The first factor –  $N_{\text{cor}}$  is the standard normalized correlation between the fragment and the image region, restricted to template pixels labeled as figure. The second term –  $E_{\text{dge}}$  is the edge detector response.

$$s_i(p, I) = w \cdot N_{\text{cor}}(p, I)|_{\substack{\text{Object} \\ \text{Pixels}}} + (1 - w) \cdot E_{\text{dge}}(p, I) \quad (3)$$

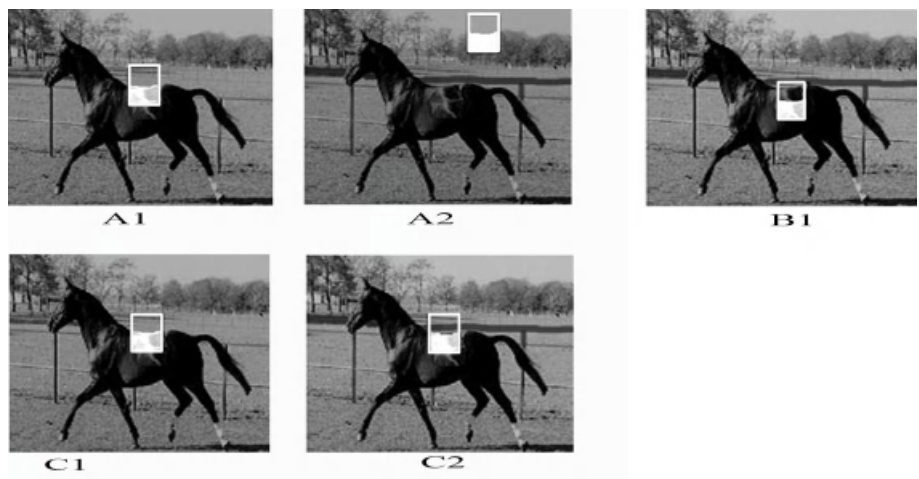


**Fig. 6.** The edge template (right) is derived from the boundary of the figure-ground label. The response to this template is the edge part  $E_{\text{dge}}$  in (3).

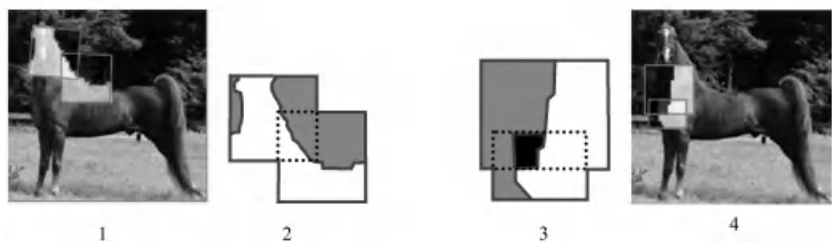
## 4.2 Consistency

In covering the image by shape fragments, the fragments should not only provide good local matches, but should also provide a consistent global cover of the shape. We therefore use a consistency criterion for the cover in the segmentation process. Since the fragments are highly overlapping, we define a consistency measure  $c_{ij}$  between a pair of overlapping fragments  $F_i$  and  $F_j$  that is proportional to the fraction of pixels labeled consistently by the two fragments (4). The maximum term in the denominator prevents overlaps smaller than a fixed value  $\mu_{ij}$  from contributing a high consistency term. This value is set to  $\frac{1}{10}$  of the maximum possible overlap size between the two fragments. Fig. 8 demonstrates two cases of overlapping horse fragments : one in which all the pixels are labeled consistently and one in which some of the pixels are labeled inconsistently.

$$c_{ij} = \frac{\# \text{ Consistent Overlapping Pixels}}{\max(\text{Total Overlap}, \mu_{ij})} \quad (4)$$



**Fig. 7.** The examples show why a combination of templates and boundary similarity are useful for segmentation. A1 – An image of a horse and a template of its back. A2 – Changes in the background reduce the similarity measure (measured using both the figure and ground parts of the template), causing the template to be more similar to another image region. B1 – In this example the template fit was measured using the figure part of the template only. This reduces background effects but also results in the loss of boundary information leading to inaccurate matches. C1,C2 – Adding an edge detector to the similarity measure yields a more stable similarity measure resulting in an accurate placement of the fragment regardless of background noise.



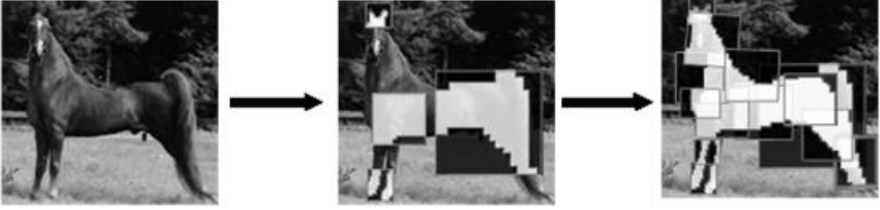
**Fig. 8.** Consistent (left) and inconsistent (right) cover by overlapping fragments. 1,4: the fragments, placed over the images, 2,3: figure-ground assignment of the fragments. Figure pixels are marked white, background pixels are grey. The inconsistent region is marked in black.



### 4.3 Fragment Reliability

The final criterion in determining an optimal cover is that of fragment reliability. Similar to a jigsaw puzzle, the task of piecing together the correct cover can be simplified by starting with some more “reliable” fragments, or anchor fragments, and then proceeding with less reliable fragments that are consistent with the initial cover (Fig. 9). Reliable fragments typically capture some distinguishing features of the shapes in the class and are unlikely to be found anywhere else. A fragment’s reliability is therefore evaluated by the likelihood ratio between the detection rate and the false alarm rate. As explained, we set the minimal threshold such that the false alarm rate does not exceed  $\alpha$ . We can therefore express this ratio using the detection rate and  $\alpha$ :

$$r_i = \frac{p(S_i > \theta_i | C)}{p(S_i > \theta_i | NC)} = \frac{\text{detection rate}}{\alpha} \quad (5)$$



**Fig. 9.** Reliable fragments guide the covering. Reliable fragments are used first (middle), and subsequently completed by less reliable ones (right).

### 4.4 The Cover Algorithm

A cover on an image by shape fragments determines a segmented figure. Among all possible covers we seek a cover (2) that maximizes the three criteria above, namely, individual match quality (3), consistency (4) and reliability (5). These three factors are therefore combined in the cover score:

$$cs = \underbrace{\sum_i r_i \cdot s_i \cdot f_i}_{\text{Individual match and reliability}} + \underbrace{\frac{1}{\lambda} \sum_{i,j} \beta_{ij} \cdot f_i \cdot f_j}_{\text{Consistency}} \quad (6)$$

The first term combines the match quality and reliability of the fragments, and the second penalizes inconsistent overlapping pairs. We define the interaction term  $\beta_{ij}$  between overlapping fragments as  $(c_{ij} - \beta) \cdot (r_i s_i + r_j s_j)$  where  $c_{ij}$  is the pairwise consistency defined above (4) and  $\beta$  a global constant that determines the magnitude of the penalty for insufficient consistency. For non-overlapping

pairs  $\beta_{ij}$  is defined as 0. The contribution of a single fragment  $F_k$  in this expression is obtained by summing up all the terms in (6) for which  $i = k$ . When a fragment contribution is negative, the score is improved by removing the fragment from the cover. Negative contribution indicates poor consistency of the fragment with other fragments and can happen only when  $c_{kj} < \beta - \lambda$  for at least one  $j$ . In our implementation  $\beta = 0.65, \lambda = 0.1$ .

The algorithm is iterative, but a small number of iterations (typically 2-3) are used. It is described in the Appendix, but the main stages are summarized next. At each stage, a small number  $M$  of good candidate fragments are identified. A subset of these  $M$  fragments, that maximally improve the current score, are selected and added to the cover. In addition, existing fragments that are inconsistent with the new match are removed. We use a small number of candidates ( $M = 15$ ) that allows us to check all  $2^M$  subsets and select the one with the highest score. The algorithm is guaranteed to converge to a local maximum since the score is bounded and increases at each iteration. To initialize the process, we select a sub-window within the image with the maximal concentration of reliable fragments. The similarity of all the reliable fragments is examined at 5 scales at all possible locations – giving a complexity which is linear in the number of reliable fragments, the number pixels of each image scale, and the number of scales. Given this information it is possible to pick the most reliable window and use the matched fragments inside as the initial  $M$  candidates for the cover. If the combined evidence from the reliable fragments falls below a classification threshold, the process terminates without producing a cover. In a system containing multiple classes that compete for segmentation (rather than just horse images), the class with the highest evidence will initiate the cover.

## 5 Experiments

We tested the algorithm on a database containing horse images. A bank of 485 fragments was constructed from a sample library of 41 horse containing images of size  $(40 \times 30)$  for which the figure-ground information was manually segmented. For each fragment we estimated  $p(S_i|C)$  and  $p(S_i|NC)$  by measuring the distribution of the fragments' similarity measure with 193 low-resolution images of horses and 253 low-resolution images of non-horses. Using these estimated distributions, the fragments were assigned their appropriate threshold and classified to 146 reliable and 339 non-reliable fragments. The algorithm was then tested on 176 novel horse images  $(40 \times 30)$  pixels. Examples are shown in Fig. 10 and compared to the results of a normalized-cuts segmentation algorithm [15] tested on the same images but with higher resolution  $(170 \times 120)$ . The algorithm obtains high-quality segmentations of figure from background for a variety of images. The algorithm can deal successfully with shape variations using a fixed repertoire of fragments extracted from the training set. The generalization to novel shapes is based in part on the use of multiple alternative fragments for the same object region, and in part in the flexibility in the fragments' arrangement. We also compared qualitatively the agreement between the figure regions produced by the algorithm and the figure region judged by humans. This can be expressed

by evaluating the ratio  $r = \frac{|S \cap F|}{|S \cup F|}$  where  $F$  is the human-segmented figure and  $S$  is the algorithm's segmented figure.  $|F|, |S|$  are the sizes of  $F, S$  and the size of the entire image  $|I| = 1$ . The maximal value of  $r$  is  $r = 1$  obtained only for perfect segmentation. The average score for the current algorithm was  $r = 0.71$ . The normalized-cuts algorithm, segmenting the images into two segments – figure and ground, gives much lower average score ( $r = 0.31$ ). The last value can also be compared with random segmentation (where  $|S|$  pixels are chosen randomly to be figure), which gives an average of  $r = 0.23$ . The most problematic figure regions were the horse legs, where variability is high. The initialization step was the most time consuming (about 30 seconds), where the similarity of every reliable fragment was examined on 5 different scales of the target image. Usually the algorithm converged after 2-3 iterations, with each iteration taking a few seconds, giving a total time of about 40 seconds per target image. We used Matlab 6.0 program on a Windows-NT, Pentium-600Mhz platform. The complexity of the algorithm is linear in the number of scales, number of fragments, and size of the fragment bank.

## 6 Discussion

The approach proposed in this paper emphasizes the role of high-level information or class-specific criteria in image segmentation. We present a class-specific segmentation method that successfully addresses ambiguities inherent to segmentation schemes based exclusively on image-based criteria.

Segmentation is obtained by covering the image with a subset of class-specific fragments and using this cover to delineate the figure boundaries. These fragments serve as class-specific shape primitives, or “building blocks,” and are used to handle a large variety of novel shapes within the class. The fragments are represented by a template together with a figure-ground label, and a reliability score.

The last two factors in this fragment representation are novel characteristics of our approach and are essential for segmentation. The figure-ground label is essential for: (a) constructing the figure-ground segmentation map; (b) defining a robust similarity measure that reduces background noise; and (c) defining the consistency between fragments. The fragment reliability enables us to detect key fragments that serve as initial anchors in the cover construction. In order to construct a complete cover we use both reliable and less reliable fragments. The less reliable fragments are constrained by the consistency relations of the cover and can therefore be used to complete the segmentation of difficult regions.

Compared with other schemes that apply segmentation and classification in sequence, in our scheme the two processes are intimately linked. The initial evidence from reliable class fragments is used to select the most likely class that serves to initiate the cover, and the final classification decision may depend on the segmentation result.

The algorithm results in the segmentation of images into two regions, figure and ground, in contrast with image-based segmentation algorithms that usually segment the image into multiple regions. The extraction of objects from these



**Fig. 10.** Segmentation results arranged in 3 groups of 4 rows. First row in each group: input images. Second row: results obtained from low-level segmentation. Third row: class-based segmentation to figure and ground. Fourth row: segmentation superimposed on the input images (boundary in red).

images is challenging because objects are often broken into sub-regions, and some of these sub-regions merge with the background. The main difficulty faced by the current algorithm lies in covering highly variable parts, such as the horse's legs and tail. This difficulty is due in part to image processing limitations and the use of relatively low-resolution images. In the future we intend to use higher resolution images or a pyramid of image segments at different scales.

Compared with the class-based segmentation, traditional image-based segmentation methods have two advantages. First, when they detect the correct figure boundaries, they can determine these boundaries with higher accuracy since they are guided directly by image discontinuities. Second, image-based algorithms are general and do not require class-specific information. The relative merits of class-specific and image-based segmentation methods suggest that they can be usefully combined into an integrated scheme. For example, image-based segmentation can be used to identify salient regions and direct class-based segmentation to these regions. At the final segmentation stage, figure boundaries produced by class-based segmentation could be refined by image-based methods (e.g. [5]) resulting in a robust and accurate delineation of object boundaries that cannot be achieved by either method alone.

## 7 Appendix: The Class-Based Segmentation Algorithm

### Pre-processing:

- For all reliable fragments  $F_i$ , compute  $s_i(p_j, I)$  for all image positions  $p_j$  and all scales (see(3)).
- Set  $s(p_j) = \max_i [r_i \cdot s_i(p_j)]$  (best fragment at  $p_j$ ).
- For each image window  $W$ :  
Pick in  $W$  at most  $M$  positions  $p_j$  with maximal values of  $s(p_j)$ . Define the score of the window  $W_{\text{score}} = \sum_{p_j} s(p_j)$ .

### Initialization:

- Choose window  $W_{\text{max}}$  to be the window with maximal  $W_{\text{score}}$  together with its fragments. These fragments compose the initial covering candidates  $B_{\text{cand}}$ .
- Set the current cover  $B_{\text{cover}}$  to be empty.

### Choosing the new covering fragments:

- With all fragments ( $F_j \in B_{\text{cover}}$ ) fixed to  $f_j = 1$ , assign the candidate fragments  $F_i \in B_{\text{cand}}$  to  $f_i = \{0, 1\}$  such that (6) is maximized.
- Add to  $B_{\text{cover}}$  all candidate fragments with  $f_i = 1$  and remove all the fragments that reduce the score of (6).
- Use  $B_{\text{cover}}$  to construct the figure ground segmentation map.
- If  $B_{\text{cover}}$  did not change, then stop.

Updating the candidate fragments sets:

- From all fragments in  $W_{\max}$  select the  $M$  with the highest score (individual match and consistency with  $B_{\text{cover}}$ ).
- Go to choosing the covering fragments.

**References**

1. K. Cho and P. Meer. Image segmentation from consensus information. *Computer Vision and Image Understanding: CVIU*, 68(1):72–89, 1997.
2. J.M.H. du Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recognition*, 23:291–309, 1990.
3. R. Duda, P. Hart, and D. Stork. Pattern classification, 2001.
4. A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.
5. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
6. T. Leung and J. Malik. Contour continuity in region based image segmentation. In *Fifth Euro. Conf. Computer Vision*, Freiburg, Germany, 1998.
7. D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, 1985.
8. A. Needham. Object recognition and object segregation in 4.5-month-old infants. *Journal of Experimental Child Psychology*, 78:3–24, 2001.
9. A. Needham and R. Baillargeon. Effects of prior experience in 4.5-month-old infants' object segregation. *Infant Behaviour and Development*, 21:1–24, 1998.
10. M.A. Peterson. Object recognition processes can and do operate before figure-ground organization. *Current Directions in Psychological Science*, 3:105–111, 1994.
11. M.A. Peterson and B.S. Gibson. Shape recognition contributions to figure-ground organization in three-dimensional displays. *Cognitive Psychology*, 25:383–429, 1993.
12. M. Pietikainen, A. Rosenfeld, and I. Walter. Split and link algorithms for image segmentation. *Pattern Recognition*, 15(4):287–298, 1982.
13. E. Sali and S. Ullman. Combining class-specific fragments for object classification. In *Proc. 10th British Machine Vision Conference*, volume 1, pages 203–213, 1999.
14. E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 70–77, South Carolina, 2000.
15. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, 1997.
16. A. Treneau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30, No. 7:1191–1203, 1997.
17. S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment based approach to object representation and classification. In *Proc. of 4th international workshop on visual form*, Capri, Italy, 2001.
18. A. Yuille and P. Hallinan. Deformable templates. In *A. Blake and A. Yuille, editors, Active Vision*, pages 21–38, MIT press, 1992.

# Quasi-Dense Reconstruction from Image Sequence

Maxime Lhuillier and Long Quan

Department of Computer Science  
Hong Kong University of Science and Technology  
Clear Water Bay, Kowloon, Hong Kong SAR

**Abstract.** This paper proposes a quasi-dense reconstruction from uncalibrated sequence. The main innovation is that all geometry is computed based on re-sampled quasi-dense correspondences rather than the standard sparse points of interest. It not only produces more accurate and robust reconstruction due to highly redundant and well spread input data, but also fills the gap of insufficiency of sparse reconstruction for visualization application. The computational engine is the quasi-dense 2-view and the quasi-dense 3-view algorithms developed in this paper. Experiments on real sequences demonstrate the superior performance of quasi-dense w.r.t. sparse reconstruction both in accuracy and robustness.

## 1 Introduction

3D reconstruction from uncalibrated sequences has been very active and successful in the past decade in computer vision. This is mainly due to the intrinsic formulation of geometric constraints in projective geometry and a better understanding of numerical and statistical properties of geometric estimation [19,42]. Many reconstruction algorithms based on point features have been published for short [5,13,16,6] or long sequences [38,36]. Almost all of these approaches have been based on sparse points of interests. More recent and complete systems based on these ideas are reported in [28,9,31,1,21] without any prior camera calibration or position information. Unfortunately, most modeling and visualization applications need dense or quasi-dense reconstructions rather than a sparse point clouds. Traditional dense stereo methods are limited to specific pre-calibrated camera geometries and closely spaced viewpoints [37,29,18,17]. Traditional dense stereo/motion analysis is not yet efficient and robust enough to be integrated into an on-line dense reconstruction to handle images captured by hand-held cameras. It should be noted that although the final results reported in [30] showed densely textured models, the method only applied the dense stereo reconstruction using an area-based algorithm after obtaining the geometry by a sparse method.

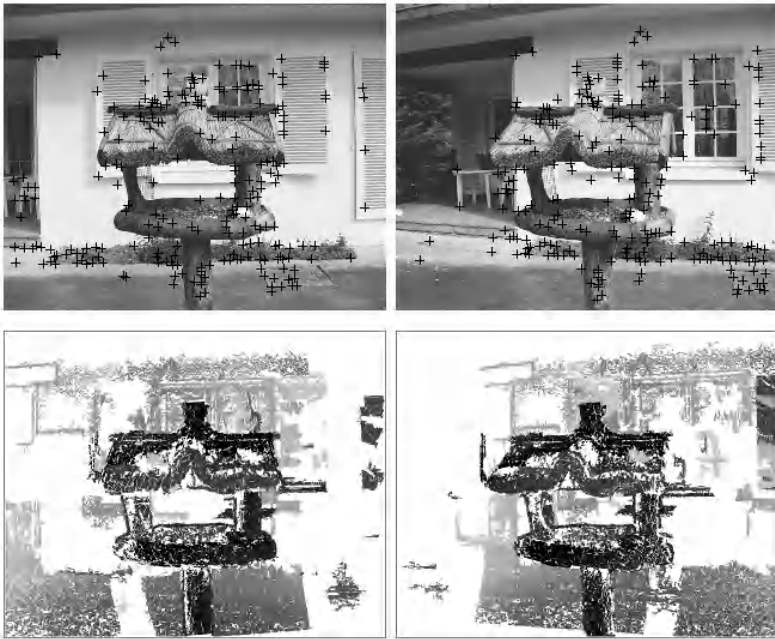
We propose to develop an intermediate approach to fill the gap between sparse and dense reconstruction methods for hand-held cameras. By quasi-dense reconstruction we mean that the geometry is directly computed on re-sampled points from quasi-dense pixel correspondences, rather than reconstructions of sparse points of interest. Quasi-dense correspondences are preferable to fully

dense ones owing to their greater robustness and efficiency with hand-held captured images. The most innovative part is that all geometry is computed with a simple and efficient quasi-dense correspondence algorithm proposed in [22,23,24] for image-based rendering applications. Quasi-dense correspondence has been integrated at the earliest stage from the building blocs of 2-view geometry and 3-view geometry up to the final sub-sequence merging. This not only gives object/scene reconstructions more suitable for visualization application, but also results in more accurate and robust estimation of camera and structure.

## 2 Review of Quasi-Dense Matching

The construction of quasi-dense matching map starts from matching some points of interest that have the highest “textureness” as seed points. This bootstraps a region growing algorithm to propagate the matches in its neighborhood from the most textured (therefore most reliable) pixels to less textured ones [22,24].

The algorithm can therefore be described in two steps: *Seed selection* and *Propagation*, which are illustrated in Figure 1.



**Fig. 1.** Top: initial seed matches for two consecutive images of the Garden-cage sequence with big disparities (some seeds are bad mainly due to the shutter periodic textures). Bottom: the resulting propagation without the epipolar constraint.

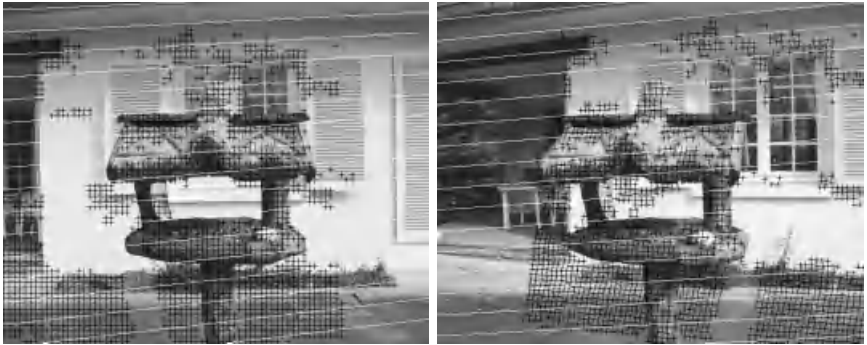
Points of interest [25,12] are naturally good seed point candidates, as points of interest are by its very definition image points which have the highest tex-



tureness, i.e. the local maxima of the auto-correlation function of the signal. We first extract points of interest from two original images, then a ZNCC correlation method is used to match the points of interest across two images, followed by a cross validation for the pair of images. This gives the initial list of seed correspondences sorted by the correlation score.

At each step of the propagation, the match  $(\mathbf{x}, \mathbf{x}')$  composed of two corresponding pixels  $\mathbf{x}, \mathbf{x}'$  with the best ZNCC score is removed from the current list of seed matches. The ZNCC is still used as it is more conservative than others such as sum of absolute or square differences in uniform regions, and is more tolerant in textured areas where noise might be important. Then we look for new potential matches  $(\mathbf{u}, \mathbf{u}')$  in their immediate spatial neighborhood  $\mathcal{N}(\mathbf{x}, \mathbf{x}')$ . This neighborhood enforces a disparity gradient limit of 1 pixel in both image dimensions  $\|(\mathbf{u}' - \mathbf{u}) - (\mathbf{x}' - \mathbf{x})\|_\infty \leq 1$  to deal with inaccurate or non available epipolar constraint. The matching uniqueness and the ending of the process are guaranteed by choosing only new matches  $(\mathbf{u}, \mathbf{u}')$  that have not yet been selected.

The time complexity of this propagation algorithm is  $O(n \log(n))$ , only dependent of the number of final matches  $n$ , and the space complexity is linear in the image size. Both complexities are independent of disparity bound. Notice that at each time only the best match is selected, this drastically limits the possibility of bad matches. For instance, the seed selection step seems very similar to many existing methods [43,39] for matching points of interest using correlation, but the crucial difference is that we need only to take the most reliable ones rather than trying to match a maximum of them. In some extreme cases, only one good seed match is sufficient to provoke an avalanche of the whole textured images. This makes our algorithm much less vulnerable to bad seeds. The same is true for propagation, the risk of bad propagation is considerably diminished by the best first strategy over all matched boundary points.



**Fig. 2.** The re-sampled matches from the propagation (described in Section 4) is represented as a set of matched black crosses in both images. These well spread matches are used to fit the fundamental matrix, shown as white epipolar lines.

### 3 Re-sampling

The matching map obtained from the propagation may still be corrupted and irregular. We assume that the scene surface is locally smooth enough to be approximated by small planar patches. Thus, the matching map can be regularized by locally fitting planar patches encoded by homographies. The first image is initially subdivided into small regular grid. For each square patch, we obtain all matched points of the square from the quasi-dense matching map. A plane homography should be tentatively fitted to these matched points of the square to look for potential planar patches. The RANdom SAMple Consensus (RANSAC) method [7] is used for robust estimation. In practice, the stability of the homography fitting decreases with the patch size. Our compromise between patch grid resolution and stability fitting is to fit a planar affine application (which counts only 6 d.o.f instead of 8 d.o.f of homography) in  $8 \times 8$ -pixel squares.

The result is a list of matches shown by crosses in Figure 2, which is better spread in image space than the usual list of matched interest points shown at the top of Figure 1.

### 4 Estimating 2-View Geometry

The 2-view geometry of a rigid scene is entirely encoded by the fundamental matrix. The actual standard approach is to compute automatically fundamental matrix and correspondences from sparse points of interest [43,39] within a random sampling framework. There are also attempts of integrating the dense correspondence into the non-linear optimization of the fundamental matrix starting from an initial sparse solution by optimizing a global correlation score [11], but the algorithm is very slow in computation time (7-12 minutes vs. 20-40 seconds for the method to be proposed here for images of size  $512 \times 512$  and similar processors) and fragile to handle occlusion for widely separated images as those in the image pair shown in Figure 2. In the context of our quasi-dense matching algorithm, we have two choices of integrating the geometry estimation into the match propagation algorithm for quasi-dense matching. The first is an epipolar constrained propagation which grows only those satisfying the epipolar constraint, while the second is an unconstrained one. The advantage of constrained propagation is that the bad propagation might be stopped earlier, but the domain of propagation might be reduced. Even more seriously, the geometry estimated with a robust method often tends to be locally fitted to a subset of images. We therefore prefer a strategy of an unconstrained propagation followed by a more robust constrained propagation as follows:

1. Detect points of interest in two images and compute the first correspondences by correlation and bidirectional consistency [10].
2. Run an unconstrained propagation.
3. Re-sample the obtained quasi-dense correspondences using a regular sampling grid in one image, and deduce the corresponding re-sampled points in the other image using the estimated local homographies.

4. Match the detected points of interest using the estimated local homographies, and add these matches in the list of re-sampled quasi-dense correspondences.
5. Estimate the fundamental matrix  $\mathbf{F}$  using a standard robust algorithm [39, 43,15] on re-sampled quasi-dense correspondences.
6. Run an epipolar constrained propagation by  $\mathbf{F}$ .
7. Again re-sample the quasi-dense matches from the constrained propagation and again add matched points of interest.
8. Re-estimate the fundamental matrix  $\mathbf{F}$  using the re-sampled quasi-dense correspondences.

## 5 Estimating 3-View Geometry

The 3-view geometry plays a central role for construction of longer sequences as 3 views is the maximum number of images which can be solved in closed-form, but also it is the minimum number of images which has sufficient geometric constraints to remove match ambiguity. The projective reconstruction from a minimum of 6 points in 3 views is therefore the basic computational engine for 3-view geometry both for robust assessment of correspondences using RANSAC and for optimal bundle adjustment of the final solution [33,34,40,15,35].

The quasi-dense 3-view algorithm can be summarized as follows.

1. Apply the previous quasi-dense 2-view algorithm to the pair  $i$  and  $i - 1$  and the pair  $i$  and  $i + 1$ .
2. Merge the two re-sampled quasi-dense correspondences between the pair  $i - 1$  and  $i$  and the pair  $i$  and  $i + 1$  via the common  $i$ th frame as the set intersection to obtain an initial re-sampled quasi-dense correspondences of the image triplet.
3. Randomly draw 6 points to run RANSAC to remove match outliers using re-projection errors of points. For 6 randomly selected points, compute the canonical projective structure of these points and the camera matrices using the closed-form 6-point algorithm [33].

The other image points are reconstructed using the current camera matrices and re-projected back onto images to evaluate their consistency with the actual estimate.

4. Bundle adjust 3-view geometry with all inliers of triplet correspondences by minimizing the re-projection errors of all image points by fixing one of the initial camera matrices.

The general philosophy of exploiting strong 3-view geometry for long sequence reconstruction is the same as the previous methods [9,21,40], but it differs from [9] in the following aspects:

- We do not transfer point pairs for guided matching. The 3-view geometry only assesses inliers and outliers from the common re-sampled quasi-dense correspondences, it is therefore fast as the percentage of outliers is small.

- We do not use trifocal tensor parametrization of 3-view geometry as suggested in [9,1,15,35]. We use the P-matrix representation directly from the canonical projective structure of 6 points to reconstruct other points and evaluate their re-projection errors for assessing inliers/outliers of correspondences by RANSAC.  
Tensor parametrization is hardly justified here as it gives a rather complicated over-parametrization of the 3-view geometry, more sophisticated numerical algorithms are necessary for its estimation. The transfer error tends to accept points which are large outliers to the re-projection error from the optimal estimate [9,8]. Tensor might be useful for guided matching [8], but is unnecessary in our case.
- We use the closed-form 6-point algorithm [33] rather than more recent methods proposed in [40,15,34] as the initial solution for robust search and optimization. It is direct and fast without any SVD computation compared with the algorithm [15] that we have also implemented and tested. The improvement provided by Schaffalitzky et al. [34] is necessary only when redundant data has to be handled.

## 6 Merging Pairs and Triplets into Sequences

From pairs and triplets to sequences, we essentially adapt the hierarchical merging strategy successfully used in [9,21] which is more efficient than an incremental merging strategy.

The general hierarchical N-view algorithm can be summarized as:

1. For each pair of consecutive images in the sequence, apply the quasi-dense 2-view algorithm described in Section 4.
2. For each triplet of consecutive images in the sequence, apply the quasi-dense 3-view algorithm described in Section 5.
3. Apply a hierarchical merging algorithm of sub-sequences. A longer sequence  $[i..j]$  is obtained by merging two shorter sequences  $[i..k+1]$  and  $[k..j]$  with two overlapping frames  $k$  and  $k+1$ , where  $k$  is the median of the index range  $[i..j]$ . The merge consists of
  - a) Merging the two re-sampled quasi-dense correspondences between two sub-sequences using the 2 overlapping images.
  - b) Estimating the space homography between two common cameras using linear least squares.
  - c) Apply the space homography for all camera matrices and all points not common in the two sub-sequences.
  - d) Bundle adjust the sequence  $[i..j]$  with all merged corresponding points.

In [9], several algorithms have been proposed to merge two triplets with 0, 1 or 2 overlapping views. The main advantage of imposing two-view overlapping is that camera matrices are sufficient for estimating the space homography to merge two reconstructions without any additional point correspondences between the two. It is also important to notice that both re-sampled quasi-dense points from 3-view geometry and sparse points of interest are contributing to the merging and optimization steps.

## 7 Optimal Euclidian Estimation of Reconstruction

The final step is to upgrade the projective reconstruction into a metric representation using self-calibration and optimal estimates of the metric representation.

- A linear solution [30] based on the parametrization of the dual of the absolute conic [41] is used for estimating constant but unknown focal lengths while assuming the other intrinsic camera parameters, such as principal point and aspect ratio, are given. If the algorithm fails, we simply perform a one-dimensional exhaustive search of the focal lengths from a table of possible values.
- Transform the projective reconstruction by the estimated camera parameters to its metric representation. The metric reconstruction coordinate system is those of the camera in the middle of the entire sequence and the scale unit is the maximum distance between any pairs of camera positions.
- Re-parametrize each Euclidian camera by its 6 individual extrinsic parameters and one common intrinsic focal length. This natural parametrization allows us to treat all cameras equally when estimating uncertainties, but leaves the 7 d.o.f scaled Euclidian transformation gauge freedom [42,27,3]. Finally, apply an Euclidian bundle adjustment over all cameras and all quasi-dense points.
- A second Euclidian bundle adjustment by adding one radial distortion parameter for all cameras is carried out in the case where the non-linear distortions of cameras are non-negligible, for instance, for image sequences captured by a very short focal length.

It is obvious that the sparse structure of the underlying numerical system as suggested in photogrammetry [2,26] and vision [14,42,15] has to be exploited for the implementations of both projective and Euclidian bundle adjustments as we are routinely handling at least 10 thousand 3D points. It is also natural to use reduced camera subsystem by eliminating the structure parameters.

## 8 Comparative Experiments

This section demonstrates the accuracy and robustness of the quasi-dense reconstruction method (QUASI) by comparing it with the standard sparse methods (SPARSE). We will use two sparse reconstruction algorithms based only on points of interest. The first consists of simply tracking all points of interest detected in each individual image. The second is a mixture of sparse and quasi-dense: it consists of assessing points of interest from individual images by geometry that is computed from quasi-dense algorithm, and to re-evaluate the complete geometry only from these matched interest points. In the following, it is meant by “SPARSE” the best result of these two methods.

To measure the reconstruction accuracy, we may consider the bundle adjustment as the maximum likelihood estimates of both camera and scene structure geometry, if we admit that the image points are normally distributed around

their true locations with an unknown standard deviation  $\sigma$ . This assumption is reasonable both from theoretical and practical point of view [20]. The confidence regions for a given probability can therefore be computed from the covariance matrix of the estimated parameters.

The covariance matrix is only defined up to the choice of gauge [42,27,3] and the common unknown noise level  $\sigma^2$ . The noise level  $\sigma^2$  is estimated from the residual error as  $\sigma^2 = r^2/(2e - d)$  where  $r^2$  is the sum of the  $e$  squared reprojection errors,  $d$  is the number of independent parameters of the minimization  $d = 1 + 6c + 3p - 7$  (1 counts for the common focal length,  $c$  is the number of cameras,  $p$  is the number of reconstructed points and 7 is the gauge freedom choice). All results given here are gauge free: the covariance is computed without imposing gauge constraints, now in the coordinate system of the camera in the middle of the sequence and with the scale unit equal to the maximum distance between camera centers. We obtain the same conclusions for the comparisons between SPARSE and QUASI with a camera-centered gauge by fixing orientation and position of the middle camera (especially, the uncertainty  $\sigma_f$  is the same for all gauge choices since  $f$  is gauge invariant). Since the full covariance matrix is very big, only its diagonal blocs for cameras and points are computed using sparse pseudo-inversion method [3,15].



**Fig. 3.** A synthetic scene composed of two spline surfaces and a very distant plane with three textures mapped on it: random textured scene (left), indoor textured scene (middle) and outdoor textured scene (right).

We choose a 90% confidence ellipsoid for any 3D position vector: if  $\mathbf{C}$  is a  $3 \times 3$  covariance sub-matrix of any camera position or point extracted from the full covariance matrix of all parameters, the confidence ellipsoid is therefore defined by  $\Delta \mathbf{x}^T \mathbf{C}^{-1} \Delta \mathbf{x} \leq 6.25$ , i.e. a 90% probability for a chi-square distribution with 3 degrees of freedom [32]. The maximum of semi-axes of 90% confidence ellipsoid is computed as the uncertainty bound for each 3D position. As the number of cameras is moderate, we only use the mean of all uncertainty bounds of camera positions  $\bar{\mathbf{x}}_{c_i}$  to characterize the camera uncertainty. The number of points is however quite consequent, particularly for the QUASI method. To have a better characterization of their uncertainties, we compute the rank 0 (the smallest uncertainty bound  $\mathbf{x}_0$ ), rank  $\frac{1}{4}$  ( $\mathbf{x}_{\frac{1}{4}}$ ), rank  $\frac{1}{2}$  (median  $\mathbf{x}_{\frac{1}{2}}$ ), rank  $\frac{3}{4}$  ( $\mathbf{x}_{\frac{3}{4}}$ ) and rank 1 (the largest uncertainty bound  $\mathbf{x}_1$ ) of the sorted uncertainty bounds to assess the uncertainty of the reconstructed points. The uncertainty of the focal length  $f$  is given by the standard deviation  $\sigma_f$ .

## 8.1 Synthetic Examples

First we experiment on a synthetic scene with two spline surfaces and a very distant plane with three textures and uneven spread of points of interest: a well textured random scene, a low textured indoor scene and an outdoor scene of low texture, as illustrated in Figure 3. The camera moves around a vertical axis at the middle of the scene by 5 degrees 5 times, and the image size is  $256 \times 256$ .

**Table 1.** Uncertainty measures for the synthetic scene. The right column are the accuracy of camera centers w.r.t ground truth

Random	#3D points	$\sigma$	$f$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$	$\ \mathbf{x}_{c_i}\ $
QUASI	2559	.25	256	8.8e-4	.011	.031	.056	.96	1.6	9.5e-4
SPARSE	126	.45	256	6.0e-3	.065	.11	.15	.19	4.7	3.1e-3
Indoor	#3D points	$\sigma$	$f$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$	$\ \mathbf{x}_{c_i}\ $
QUASI	1459	.36	256	1.8e-3	.022	.046	.076	0.14	5.0	1.6e-3
SPARSE	114	.42	256	4.8e-3	.055	0.61	0.69	0.11	1.6	3.8e-3
Outdoor	#3D points	$\sigma$	$f$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$	$\ \mathbf{x}_{c_i}\ $
QUASI	1547	.34	256	1.6e-3	.019	.041	.071	0.11	2.9	2.0e-3
SPARSE	66	.46	256	7.3e-3	.070	0.93	0.15	0.21	3.0	6.8e-3

The computed uncertainty measures are shown in Table 1. With 10 to 20 times more points, the QUASI uncertainties are usually 2 to 5 times smaller. As expected, the points on the textured and distant plane are very uncertain in comparison with the others. In this particular case of synthetic scene, all intrinsic parameters (including the known focal length  $f$ ) are enforced by the final Euclidian bundle adjustment. Furthermore, the true camera motion is known: we compute the accuracy of the movement  $\|\mathbf{x}_{c_i}\|$  as the mean of the Euclidian distance between the estimated and the true centers of cameras. The QUASI accuracy is usually 3 times better than the SPARSE one. These conclusions are the same if the focal length  $f$  is estimated in the final Euclidian bundle, with a better  $f$  accuracy for QUASI.

We have also experimented with other synthetic examples where the set of matched interest points is well spread in image space, and have found in these last cases that the accuracies are usually better for SPARSE than for QUASI, although uncertainties are usually better for QUASI than for SPARSE.

## 8.2 Real Examples

We also give detailed experimental results on three real sequences. The Corridor sequence (11 images at resolution  $512 \times 512$ ) has a forward motion along the scene which does not provide strong geometry, but favors the SPARSE method as it is a low textured polyhedric scene, points of interest are abundant and well spread over the scene. The Lady sequence (20 images at  $768 \times 512$ ) has a more favorable lateral motion in close-range. The Garden-cage sequence (34 images



**Fig. 4.** From left to right: Corridor (11 images at  $512 \times 512$  resolution), Lady (20 images at  $768 \times 512$ ), Garden-cage (34 images at  $640 \times 512$ ) sequences.

at  $640 \times 512$ ) are captured by a hand-held still camera (Olympus C2500L) with an irregular but complete inward walk around the object.

**Table 2.** Uncertainty measures for the Corridor sequence: the mean of the uncertainty bounds of camera centers and the rank- $k$  of the sorted uncertainty bounds of points.

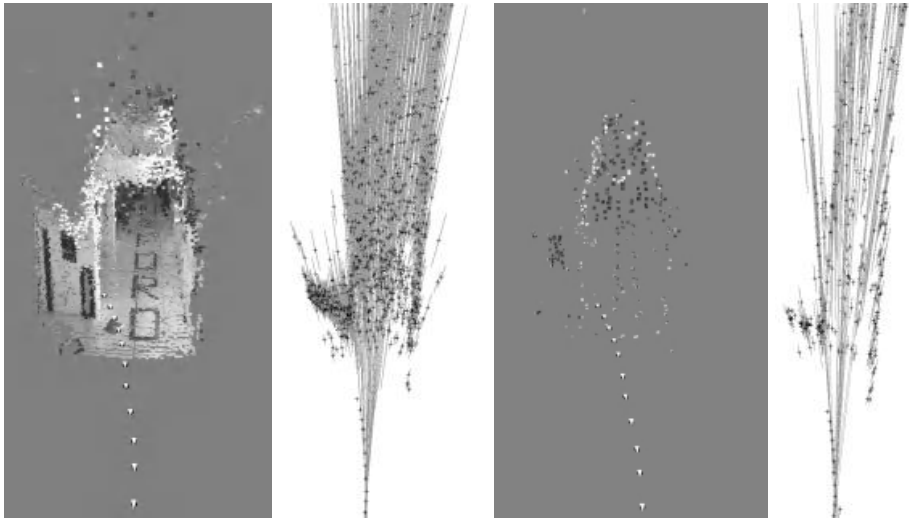
Corridor	#3D points	$\sigma$	$f$	$\sigma_f$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$
QUASI	16976	0.41	714	4.36	7.0e-4	.014	.070	.13	.38	15700
SPARSE	427	0.52	761	17.3	1.7e-3	.016	.056	.12	.32	106

**Corridor.** Table 2 shows the comparative uncertainty measures for the Corridor sequence. With almost 40 times redundancy, camera position (resp. focal length) uncertainties from QUASI are two times (resp. four times) smaller than those from SPARSE. However, the point uncertainties for SPARSE are slightly better than those of QUASI for the majority of points. As the camera direction and path is almost aligned with the scene points, the points on the far background of the corridor are almost at infinity. Not surprisingly with the actual fixing rules of the coordinate choice, they have extremely high uncertainty bound along the camera direction for both methods. Figure 5 shows the reconstruction results in which each 3D points is displayed as a small texture square around it, and illustrates a plane view of the 90% confidence ellipsoids.

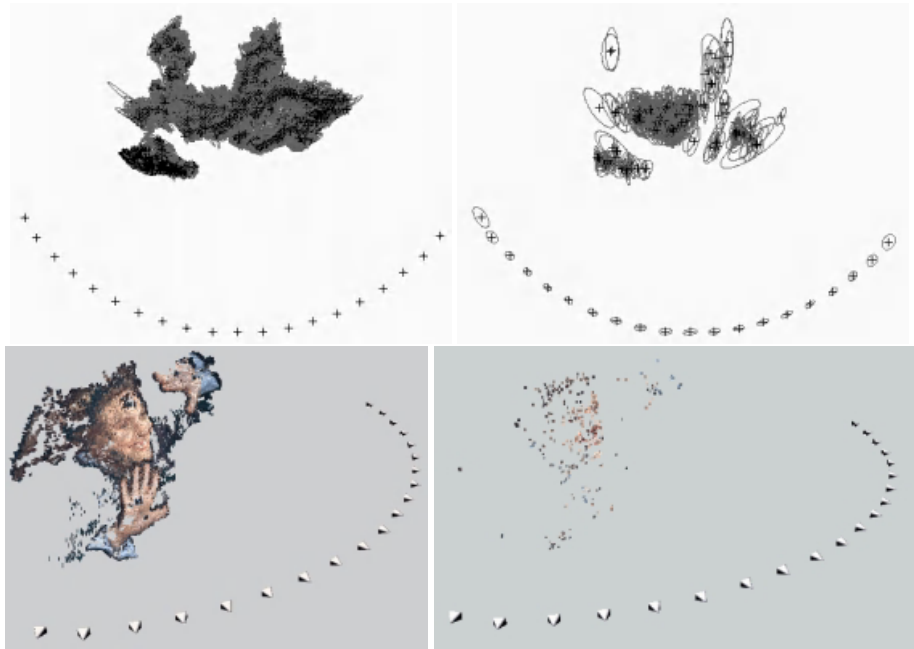
**Lady.** For the Lady sequence, we show the results obtained from SPARSE and QUASI methods in Table 3 and Figure 6. The uncertainties for QUASI are smaller than for SPARSE, 6 times smaller for focal length and camera positions. We have noticed that the very small number of 3D points makes the SPARSE method fragile.

**Garden-cage.** The Garden-cage sequence is particularly difficult as it is consisting of a close-up bird cage and background houses and trees. The viewing field is therefore very profound. SPARSE methods failed because some triplets of consecutive images do not have sufficient matched interest points. The QUASI





**Fig. 5.** QUASI (left) and SPARSE (right) reconstructions for Corridor and their 90% confidence ellipsoids viewed on a horizontal plane. Only 1 out of 10 ellipsoids for QUASI is displayed.



**Fig. 6.** QUASI (left) and SPARSE (right) reconstruction of the Lady sequence. The 90% ellipsoids for final Euclidian bundle adjustment are enlarged 4 times on the top.

**Table 3.** Uncertainties for the Lady sequence.

Lady	#3D points	$\sigma$	$f$	$\sigma_f$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$
QUASI	26823	.53	849	2.26	6.1e-4	1.2e-3	4.2e-3	5.4e-3	6.1e-3	1.9e-2
SPARSE	383	.54	866	13.6	3.8e-3	5.2e-3	9.8e-3	1.1e-2	1.3e-2	2.6e-2

method gives the uncertainties in Table 4 and 90% ellipsoids in Figure 7. As the images have been captured with the smallest focal length available, the camera non-linear distortion is becoming not negligible. After a first round of Euclidian bundle adjustment, a second adjustment by adding one radial distortion parameter  $\rho$  for all cameras is carried out.

Let  $\mathbf{u}_0$ ,  $\mathbf{x}$  and  $\mathbf{u}$  be respectively the image center, undistorted and distorted points for an image, the first order radial distortion parameter  $\rho$  is defined as:

$$\mathbf{u} = \mathbf{u}_0 + (1 + \rho(r/2v_0)^2)(\mathbf{x} - \mathbf{u}_0),$$

where  $r = \sqrt{\|\mathbf{x} - \mathbf{u}_0\|^2}$  and  $(u_0, v_0)^\top = \mathbf{u}_0$ .

We have estimated  $\rho = -0.086$  with our method. This estimate is similar to that obtained with a very different method proposed in [4] for the same camera but different images:  $\rho = -0.084$ .

Computation times in minutes for the QUASI method are given in Table 5 for all sequences using a Pentium III 500 Mhz processor.

### 8.3 Robustness

The robustness of the methods can be measured by the success rate of reconstruction for a given sequence. The QUASI method is clearly more robust for all sequences we have tested: whenever a sequence is successful for SPARSE, it is equally for QUASI, while SPARSE fails for many other sequences (not shown in this paper) including the Garden-cage sequence in which QUASI succeeds. Furthermore in all our test, the SPARSE method was defined at the very beginning of this Section as the best result between a pure sparse and a mixed sparse-quasi methods, where the mixed one is sometimes the only one which succeed.

**Table 4.** Uncertainties for the Garden-cage sequence.

Garden-cage	#3D points	$\sigma$	$f$	$\sigma_f$	$\rho$	$\sigma_\rho$	$\bar{\mathbf{x}}_{c_i}$	$\mathbf{x}_0$	$\mathbf{x}_{\frac{1}{4}}$	$\mathbf{x}_{\frac{1}{2}}$	$\mathbf{x}_{\frac{3}{4}}$	$\mathbf{x}_1$
QUASI	50161	.46	732	0.27	-0.086	1.1e-4	3.8e-4	5.2e-4	1.9e-2	4.4e-2	0.12	2.6

## 9 Conclusion

In this paper, we have proposed a general quasi-dense 3D reconstruction from uncalibrated sequences. The main innovative idea is that all geometry is computed based on re-sampled quasi-dense correspondences rather than only standard sparse points of interest. Experiments demonstrate its superior performance



**Fig. 7.** Top view of the 90% confidence ellipsoids and 3 re-projected views of QUASI reconstruction. The small square shaped connected component at the center is the reconstructed bird cage while the visible crosses forming a circle are camera positions.

**Table 5.** Computation times (min.) for the QUASI method with a PIII 500 Mhz.

	#cameras	#3D points	matching and 2-views	3-views and merge
Corridor	11	16976	6	11
Lady	20	26823	13	16
Garden-cage	34	50161	25	27

both in accuracy and robustness due to highly redundant and well spread input data. Quasi-dense reconstruction has also more visualization related application than sparse reconstruction. Future research directions include time reduction for longer sequences by intelligent decimation of reconstructed points in the hierarchical bundle, and all rendering related topics such as meshing and texture merging for a full 3D surface models.

**Acknowledgments.** We would like to thanks A. Zisserman for the “Corridor” sequence, D. Taylor for the “Lady” sequences, Jerome Blanc for the synthetic data, and Bill Triggs for discussions.

## References

1. P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. *ECCV'96*.
2. D.C. Brown. The bundle adjustment – progress and prospects. *International Archive of Photogrammetry*, 21, 1976.
3. D.D. Morris, K. Kanatani and T. Kanade. Uncertainty modeling for optimal structure from motion. In *ICCV'99 Workshop Vision Algorithms : Theory and Practice*.
4. F. Devernay and O. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. *Proceedings of the SPIE Conference on Investigate and Trial Image Processing, San Diego, California, USA*, volume 2567.
5. O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? *ECCV'92*.
6. O. Faugeras and Q.T. Luong, *The Geometry of Multiple Images* The MIT Press, 2001.
7. 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.
8. A.W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3d model construction for turn-table sequences. *SMILE'98*.
9. A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. *ECCV'98*.
10. P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve discontinuities. *IJCAI'91*.
11. C. Gauclin and T. Papadopoulos. Fundamental matrix estimation driven by stereo-correlation. *ACCV'00*.
12. C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 1988.
13. R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. *CVPR'92*.
14. R.I. Hartley. Euclidean Reconstruction from Uncalibrated Views. *DARPA-ESPRIT Workshop on Applications of Invariants in Computer Vision*, 1993.
15. R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.
16. A. Heyden, *Geometry and Algebra of Multiple Projective Transformations*. PhD. Thesis, Lund Institute of Technology, Sweden, December 1995.
17. M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. *ICCV'95*.
18. T. Kanade, P.J. Narayanan, and P.W. Rander. Virtualized reality: Concepts and early results. *Workshop on Representation of Visual Scenes*, June 1995.

19. K. Kanatani. *Statistical Optimisation for Geometric Computation: Theory and Pra*. Elsevier Science, 1996.
20. Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features ? *ICCV'01*.
21. S. Laveau. *Géométrie d'un système de N caméras. Théorie, estimation, et applications*. PhD Thesis, École Polytechnique, France, May 1996.
22. M. Lhuillier and L. Quan. Image interpolation by joint view triangulation. *CVPR'99*.
23. M. Lhuillier and L. Quan. Robust Dense Matching Using Local and Global Geometric Constraints. *ICPR'00*.
24. M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *IEEE Trans. on PAMI*, 2002.
25. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI'81*.
26. *Manual of Photogrammetry, Fourth Edition*. American Society of Photogrammetry and Remote Sensing, 1980.
27. P. F. McLauchlan. Gauge independence in optimization algorithms for 3D vision. *Proceedings of the Vision Algorithms Workshop*, 2000.
28. D. Nister. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. *ECCV'00*.
29. Y. Ohta and T. Kanade. Stereo by intra and inter-scanline search using dynamic programming. *IEEE Trans. on PAMI*, 7(2):139–154, 1985.
30. M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *ICCV'98*.
31. M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. *SMILE'98*.
32. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 1992.
33. L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. on PAMI*, 17(1):34–46, January 1995.
34. F. Schaffalitzky, A. Zisserman, R. Hartley, and P.H.S. Torr. A six point solution for structure and motion. *ECCV'00*.
35. A. Shashua. Trilinearity in visual recognition by alignment. *ECCV'94*.
36. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. *ECCV'96*.
37. H.S. Sawhney H. Tao and R. Kumar. A global matching framework for stereo computation. *ICCV'01*.
38. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, November 1992.
39. P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3):271–300, 1997.
40. P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *IVC*, 5(15):591–605, 1997.
41. B. Triggs. Autocalibration and the Absolute Quadric. *CVPR'97*.
42. B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle ajustment — a modern synthesis. *Vision Algorithms: Theory and Practice*, 2000.
43. Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI*, 78(1-2):87–119, 1994.

# Properties of the Catadioptric Fundamental Matrix

Christopher Geyer and Kostas Daniilidis\*

GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104  
{cgeyer,kostas}@seas.upenn.edu

**Abstract.** The geometry of two uncalibrated views obtained with a parabolic catadioptric device is the subject of this paper. We introduce the notion of circle space, a natural representation of line images, and the set of incidence preserving transformations on this circle space which happens to equal the Lorentz group. In this space, there is a bilinear constraint on transformed image coordinates in two parabolic catadioptric views involving what we call the catadioptric fundamental matrix. We prove that the angle between corresponding epipolar curves is preserved and that the transformed image of the absolute conic is in the kernel of that matrix, thus enabling a Euclidean reconstruction from two views. We establish the necessary and sufficient conditions for a matrix to be a catadioptric fundamental matrix.

## 1 Introduction

The geometry of perspective views has been extensively studied in the past decade. Two books [6] and [2] contain comprehensive treatments of the subject. At the same time, the need for a larger field of view in surveillance, robotics, and image based rendering motivated the design of omnidirectional cameras. Among several designs, the catadioptric systems with a single effective viewpoint, called central catadioptric [10], attracted special attention due to their elegant and useful geometric properties (see the collection [1]). Structure from motion given omnidirectional views is an evolving research area. Gluckman and Nayar [5] studied ego-motion estimation by mapping the catadioptric image to the sphere. Svoboda et al [14] first established the epipolar geometry for all central catadioptric systems. Kang [8] proposed a direct self-calibration by minimizing the epipolar constraint.

In this paper we study the geometry of two uncalibrated views obtained with a parabolic catadioptric device. We assume that the optical axes of the lens and the mirror are parallel and that the aspect ratio and skew parameter are known leaving only the focal length (combined scaling factor of mirror, lens, and CCD-chip) and the image center (intersection of the optical axis with the

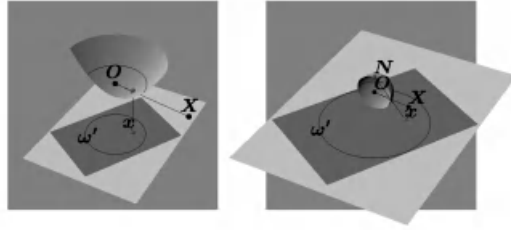
---

\* The authors are grateful for support through the following grants: NSF-IIS-0083209, SNF-EIA-0120565, NSF-IIS-0121293, NSF-EIA-9703220, a DARPA/ITO/NGI sub-contract to UNC, and a Penn Research Foundation grant.

image plane) as unknown. The parabolic projection  $\mathbf{x} = (u, v, 1)^T$  of a point  $\mathbf{X} = (x, y, z, w)^T \in \mathbb{P}^3$  incorporates two steps: 1. intersecting the paraboloid and the ray from the paraboloid's focus through  $\mathbf{X}$ ; and 2. orthographically projecting this intersection to the image plane. It reads [10,14,3] as follows

$$u = c_x + \frac{2fx}{-z + \sqrt{x^2 + y^2 + z^2}} \quad \text{and} \quad v = c_y + \frac{2fy}{-z + \sqrt{x^2 + y^2 + z^2}}, \quad (1)$$

where  $(c_x, c_y, 1)$  is the intersection of the optical axis with the image plane and  $f$  is the projected focal length of the mirror, and where it is also assumed that the focus is  $O = (0, 0, 0, 1)$ , the origin, and the  $z$ -axis is parallel to the optical axis (1 left). The circle centered at  $(c_x, c_y, 1)$  and whose imaginary radius is  $2f$  will be named  $\omega'$  and is called the calibrating conic because it gives the three intrinsics  $c_x$ ,  $c_y$  and  $f$ . Every image of a line is a circle which intersects  $\omega'$  antipodally [3].



**Fig. 1.** The projection on a paraboloidal mirror with subsequent orthographic projection (left) and the equivalent model: spherical projection with subsequent stereographic projection (right).

It was shown in [3] that the parabolic projection described above is equivalent to another two step projection: project the point in space to the sphere and then project this point from the north pole to the plane of the equator; see figure 1 (right). This type of projection is equivalent to a parabolic projection in which the calibrating conic  $\omega'$  is identical to the projection of the equator. The second step in the two step projection is stereographic projection which has two properties which will be relevant to us: 1. it projects any circle on the sphere great or small to a circle in the plane; and 2. stereographic projection is conformal in that locally it preserves angles [11].

In [4] an extra coordinate is added to the image coordinates so that a general perspective projection becomes proportional to a linear transformation of the new image coordinates. The mechanism is achieved by “lifting” a point in the image plane to the surface of a paraboloid that is not necessarily equal to the physical paraboloid being used as the mirror. Once lifted to the paraboloid, a special class of linear transformations preserves the surface of the parabola while inducing translation and scaling in the image points. An appropriate transformation exists which maps lifted image points into rays which are calibrated and are collinear with the space point and the focus. This lifting space also has the

advantage of being able to represent the images of lines (circles) in the image plane.

In this paper we combine the lifting idea, which has the effect of factoring out some portion of the non-linearity of the problem, with the use of stereographic projection. Thus instead of using the paraboloid as a lifting surface, we intend to use the sphere, where we will apply the inverse of stereographic projection to lift image points to the sphere. Though this can be seen to be the same as using the paraboloid, using the sphere has the advantage of being more symmetric and drastically simplifies our derivations.

We summarize here the original contributions of this paper:

1. A new representation of image points and line images for parabolic catadioptric images is defined using inverse stereographic projection.
2. The equivalent class of linear transformations of this space is shown to preserve angles and is equal to the Lorentz group.
3. A projection formula analogous to the perspective projection formula is derived. Using this projection formula we reformulate the multiple view matrix and the rank deficiency condition remains from the perspective case. Mixed sensor types can be included in the multiple view matrix.
4. From this catadioptric multiple view matrix the catadioptric fundamental matrix is derived. We prove that the lifted images of the absolute conic of the left (right) camera belong to the two-dimensional left (right) null-space of the catadioptric fundamental matrix. Self-calibration becomes, thus, the intersection of two null-spaces. It is possible with two parabolic views as opposed to three views required in the perspective case (even with known aspect ratio and skew).
5. Because of the stereographic projection involved in the parabolic projection, angles between epipolar circles are preserved. We prove the equivalent algebraic condition on the singular vectors of the catadioptric fundamental matrix.
6. Based on the last two facts, we derive the necessary and sufficient conditions for a given matrix to be a catadioptric fundamental matrix.

## 2 The Spherical Representation of Points and Circles: Circle Space

A unit sphere centered at the origin has the quadratic form

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (2)$$

Given a point  $\mathbf{x} = (u, v, 0, 1)$  we wish to find the point  $\tilde{\mathbf{x}}$  on the sphere which when stereographically<sup>1</sup> projected from  $\mathbf{N} = (0, 0, 1, 1)$  would give  $\mathbf{x}$ . It is easy to verify that the point

<sup>1</sup> This is **not** necessarily the same stereographic projection which was used to generate the image point from a point in space.



$$\tilde{\mathbf{x}} = (2u, 2v, u^2 + v^2 - 1, u^2 + v^2 + 1)^T \quad (3)$$

lies on the sphere and is collinear with  $\mathbf{N}$  and  $\mathbf{x}$ . The point  $\tilde{\mathbf{p}}$  will be called the “lifting” of the point  $\mathbf{x}$ , whereas  $\mathbf{x}$  is the stereographic projection of  $\tilde{\mathbf{x}}$ .

Circles can also be represented in this framework due to the following fact. Stereographic projection maps points on the sphere to co-circular points in the plane if and only if the points on the sphere also lie on a plane. We represent a circle in the image plane with the polar point of the plane containing the lifted image points lying on the circle. Recall from projective geometry that the polar point of a plane is the vertex of the cone tangent to the sphere (or any quadric surface) at the intersection of the plane with the sphere. The polar plane of a point has the reverse relationship.

Let  $\gamma$  be a circle centered in the image plane at  $(c_x, c_y, 1)$  with radius  $r$ . We claim that the plane containing the lifted points of  $\gamma$  is

$$\boldsymbol{\pi} = (2c_x, 2c_y, c_x^2 + c_y^2 - r^2 - 1, -c_x^2 - c_y^2 + r^2 - 1)^T.$$

The polar point of this plane  $\boldsymbol{\pi}$  will be the point representation  $\tilde{\gamma}$  (Fig. 2 (left)) of the circle  $\gamma$ , where

$$\tilde{\gamma} = \mathbf{Q}\boldsymbol{\pi} = (2c_x, 2c_y, c_x^2 + c_y^2 - r^2 - 1, c_x^2 + c_y^2 - r^2 + 1)^T. \quad (4)$$

As a result it can be shown that  $\mathbf{p} \in \gamma$  if and only if  $\tilde{\mathbf{p}}^T \mathbf{Q}\tilde{\gamma} = 0$ . This has dual interpretations: 1. the set of points  $\mathbf{p}$  lying on  $\gamma$  have liftings lying on the plane  $\mathbf{Q}\tilde{\gamma}$ ; and 2. the set of circles  $\gamma$  containing a point  $\mathbf{p}$  have point representations lying on the plane  $\mathbf{Q}\tilde{\mathbf{p}}$ . We claim that definition (4) also applies when  $r$  is imaginary.

The value of  $\rho = \mathbf{x}^T \mathbf{Q}\mathbf{x}$  determines whether  $\mathbf{x}$  lies inside ( $\rho < 0$ ), outside ( $\rho > 0$ ), or on the surface of the sphere ( $\rho = 0$ ). We find that under the condition that  $\tilde{\gamma}$  have not been scaled from their definition in (4)<sup>2</sup> then  $\rho = \tilde{\gamma}^T \mathbf{Q}\tilde{\gamma} = 4r^2$ , implying that if  $\tilde{\gamma}$  lies inside the sphere then it represents a circle with imaginary radius since  $\rho$  must be negative; if  $\tilde{\gamma}$  lies on the sphere then  $\rho = 0$  which implies that  $\gamma$  is a circle of zero radius or a point, which we already knew since it is then of the form (3); otherwise  $\tilde{\gamma}$  lies outside the sphere and represents a circle with real radius. Hence  $\tilde{\omega}$ , representing an imaginary circle, must lie inside the sphere and  $\tilde{\omega}$  must lie outside the sphere because it represents a real circle.

In particular from the definition in (4) we determine that

$$\tilde{\omega}' = (2c_x, 2c_y, c_x^2 + c_y^2 - 4f^2 - 1, c_x^2 + c_y^2 - 4f^2 + 1)^T \quad (5)$$

is the point representation of the calibrating conic. Similarly  $\omega$ , the image of the absolute conic [13], has point representation

$$\tilde{\omega} = (2c_x, 2c_y, c_x^2 + c_y^2 + 4f^2 - 1, c_x^2 + c_y^2 + 4f^2 + 1)^T. \quad (6)$$

<sup>2</sup> The circle space representation lies in  $\mathbb{P}^3$  and so is a homogeneous space, but in some rare instances like this one we will require that  $\tilde{\gamma}$  is exactly of the form in (4).

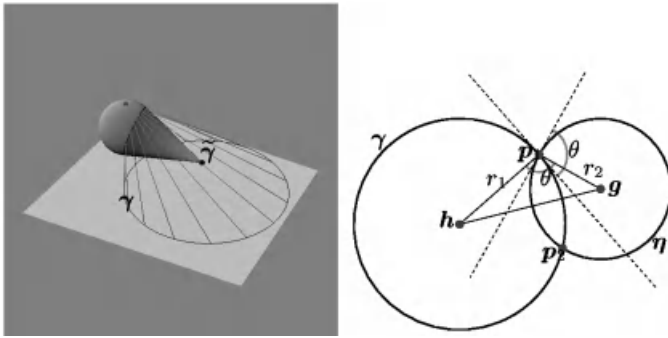
Their geometric interpretation will be elucidated in Proposition 2.

We now state without proof some miscellaneous facts. We define  $\pi_x = Qx$  to be the polar plane of the point  $x$  with respect to  $Q$ . The first fact is that a circle on the sphere projects to a line if and only if the circle contains  $N$ . All points on  $\pi_N$ , in this case the plane tangent to the sphere at  $N$ , are points whose polar planes must contain  $N$ . Therefore  $\pi_N$  contains the point representations of lines in the image plane.

The second fact is that points on  $\pi_\infty$  have polar planes going through the origin and therefore yield great circles. Thus the points at infinity represent exactly the lines of  $\mathbb{P}^2$  as they are represented on the sphere. Is there a linear transformation of circle space which maps the point representations of line images to  $\pi_\infty$  so that they represent line images in  $\mathbb{P}^2$ ?

The third fact is a cautionary note. Unlike in perspective geometry where the line image between two image points is uniquely defined, this is not the case in circle space. For any two image points there is a one parameter family of circles, a line in circle space, going through them. The correct circle for a given parabolic catadioptric image is the one which intersects  $\tilde{\omega}'$  antipodally.

Since we will be dealing with the angle of intersection of two circles we need a well-defined way to determine this angle. If two circles  $\gamma$  and  $\eta$  are centered respectively at  $g$  and  $h$ , have radii  $r_1$  and  $r_2$ , and intersect at  $p_1$  and  $p_2$ , we define the angle between them to be the angle  $\angle gp_1h$ . This angle is the same as  $\pi$  minus the angle between the tangent vectors as can be seen from Figure 2 (right). Let  $\langle x, y \rangle_Q = x^T Q y$  and  $\|x\|_Q = \sqrt{\langle x, x \rangle_Q}$ <sup>3</sup>.



**Fig. 2.** On the left, the lifting of a circle  $\gamma$  to the point  $\tilde{\gamma}$  in spherical circle space. On the right, the angle of intersection of two circles  $\gamma$  and  $\eta$  is defined to be the angle  $\angle gp_1h$  since this is the same as at least one of the two angles between the tangent vectors.

**Proposition 1.** The angle  $\theta$  between two circles  $\gamma$  and  $\eta$  can be obtained from the “dot product” in circle space:

<sup>3</sup>  $\langle \cdot, \cdot \rangle_Q$  is *not* a real dot product nor is  $\|\cdot\|_Q$  a real norm.  $\langle \cdot, \cdot \rangle_Q$  is a symmetric bilinear form but since  $Q$  is not positive definite it does not officially qualify as a dot product.

$$\cos^2 \theta = \frac{\langle \tilde{\gamma}, \tilde{\eta} \rangle_Q^2}{\|\tilde{\gamma}\|_Q^2 \|\tilde{\eta}\|_Q^2}. \quad (7)$$

**Proof:** As shown in Fig. 2 let  $\mathbf{g}$  be the center and  $r_1$  the radius of the circle  $\gamma$ , and let  $\mathbf{h}$  be the center and  $r_2$  the radius of  $\eta$ . Let  $\mathbf{p}_1$  be one of the intersections of the two circles. By solving for  $\cos \theta$  in the law of cosines the angle  $\theta = \angle \mathbf{g}\mathbf{p}_1\mathbf{h}$  satisfies

$$\cos^2 \theta = \frac{(r_1^2 + r_2^2 - \|\mathbf{g} - \mathbf{h}\|^2)^2}{4r_1^2 r_2^2} = \frac{(r_1^2 + r_2^2 - \mathbf{g}^T \mathbf{g} - \mathbf{h}^T \mathbf{h} + 2\mathbf{g}^T \mathbf{h})^2}{4r_1^2 r_2^2}.$$

According to the assumptions we must have

$$\begin{aligned} \tilde{\gamma} &= \lambda (2\mathbf{g}^T, \mathbf{g}^T \mathbf{g} - r_1^2 - 1, \mathbf{g}^T \mathbf{g} - r_1^2 + 1)^T \\ \tilde{\eta} &= \mu (2\mathbf{h}^T, \mathbf{h}^T \mathbf{h} - r_2^2 - 1, \mathbf{h}^T \mathbf{h} - r_2^2 + 1)^T. \end{aligned} \quad (8)$$

First notice that by calculating  $\tilde{\gamma}^T \mathbf{Q} \tilde{\eta}$  one finds that

$$\mathbf{g}^T \mathbf{h} = \mathbf{g}^T \mathbf{g} + \mathbf{h}^T \mathbf{h} - r_1^2 - r_2^2 + \frac{\langle \tilde{\gamma}, \tilde{\eta} \rangle_Q}{2\lambda\mu}, \quad (9)$$

and also that  $r_1^2 = \|\tilde{\gamma}\|_Q^2/4\lambda^2$  and  $r_2^2 = \|\tilde{\eta}\|_Q^2/4\mu^2$ . Substituting (9) into (8) and then substitutions for  $r_1^2$  and  $r_2^2$  yields (7).  $\square$

The square in  $\cos^2 \theta$  is necessary because  $\tilde{\gamma}$  and  $\tilde{\eta}$  are homogeneous and the scale factors  $\lambda$  and  $\mu$  could be negative. The corollary follows immediately from the proposition.

**Corollary 1.** Two circles  $\gamma$  and  $\eta$  are orthogonal if and only if  $\tilde{\gamma}^T \mathbf{Q} \tilde{\eta} = 0$ .

**Lemma 1.** Two circles  $\gamma$  and  $\eta$  are centered at the same point and have a ratio of radii equal to  $i$  (one is imaginary, the other is real, but excluding complex circles) if and only if they are orthogonal and their polar planes intersect in a line on  $\pi_N$ .

**Proof:** The forward and reverse directions can be verified by direct calculation. Verify that the first conditions imply  $\tilde{\gamma} \mathbf{Q} \tilde{\eta} = 0$  and that the three planes are linearly dependent (the  $3 \times 3$  sub-determinants of the matrix  $(\mathbf{Q}\tilde{\gamma}, \mathbf{Q}\tilde{\eta}, \pi_N)^T$  are zero). The converse can be shown by solving for the center and radius of  $\eta$  in terms of  $\gamma$ .  $\square$

**Lemma 2.** A set of circles  $\{\gamma_\lambda\}_{\lambda \in A}$  are coaxial if and only if their point representations  $\{\tilde{\gamma}_\lambda\}_{\lambda \in A}$  are collinear.

See [12] for a proof when  $\mathbf{Q}$  is the parabola instead of the sphere. The same reasoning applies.

**Proposition 2.** Let  $\omega'$  be a circle representing the calibrating conic. The set of circles intersecting  $\omega'$  antipodally, i.e. the set of line images, lie on a plane whose polar point with respect to  $\mathbf{Q}$  is  $\omega$ .

**Proof:** All lines through the center of  $\omega'$  intersect  $\omega'$  antipodally and are also orthogonal to  $\omega'$ , therefore these lines' point representations lie on the line  $\ell$

which is the intersection of the plane  $\pi_N$  (containing all point representations of lines) and the plane  $\pi_{\tilde{\omega}'} = Q\tilde{\omega}'$  (containing all point representations of circles or lines orthogonal to  $\omega'$ ). Any circle  $\gamma$  intersecting  $\omega'$  antipodally in points  $p_1$  and  $p_2$  is coaxial with  $\omega$  and the line through  $p_1$  and  $p_2$ , which also goes through the center of  $\omega'$ . Thus by Lemma 2 their point representations are collinear. Hence  $\tilde{\gamma}$ , the representation of an arbitrary circle antipodal to  $\tilde{\omega}'$ , lies on the plane  $\pi$  through  $\ell$  and  $\tilde{\omega}'$ .

Now we show that the polar point of  $\pi$  must equal  $\tilde{\omega}$ . Let  $A = Q^{-1}\pi$  to be the polar point of  $\pi$ . The circle represented by  $A$  is orthogonal to  $\omega'$  since

$$A^T Q \tilde{\omega}' = \pi^T Q^{-T} Q \tilde{\omega}' = \pi^T \tilde{\omega}' = 0,$$

the last equality following by the definition of  $\pi$ . Since they are orthogonal and their polar planes intersect in the line  $\ell$  on  $\pi_N$ , by Lemma 1,  $\tilde{\omega}'$  and the circle represented by  $A$  must have the same center and have a ratio of radii equal to  $i$ . Therefore  $A = \tilde{\omega}$ .  $\square$

## 2.1 The Lorentz Group and Plane Preserving Subgroups

In a perspective image a natural class of transformations on image points is the set of collineations, projective transformations specified by non-singular  $3 \times 3$  matrices. We would like to find an equivalent structure for parabolic catadioptric images under the requirement that the transformation operate linearly on the circle space. Therefore this class must consist of some subset of  $4 \times 4$  matrices. These transformations also should not act in a way which happens to transform a point into a circle or vice versa, for this would inviolate incidence relationships in the image plane. Thus the surface of the sphere must remain invariant under any such transformation. This is the barest of conditions necessary to determine the set of transformations and we therefore investigate the set  $\mathcal{L} = \{A : A^T Q A = Q\}$ . This is a group since it is closed under multiplication and inversion and contains the identity. As it turns out this is a well known six dimensional<sup>4</sup> Lie group from the study of physics called the Lorentz group [7]. Any transformation from this group preserves angles between circles, for if  $A \in \mathcal{L}$  then  $\langle x, y \rangle_Q = \langle Ax, Ay \rangle_Q$ . Since two circles can be constructed to form any angle, these transformations must preserve all angles when they transform the image plane. Angles replace the cross ratio as the invariance under these transformations. It also implies that general projective transformations applied to image points that do not preserve angles, such as shearing or change of aspect ratio, can not be represented as a linear transformation of circle space, at least not in a way which preserves incidence relationships.

In the previous section it was said that the set of line images of a given parabolic projection have point representations lying on a plane in circle space. The plane on which they lie is polar to the point representation of the image of the absolute conic,  $\tilde{\omega}$ . What is the set of transformations preserving this plane and what meaning does this have? In order that a transformation preserve the

<sup>4</sup> The inclusion of scale yields an additional dimension, and then  $A^T Q A = \lambda Q$ .

plane it must preserve  $\tilde{\omega}$ . Therefore  $\tilde{\omega}$  must be an eigenvector of the transformation for any eigenvalue (since  $\tilde{\omega}$  is homogeneous). Let

$$\mathcal{L}_{\tilde{\omega}} = \{A : A^T Q A = Q \text{ and } A\tilde{\omega} = \lambda\tilde{\omega} \text{ for some } \lambda\}.$$

This is a group since it is also closed under multiplication and inversion.

We examine two subcases,  $\tilde{\omega} = (0, 0, 0, 1)$  and  $\tilde{\omega} = N$ . We will calculate the Lie algebra for the connected component containing the identity. If  $A(t)$  is a continuous parameterization of matrices in  $\mathcal{L}_{\tilde{\omega}}$  such that  $A(0) = I$ , then the first condition gives

$$\left. \frac{d}{dt} A(t)^T Q A(t) \right|_{t=0} = \left. \frac{d}{dt} Q \right|_{t=0} \quad \text{and} \quad A'(0)^T Q + Q A'(0) = 0.$$

The second condition is equivalent to the  $2 \times 2$  sub-determinants of the matrix  $(\tilde{\omega}, A(t)\tilde{\omega})^T$  being zero. Each of the six equations for the sub-determinants can be differentiated with respect to  $t$  and evaluated at  $t = 0$  and then one can solve for the entries  $A'(0)$ . When  $\tilde{\omega} = (0, 0, 0, 1)$ , this yields

$$A'(0) = \begin{pmatrix} 0 & a_{12} & a_{13} & 0 \\ -a_{12} & 0 & a_{23} & 0 \\ -a_{13} & -a_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

which is just the set of matrices which are skew symmetric in the first three rows and columns and zero elsewhere. Therefore  $\mathcal{L}_{(0,0,0,1)}$  is the set of rotations in  $\mathbb{P}^3$ .

If  $\tilde{\omega} = N$ , the north pole, then  $A'(0)$  must be of the form

$$A'(0) = \begin{pmatrix} 0 & -a_{12} & -a_{13} & a_{13} \\ a_{12} & 0 & -a_{23} & a_{23} \\ a_{13} & a_{23} & 0 & a_{34} \\ a_{13} & a_{23} & a_{34} & 0 \end{pmatrix}.$$

The Lie group generated by this Lie algebra preserves  $N$  and therefore it preserves the plane tangent to  $N$  on which lie the point representations of lines. They therefore send lines to lines while also by default preserving angles. Therefore this subgroup corresponds to affine transformations in the plane. An important subcase and reparameterization of  $\mathcal{L}_N$  is defined under the following substitutions,  $a_{12} = 0, a_{34} = -\log 2f, a_{13} = -c_x \frac{\log 2f}{2f-1}, a_{23} = -c_y \frac{\log 2f}{2f-1}$ . Upon exponentiation we have the following matrix dependent on  $\tilde{\omega}$ ,

$$K_{\tilde{\omega}} = \begin{pmatrix} 1 & 0 & c_x & -c_x \\ 0 & 1 & c_y & -c_y \\ -\frac{c_x}{2f} & -\frac{c_y}{2f} & \frac{1-c_x^2-c_y^2+4f^2}{2f} & \frac{1+c_x^2+c_y^2-4f^2}{2f} \\ -\frac{c_x}{2f} & -\frac{c_y}{2f} & \frac{1-c_x^2-c_y^2-4f^2}{2f} & \frac{1+c_x^2+c_y^2+4f^2}{2f} \end{pmatrix}. \quad (10)$$

This has the effect on the image points of translating by  $(-c_x, -c_y)$  and then scaling by  $\frac{1}{2f}$ . Also notice that  $K_{\tilde{\omega}}\tilde{\omega} = \lambda O$  and  $K_{\tilde{\omega}}\tilde{\omega}' = (0, 0, \lambda, 0)$ . We will use this matrix in the next section.

In general when  $\tilde{\omega}$  does not lie on the sphere, the dimension of  $\mathcal{L}_{\tilde{\omega}}$  is three because the sub-determinants give three independent constraints; this Lie group corresponds to rotations about the viewpoint. When  $\tilde{\omega}$  lies on the sphere an additional dimension arises because the number of independent constraints decreases by one; this Lie group leaves the image point corresponding to  $\tilde{\omega}$  invariant. One additional comment, since  $\exp A^T = (\exp A)^T$ , and since the Lie algebra of  $\mathcal{L}$  can be seen to contain  $\mathbf{A}$  if and only if it contains  $\mathbf{A}^T$ , then  $\mathbf{B} \in \mathcal{L}$  if and only if  $\mathbf{B}^T \in \mathcal{L}$ .

### 3 Multiple Parabolic Views

We now wish to find a parabolic projection equation more closely resembling the perspective projection formula  $\mathbf{I}\mathbf{X} = \lambda\mathbf{x}$ , where  $\mathbf{x} \in \mathbb{P}^2$  is the image of  $\mathbf{X} \in \mathbb{P}^3$ ,  $\mathbf{I}$  is the  $4 \times 3$  camera matrix, and  $\lambda$  is the projective depth depending on  $\mathbf{I}$ ,  $\mathbf{X}$  and  $\mathbf{x}$ . As it stands, because of the non-linearity of the definition in (1) it is not trivial to apply the multiview results found for perspective cameras to the parabolic catadioptric case.

First we apply  $\mathbf{K}_{\tilde{\omega}}$  to the lifting of point  $\mathbf{x}$  in (1), obtaining  $\frac{4f}{r-z}(x, y, z, 4fr)^T$ , where  $r = \sqrt{x^2 + y^2 + z^2}$ . This is a point collinear with  $\mathbf{O}$  (the origin) and  $\mathbf{X} = (x, y, z, w)$ . Hence for some  $\lambda$  and  $\mu$ ,  $\lambda\mathbf{O} + \mu\mathbf{K}_{\tilde{\omega}}\tilde{\mathbf{x}} = \mathbf{X}$ . Because one of the four equations in this vector equation are redundant we can multiply on both sides by

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

from which we find that  $\mu\mathbf{P}\mathbf{K}_{\tilde{\omega}}\tilde{\mathbf{x}} = \mathbf{P}\mathbf{X}$ . Upon performing the multiplication on the left hand side, one finds that in fact

$$\mathbf{P}\mathbf{K}_{\tilde{\omega}}\tilde{\mathbf{x}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{c_x}{2f} & -\frac{c_y}{2f} & \frac{1}{2f} \end{pmatrix} (\mathbf{P}\tilde{\mathbf{x}} - \mathbf{P}\tilde{\omega}) = \mathbf{J}_{\tilde{\omega}} (\mathbf{P}\tilde{\mathbf{x}} - \mathbf{P}\tilde{\omega}),$$

but this is satisfied only under the condition that  $\tilde{\mathbf{x}}$  and  $\tilde{\omega}$  have not been arbitrarily scaled from their respective definitions in (3) and (6).

Now assume that  $\mathbf{X}$  lies in a coordinate system translated by  $\mathbf{t}$  and rotated by  $\mathbf{R}$ . Introduce a projection matrix  $\mathbf{I} = \mathbf{J}_{\tilde{\omega}}^{-1}(\mathbf{R}, \mathbf{t})$  similar to the standard perspective projection matrix and define  $\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{x}}$ , then equation (3) becomes

$$\lambda(\tilde{\mathbf{x}} - \tilde{\omega}) = \mathbf{I}\mathbf{X}. \quad (11)$$

The vector  $\tilde{\omega}$  can not be incorporated into the projection matrix  $\mathbf{I}$  because the subtraction is dependent on the non-homogeneity of  $\tilde{\mathbf{x}}$ . It is interesting to note that the matrix  $\mathbf{J}_{\tilde{\omega}}^{-1}$  which fills the role of a calibration matrix is lower triangular as opposed to the perspective calibration matrix which is upper triangular. With equation (11) we can now reformulate the multiple view matrix.

Assume that  $n$  parabolic catadioptric cameras image the same point  $\mathbf{X} \in \mathbb{P}^3$  so that there are  $n$  equations of the form (11). This implies that each of the  $n$  matrices  $(\mathbf{I}_i, \tilde{\mathbf{x}}_i - \tilde{\omega}_i)$  is rank deficient because within each nullspace must

respectively lie the vector  $(\mathbf{X}, \lambda_i)^T$ . We can combine all of these matrices into the single matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{\Pi}_1 \check{\mathbf{x}}_1 - \check{\boldsymbol{\omega}}_1 & 0 & \cdots & 0 \\ \mathbf{\Pi}_2 & 0 & \check{\mathbf{x}}_2 - \check{\boldsymbol{\omega}}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Pi}_n & 0 & 0 & \cdots & \check{\mathbf{x}}_n - \check{\boldsymbol{\omega}}_n \end{pmatrix} \quad (12)$$

which again must be rank deficient because within its nullspace lies the vector  $(\mathbf{X}, -\lambda_1, -\lambda_2, \dots, -\lambda_n)^T$ . In the perspective formulation  $\mathbf{M}$  is known as the multiple view matrix [6]. By manipulating its columns and rows its rank deficiency has been used by [9] to show that the only independent constraints between multiple views are at most trilinear, all others are redundant. The same method can be applied to this parabolic catadioptric multiple view matrix to show that the only independent constraints among multiple parabolic catadioptric views are trilinear. In the next section we derive the bilinear constraints and find a form of the parabolic catadioptric fundamental matrix.

Notice that it is possible to mix different point features from different camera types. This only changes the form of one triplet of rows of the matrix  $\mathbf{M}$ . In each row the difference will be in the form of  $\mathbf{\Pi}_i$ , the presence or absence of an  $\check{\boldsymbol{\omega}}_i$  as well as lifting or not of  $\mathbf{x}$ . If all sensors image the same point in space, the multiple view matrix will be rank deficient regardless of the type of sensors.

### 3.1 Deriving the Catadioptric Fundamental Matrix

We now derive the constraint on two parabolic catadioptric views. For two views  $\mathbf{M}$  becomes

$$\mathbf{M} = \begin{pmatrix} \mathbf{\Pi}_1 \check{\mathbf{x}}_1 - \check{\boldsymbol{\omega}}_1 & 0 \\ \mathbf{\Pi}_2 & 0 & \check{\mathbf{x}}_2 - \check{\boldsymbol{\omega}}_2 \end{pmatrix},$$

where we assume  $\mathbf{\Pi}_1 = \mathbf{J}_{\check{\boldsymbol{\omega}}_1}^{-1}(\mathbf{I}, 0)$  and  $\mathbf{\Pi}_2 = \mathbf{J}_{\check{\boldsymbol{\omega}}_2}^{-1}(\mathbf{R}, \mathbf{t})$ . This a square matrix and its rank deficiency implies that its determinant is zero:

$$0 = \det \mathbf{M} = (\check{\mathbf{x}}_1 - \check{\boldsymbol{\omega}}_1)^T \mathbf{G} (\check{\mathbf{x}}_2 - \check{\boldsymbol{\omega}}_2) = (\check{\mathbf{x}}_1^T \quad 1) \begin{pmatrix} \mathbf{G} & -\mathbf{G}\check{\boldsymbol{\omega}}_2 \\ -\check{\boldsymbol{\omega}}_1^T \mathbf{G} & \check{\boldsymbol{\omega}}_1^T \mathbf{G} \check{\boldsymbol{\omega}}_2 \end{pmatrix} \begin{pmatrix} \check{\mathbf{x}}_2 \\ 1 \end{pmatrix}, \quad (13)$$

where we know from previous results for perspective cameras [6] that  $\mathbf{G} = \mathbf{J}_{\check{\boldsymbol{\omega}}_1}^T \mathbf{E} \mathbf{J}_{\check{\boldsymbol{\omega}}_2}$  for the essential matrix  $\mathbf{E} = [t]_{\times} \mathbf{R}$ . Unfortunately expression (13) is a constraint on  $\check{\mathbf{x}}_1$  and  $\check{\mathbf{x}}_2$  and not  $\tilde{\mathbf{x}}_1$  and  $\tilde{\mathbf{x}}_2$ , however note that

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \tilde{\mathbf{x}} = \mathbf{H} \tilde{\mathbf{x}}$$

Thus we can rewrite equation (13) as

$$\tilde{\mathbf{x}}_1^T \mathbf{F} \tilde{\mathbf{x}}_2 = 0 \quad \text{where} \quad \mathbf{F} = \mathbf{H}^T \begin{pmatrix} \mathbf{G} & -\mathbf{G}\check{\boldsymbol{\omega}}_2 \\ -\check{\boldsymbol{\omega}}_1^T \mathbf{G} & \check{\boldsymbol{\omega}}_1^T \mathbf{G} \check{\boldsymbol{\omega}}_2 \end{pmatrix} \mathbf{H}. \quad (14)$$

Equation (14) is the parabolic catadioptric epipolar constraint. It can be verified that

$$\mathbf{F} = \mathbf{K}_{\tilde{\omega}_1}^T \begin{pmatrix} \mathbf{E} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{K}_{\tilde{\omega}_2}. \quad (15)$$

A matrix expressed in this way will be called a catadioptric fundamental matrix. The first thing to note about this new  $\mathbf{F}$  is that since  $H\tilde{\omega} = \tilde{\omega}$ , we must have

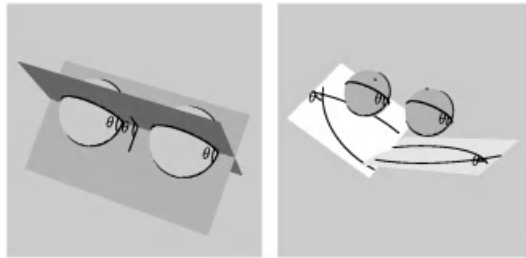
$$\mathbf{F}\tilde{\omega}_2 = 0 \quad \text{and} \quad \mathbf{F}^T\tilde{\omega}_1 = 0.$$

Hence, the lifted left and right images of the absolute conic belong to the left and right nullspace of  $\mathbf{F}$ , respectively. Also since  $\mathbf{G}$  is rank 2,  $\mathbf{F}$  will remain rank 2 because  $(-\tilde{\omega}_1^T \mathbf{G}, \tilde{\omega}_1^T \mathbf{G} \tilde{\omega}_2)$  is linearly dependent on the first three rows.

Note that the expression  $\tilde{\mathbf{x}}_1^T \mathbf{F} \tilde{\mathbf{x}}_2$  is linear in the entries of the matrix  $\mathbf{F}$ . Hence just like in the perspective case, given a set of correspondences a matrix whose entries are the coefficients in the epipolar equation of each entry of  $\mathbf{F}$  can be constructed whose nullspace contains the matrix  $\mathbf{F}$  flattened into a single vector in  $\mathbb{R}^{16}$ . The nullspace can be calculated using singular value decomposition by selecting the vector with the smallest singular value.

## 4 The Space of Catadioptric Fundamental Matrices

In the previous section we found that there is a bilinear constraint on the liftings of corresponding image points in the form of a  $4 \times 4$  matrix analogous to the fundamental matrix for perspective cameras. It would be nice to find the necessary and sufficient conditions that a given matrix be a catadioptric fundamental matrix, that is, of the form (15). We will show that the condition that  $\mathbf{F}$  be rank 2 is necessary but not sufficient.



**Fig. 3.** Left: If two epipolar planes intersect two spheres representing two views at an angle  $\theta$ , then the angle of intersection of the epipolar great circles is also  $\theta$ . Right: By the angle preserving property of stereographic projection, the epipolar circles also must intersect at an angle  $\theta$ .

The condition that we describe is based on the fact that  $\mathbf{F}$  must preserve angles between epipolar circles. In Figure 3 (left) notice that two epipolar planes with a dihedral angle of  $\theta$  intersect two spheres, representing two catadioptric views, in two pairs of great circles, both of which pairs have an angle of intersection of  $\theta$ . Because stereographic projection preserves angles, the projections



of the great circles, two pairs of line images must also intersect at an angle  $\theta$  as shown in Figure 3 (right). The fundamental matrix is a rank 2 space correlation [13] meaning that it maps points to planes, polar planes actually, on which corresponding points must lie.

Assume that points  $\mathbf{p}$  and  $\mathbf{q}$  respectively lie on epipolar circles  $\gamma$  and  $\eta$  and satisfy  $\tilde{\mathbf{p}}^T \mathbf{F} \tilde{\mathbf{q}} = 0$ . The epipolar circle  $\eta$  can be determined from  $\mathbf{F}$  since  $\tilde{\eta} = \mathbf{Q}^{-1} \mathbf{F}^T \tilde{\mathbf{p}}$ . Can  $\tilde{\gamma}$  also be determined from  $\mathbf{F}$ ? If we knew  $\tilde{\mathbf{q}}$  we would have  $\tilde{\gamma} = \mathbf{Q}^{-1} \mathbf{F} \tilde{\mathbf{q}}$ . Let us assume we do not know  $\mathbf{q}$ . But clearly  $\tilde{\gamma}$  is in the range of  $\mathbf{Q}^{-1} \mathbf{F}$ . So assume that  $\mathbf{F} = \lambda_1 \mathbf{u}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{v}_2^T$  is the singular value decomposition of  $\mathbf{F}$  which is rank 2. Thus  $\tilde{\gamma} = \mathbf{Q}^{-1}(\alpha \mathbf{u}_1 + \beta \mathbf{u}_2)$  for some  $\alpha$  and  $\beta$ . Since  $\tilde{\gamma}^T \mathbf{Q} \tilde{\mathbf{p}} = 0$ , solutions unique up to scale are  $\alpha = \tilde{\mathbf{p}}^T \mathbf{u}_2$  and  $\beta = -\tilde{\mathbf{p}}^T \mathbf{u}_1$ . Then  $\tilde{\gamma} = \mathbf{Q}^{-1} \mathbf{W} \tilde{\mathbf{p}}$  where  $\mathbf{W} = \mathbf{u}_1 \mathbf{u}_2^T - \mathbf{u}_2 \mathbf{u}_1^T$ . In summary, corresponding epipolar circles as a function of the point  $\mathbf{p}$  in one image are  $\tilde{\gamma} = \mathbf{Q}^{-1} \mathbf{W} \tilde{\mathbf{p}}$  and  $\tilde{\eta} = \mathbf{Q}^{-1} \mathbf{F}^T \tilde{\mathbf{p}}$ . Note that these two definitions do not depend on any component in  $\tilde{\mathbf{p}}$  orthogonal to  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , we may therefore rewrite them as

$$\tilde{\gamma} = \mathbf{Q}^{-1}(\beta \mathbf{u}_1 - \alpha \mathbf{u}_2) \quad \text{and} \quad \tilde{\eta} = \mathbf{Q}^{-1}(\lambda_1 \alpha \mathbf{v}_1 + \lambda_2 \beta \mathbf{v}_2), \quad (16)$$

hence parameterizing all corresponding epipolar circles.

The sets  $\{\tilde{\gamma}\}$  and  $\{\tilde{\eta}\}$  generated by all choices of  $\alpha$  and  $\beta$  are two lines in circle space. They therefore represent coaxal circles, whose respective intersections have to be the epipoles. In order for the coaxal circles to have real intersection points the line in circle space ought not to intersect the sphere. For some coaxal system  $\mathbf{a} + \lambda \mathbf{b}$  this is the case if and only if

$$\|\mathbf{a} + \lambda \mathbf{b}\|_Q^2 = (\mathbf{a} + \lambda \mathbf{b})^T \mathbf{Q} (\mathbf{a} + \lambda \mathbf{b}) > 0 \quad (17)$$

for all  $\lambda$  which is the case if and only if the discriminant of the left hand side as a polynomial in  $\lambda$  is negative. The discriminant being negative gives

$$\langle \mathbf{a}, \mathbf{b} \rangle_Q^2 < \|\mathbf{a}\|_Q^2 \|\mathbf{b}\|_Q^2. \quad (18)$$

**Lemma 3.** If (18) is satisfied then for any two circles  $\alpha_1 \mathbf{a} + \beta_1 \mathbf{b}$  and  $\alpha_2 \mathbf{a} + \beta_2 \mathbf{b}$  in the coaxal space,

$$0 \leq \frac{\langle \alpha_1 \mathbf{a} + \beta_1 \mathbf{b}, \alpha_2 \mathbf{a} + \beta_2 \mathbf{b} \rangle_Q^2}{\|\alpha_1 \mathbf{a} + \beta_1 \mathbf{b}\|_Q^2 \|\alpha_2 \mathbf{a} + \beta_2 \mathbf{b}\|_Q^2} \leq 1,$$

in which case the angle between them is well-defined.

**Proof:** From (17) and from the fact that  $(\beta_1 \alpha_2 - \alpha_1 \beta_2)^2 (\langle \mathbf{a}, \mathbf{b} \rangle_Q^2 - \|\mathbf{a}\|_Q^2 \|\mathbf{b}\|_Q^2) = \langle \alpha_1 \mathbf{a} + \beta_1 \mathbf{b}, \alpha_2 \mathbf{a} + \beta_2 \mathbf{b} \rangle_Q^2 - \|\alpha_1 \mathbf{a} + \beta_1 \mathbf{b}\|_Q^2 \|\alpha_2 \mathbf{a} + \beta_2 \mathbf{b}\|_Q^2$ .  $\square$

**Definition.** When we say that a rank 2 space correlation  $\mathbf{F}$  preserves epipolar angles (i.e. angles between epipolar circles) we mean that for all  $\tilde{\mathbf{p}}$ ,

$$\frac{\langle \mathbf{W} \tilde{\mathbf{p}}_1, \mathbf{W} \tilde{\mathbf{p}}_2 \rangle_Q^2}{\|\mathbf{W} \tilde{\mathbf{p}}_1\|_Q^2 \|\mathbf{W} \tilde{\mathbf{p}}_2\|_Q^2} = \frac{\langle \mathbf{F}^T \tilde{\mathbf{p}}_1, \mathbf{F}^T \tilde{\mathbf{p}}_2 \rangle_Q^2}{\|\mathbf{F}^T \tilde{\mathbf{p}}_1\|_Q^2 \|\mathbf{F}^T \tilde{\mathbf{p}}_2\|_Q^2}. \quad (19)$$

Equation (19) is obtained by substituting definitions of  $\tilde{\gamma}_i$  and  $\tilde{\eta}_i$  from (16)

into (7) while noticing that the  $Q^{-1}$ 's cancel. This definition skirts the issue of whether this formula actually implies angles are preserved, but if angles are preserved then this formula must be true. Whether the converse is true turns out to be irrelevant.

**Proposition 3.** If (18) is satisfied by left and right singular vectors of a rank 2 space correlation  $F$  having SVD  $\lambda_1 \mathbf{u}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{v}_2^T$  then the following statement is true.  $F$  preserves angles between epipolar circles if and only if

$$\begin{aligned} & \left( \|\mathbf{u}_1\|_Q^2 \|\mathbf{u}_2\|_Q^2 \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q^2 = \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q^2 \|\mathbf{v}_1\|_Q^2 \|\mathbf{v}_2\|_Q^2 \right. \\ & \text{and } \lambda_1 \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q \|\mathbf{v}_1\|_Q^2 = -\lambda_2 \|\mathbf{u}_2\|_Q^2 \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q \Big) \\ & \text{or } \left( \lambda_1^2 \|\mathbf{u}_1\|_Q^2 \|\mathbf{v}_1\|_Q^2 = \lambda_2^2 \|\mathbf{u}_2\|_Q^2 \|\mathbf{v}_2\|_Q^2, \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = 0 \text{ and } \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0 \right) \end{aligned} \quad (20)$$

**Proof:** See appendix for  $(\rightarrow)$ .

**Corollary 2.** A matrix  $E$  is an essential matrix if and only if the matrix  $E^{(4)}$  satisfies (20), where we define  $E^{(4)} = \begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix}$ .

**Proof:** A matrix  $E^{(4)}$  has SVD  $\lambda_1 \mathbf{u}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{v}_2^T$  where  $\mathbf{u}_i, \mathbf{v}_i \in \pi_\infty$ . Because they lie on  $\pi_\infty$ , the dot product reduces to the Euclidean dot product and therefore (18) is just the Schwartz inequality satisfied by any vectors, and also by the properties of the SVD,  $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0$ ,  $\|\mathbf{u}_i\|_Q = \|\mathbf{v}_i\|_Q = 1$ .

If  $E$  is an essential matrix then  $\lambda_1 = \lambda_2$  and then the second clause of (20) is satisfied. Therefore  $E^{(4)}$  is angle preserving.

If  $E^{(4)}$  is angle preserving then since  $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0$ , the second clause applies and  $\lambda_1^2 = \lambda_2^2$ , thus  $E$  is an essential matrix.  $\square$

**Lemma 4.** If a rank 2 space correlation  $F$  preserves angles and the transformation  $K \in \mathcal{L}$  then  $FK$  preserves angles between epipolar circles. Similarly for  $K^T F$ .

**Proof:**  $\langle K^T F^T \tilde{\mathbf{p}}_1, K^T F^T \tilde{\mathbf{p}}_2 \rangle_Q = \langle F^T \tilde{\mathbf{p}}_1, F^T \tilde{\mathbf{q}}_2 \rangle_Q$  since  $KQK^T = \lambda Q$ . For the other notice by relabeling the SVD, Proposition 3 implies that if  $F$  is angle preserving then  $F^T$  is too.  $\square$

**Lemma 5.** If  $\mathbf{a}$  and  $\mathbf{b}$  satisfy (18) then the nullspace of  $(\mathbf{a}^T, \mathbf{b}^T)$  intersects  $Q$ .

**Proof:**  $Q^{-1}\mathbf{a}$  and  $Q^{-1}\mathbf{b}$  also satisfy (18) and their span is a line not intersecting the sphere. Let  $\pi_1$  and  $\pi_2$  be two lines through the span and tangent to the sphere at points  $\mathbf{p}_1$  and  $\mathbf{p}_2$ . Both  $\mathbf{p}_i$  are orthogonal to  $\mathbf{a}$  and  $\mathbf{b}$  because they lie on the polar planes of  $Q^{-1}\mathbf{a}$  and  $Q^{-1}\mathbf{b}$  and therefore satisfy  $\mathbf{p}_i^T Q Q^{-1} \mathbf{a} = \mathbf{p}_i^T Q Q^{-1} \mathbf{b} = 0$ . The  $\mathbf{p}_i$ 's are therefore a basis of the nullspace which obviously intersects the sphere.  $\square$

**Theorem.** A rank 2 space correlation  $F$  can be decomposed as  $K_1^T E^{(4)} K_2$  where  $K_i \in \mathcal{L}_N$  and  $E$  is an essential matrix if and only if (20) and (18) are satisfied by the vectors of its singular value decomposition.

**Proof:** Assume  $F$  is a rank 2 space correlation for which there exists  $K_{i=1,2} \in \mathcal{L}_N$  and an essential matrix  $E$  such that  $F = K_1^T E^{(4)} K_2$ . First, in Corollary 2 we saw that  $E^{(4)}$ 's singular vectors satisfy (18), since  $K_{i=1,2} \in \mathcal{L}$ , the inequality is preserved by the pre- and post-multiplication of these matrices, implying (18)

is satisfied by  $\mathbf{F}$  as well (even though the singular vectors change the spans are equal). By Corollary 2,  $\mathbf{E}^{(4)}$  preserves angles between epipolar circles, and therefore by Lemma 4,  $\mathbf{E}\mathbf{K}_2$  and then  $\mathbf{K}_1^T \mathbf{E}^{(4)} \mathbf{K}_2$  also preserve angles between epipolar circles. By Proposition 3,  $\mathbf{K}_1^T \mathbf{E}^{(4)} \mathbf{K}_2$ , a rank 2 space correlation preserving epipolar angles and satisfying (18) must satisfy condition (20).

Now assume that  $\mathbf{F}$  is an angle preserving, rank 2 space correlation satisfying (18), show that it is decomposable. Since it is rank 2 and satisfies (17), by Lemma 5 there is some  $\tilde{\omega}_1$  inside the sphere such that  $\mathbf{F}^T \tilde{\omega} = 0$  and some  $\tilde{\omega}_2$  inside the sphere such that  $\mathbf{F} \tilde{\omega} = 0$ . If we calculate  $\mathbf{K}_{\tilde{\omega}_1}^{-1}$  we find that for some  $\mathbf{a}, \mathbf{b}$  that  $\mathbf{K}_{\tilde{\omega}_1}^{-1} = (\mathbf{a}, \mathbf{b}, \alpha \tilde{\omega}', \beta \tilde{\omega})$ . The important point is that if the singular value decomposition of  $\mathbf{F} = \lambda_1 \mathbf{u}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{v}_2^T$ , then because  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are orthogonal to  $\tilde{\omega}_1$  which is the last column of  $\mathbf{K}_{\tilde{\omega}_1}$ ,  $\mathbf{K}_{\tilde{\omega}_1}^{-T} \mathbf{u}_i \in \pi_\infty$ .  $\mathbf{K}_{\tilde{\omega}_2}$  has the same effect on  $\mathbf{v}_i$ . Therefore

$$\mathbf{K}_{\tilde{\omega}_1}^{-T} \mathbf{F} \mathbf{K}_{\tilde{\omega}_2}^{-1} = \begin{pmatrix} \mathbf{E} & 0 \\ 0 & 0 \end{pmatrix}.$$

We now show that  $\mathbf{E}$  is an essential matrix. Since  $\mathbf{F}$  preserves angles between epipolar circles, so does  $\mathbf{K}_{\tilde{\omega}_1}^{-T} \mathbf{F} \mathbf{K}_{\tilde{\omega}_2}^{-1}$ . Since it preserves angles, by Corollary 2, it must be an essential matrix with equal non-null singular values. Thus  $\mathbf{F} = \mathbf{K}_{\tilde{\omega}_1}^T \mathbf{E} \mathbf{K}_{\tilde{\omega}_2}$  for some  $\mathbf{K}_{\tilde{\omega}_i} \in \mathcal{L}_N$  and some essential matrix  $\mathbf{E}$ .  $\square$

## 5 Conclusion

In this paper we introduced the spherical circle space to describe points and line images in parabolic catadioptric views. We described the class of linear transformations in that space which turned out to be the Lorentz group. We derived the catadioptric fundamental matrix and proved that the lifted image of the absolute conic belongs to its nullspace. Based on the fact that angles between epipolar circles are preserved we proved necessary and sufficient conditions for a matrix to be a catadioptric fundamental matrix.

## Appendix (Proof of Proposition 3)

( $\implies$ ) Since (18) is true, (19) is well-defined for all  $\tilde{\mathbf{p}}_1$  and  $\tilde{\mathbf{p}}_2$ . It is therefore true when  $\tilde{\mathbf{p}}_1 = \mathbf{u}_1$  and  $\tilde{\mathbf{p}}_2 = \alpha \mathbf{u}_1 + \mathbf{u}_2$ . Substitute these definitions into (19) and cross-multiply the denominators. If the both sides are equal for all  $\alpha$  then for all  $\alpha$  the polynomial

$$f(\alpha) \equiv \langle -\mathbf{u}_2, \mathbf{u}_1 - \alpha \mathbf{u}_2 \rangle_Q^2 \|\alpha \lambda_1 \mathbf{v}_1\|_Q^2 \|\alpha \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2\|_Q^2 - \langle \lambda_1 \mathbf{v}_1, \alpha \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 \rangle_Q^2 \|\mathbf{u}_1 - \alpha \mathbf{u}_2\|_Q^2 = 0. \quad (21)$$

In order that this polynomial be zero everywhere all its coefficients must be zero. Then the coefficients of  $\alpha^0$ ,  $\alpha^1$ , and  $\alpha^2$  generate the three equations below which have been divided by  $\lambda_1^i \lambda_2^j$  where appropriate ( $\lambda_i > 0$  by assumption):

$$\langle \mathbf{u}_1, \mathbf{u}_1 \rangle_Q \langle \mathbf{u}_2, \mathbf{u}_2 \rangle_Q \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q^2 = \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q^2 \langle \mathbf{v}_1, \mathbf{v}_1 \rangle_Q \langle \mathbf{v}_2, \mathbf{v}_2 \rangle_Q, \quad (22)$$

$$\begin{aligned} \lambda_1 \|\mathbf{v}_1\|_Q^2 \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q (\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q^2 - \|\mathbf{u}_1\|_Q^2 \|\mathbf{u}_2\|_Q^2) \\ = -\lambda_2 \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q \|\mathbf{u}_2\|_Q^2 (\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q^2 - \|\mathbf{v}_1\|_Q^2 \|\mathbf{v}_2\|_Q^2), \end{aligned} \quad (23)$$

$$\begin{aligned} \lambda_1^2 \|\mathbf{v}_1\|_Q^4 (\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q^2 - \|\mathbf{u}_1\|_Q^2 \|\mathbf{u}_2\|_Q^2) \\ = -\lambda_2^2 \|\mathbf{u}_2\|_Q^4 (\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q^2 - \|\mathbf{v}_1\|_Q^2 \|\mathbf{v}_2\|_Q^2). \end{aligned} \quad (24)$$

Condition (18) and hence (17) implies that  $\|\mathbf{u}_i\|_Q^2 > 0$  and  $\|\mathbf{v}_i\|_Q^2 > 0$ . Thus if neither  $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = 0$  nor  $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0$ , then we can solve for  $\|\mathbf{v}_1\|_Q^2$  in equation (22), and substitute into (23) and (24); then both reduce to  $\lambda_1 \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q \|\mathbf{v}_1\|_Q^2 = -\lambda_2 \|\mathbf{u}_2\|_Q^2 \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q$ . This satisfies the first clause of (20).

Otherwise if  $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0$ , then (22) implies that  $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = 0$ ; the converse is true as well by (22). Substituting  $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_Q = 0$  and  $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_Q = 0$  into equation (23) gives no constraint and into equation (24) yields  $\lambda_1^2 \|\mathbf{u}_1\|_Q^2 \|\mathbf{v}_1\|_Q^2 = \lambda_2^2 \|\mathbf{u}_2\|_Q^2 \|\mathbf{v}_2\|_Q^2$ . Then the second clause of (20) is satisfied. Therefore if  $\mathbf{F}$  preserves the angles between epipolar circles according to the definition given above and has left and right singular vectors satisfying (18), then one of the conditions in (20) is true. □

## References

1. R. Benosman and S.B. Kang. *Panoramic Vision*. Springer-Verlag, 2000.
2. O. Faugeras, Q.-T. Luong, and T. Papadopoulos. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2001.
3. C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 43:223–243, 2001.
4. C. Geyer and K. Daniilidis. Structure and motion from uncalibrated catadioptric views. In *IEEE Conf. Computer Vision and Pattern Recognition*, Hawaii, Dec. 11–13, 2001.
5. J. Gluckman and S.K. Nayar. Ego-motion and omnidirectional cameras. In *Proc. Int. Conf. on Computer Vision*, pages 999–1005, Bombay, India, Jan. 3–5, 1998.
6. R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge Univ. Press, 2000.
7. V. Heine. *Group Theory in Quantum Mechanics*. Pergamon Press, Oxford, 1960.
8. S.B. Kang. Catadioptric self-calibration. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages I–201–207, Hilton Head Island, SC, June 13–15, 2000.
9. Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. Sastry. Rank conditions of the multiple view matrix. Technical Report UILU-ENG 01-2214 (DC-220), University of Illinois at Urbana-Champaign, CSL-Technical Report, June 2001.
10. S. Nayar. Catadioptric omnidirectional camera. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 482–488, Puerto Rico, June 17–19, 1997.
11. T. Needham. *Visual Complex Analysis*. Clarendon Press, Oxford, 1997.
12. D. Pedoe. *Geometry: A comprehensive course*. Dover Publications, New York, NY, 1970.
13. J. Semple and G. Kneebone. *Algebraic Projective Geometr.* Oxford University Press, 1979.
14. T. Svoboda, T. Pajdla, and V. Hlavac. Epipolar geometry for panoramic cameras. In *Proc. 5th European Conference on Computer Vision*, pages 218–231, 1998.

# Building Architectural Models from Many Views Using Map Constraints

D.P. Robertson and R. Cipolla

University of Cambridge, Department of Engineering  
Trumpington Street, Cambridge CB2 1PZ.  
dpr20@eng.cam.ac.uk, cipolla@eng.cam.ac.uk

**Abstract.** This paper describes an interactive system for creating geometric models from many uncalibrated images of architectural scenes. In this context, we must solve the structure from motion problem given only few and noisy feature correspondences in non-sequential views. By exploiting the strong constraints obtained by modelling a map as a single affine view of the scene, we are able to compute all 3D points and camera positions simultaneously as the solution of a set of linear equations. Reconstruction is achieved without making restrictive assumptions about the scene (such as that reference points or planes are visible in all views). We have implemented a practical interactive system, which has been used to make large-scale models of a variety of architectural scenes. We present quantitative and qualitative results obtained by this system.

## 1 Introduction

Despite much progress [1,15,21] in the development of completely automatic techniques for obtaining geometric models from images, the best-looking architectural models are still produced interactively [6,5,17]. Interactive approaches exploit the user's higher-level knowledge to solve the difficult problems of identifying geometrically important features and wide-baseline matching. Existing interactive approaches have used one or relatively few calibrated or uncalibrated views in order to build models of a few buildings. In this paper, we address some of the problems associated with making much larger models from an arbitrarily large number of uncalibrated views of many buildings.

### 1.1 Previous Work

Given feature correspondences, the optimal solution for camera parameters and structure may be determined using bundle adjustment [18], which is used to distribute back projection error optimally across all feature measurements. In the context of an interactive system, we must solve the structure from motion problem given only few and noisy feature correspondences defined in images obtained from sparse viewpoints. This presents two problems: (i) bundle adjustment will only succeed provided a sufficiently good initial guess can be obtained and (ii)

even an ‘optimal’ reconstruction is no guarantee of a subjectively good-looking model.

One solution to the problem of obtaining an initial guess is to estimate camera parameters and structure simultaneously as the solution of a linear equation. Although image coordinates have a non-linear relationship with 3D coordinates under perspective projection, it is well known [6,17,16] that feature correspondences allow a linear solution for structure and camera positions if camera calibration and orientation are known. In [6] and [17] camera orientation is determined for calibrated cameras from prior knowledge of line directions. In [12] and [5] camera calibration and orientation are determined simultaneously from three vanishing points corresponding with three orthogonal directions. This approach allows a two-view initialisation of a Euclidean frame but does not address the problem of how extra views should be registered in that frame, unless we make the restrictive assumption that three vanishing points corresponding with known and orthogonal directions are visible in *all* images. Furthermore, this approach has the disadvantage that camera calibration cannot be determined for the degenerate (but very common) case of vanishing points lying at (or near) infinity in the image plane, e.g. in a photograph of a wall parallel to the image plane. [16] generalises the concept in [5,12] to any three points lying on a reference plane but relies on the almost equally restrictive assumption that four points on the reference plane are visible in all images (or at least that the reference plane is visible in all images).

Another solution is to register uncalibrated cameras sequentially [1]. A two-view initialisation defines a projective frame via the fundamental matrix. A partial reconstruction may be computed within this frame using feature correspondences. Then additional views are registered one at a time using the Discrete Linear Transformation [20]. Having determined the projection matrix for an additional view, structure may be computed for all correspondences defined in two or more views and the partial reconstruction is extended.

Whilst this approach is effective in the context of tracked features in video frames [1,8], it has severe limitations in the context of interactive systems:

1. Given only a few noisy feature correspondences partial reconstructions are likely to be quite inaccurate. Bundle adjustment may be used to improve estimated camera parameters and structure but this approach often fails in practice due to convergence to a local minima. In any case, carrying out bundle adjustment after the addition of each subsequent viewpoint is very computationally expensive.
2. Accurate camera registration by DLT depends on the accuracy of the partial reconstruction. Some reconstructed points may be quite degenerate with respect to included views and therefore inaccurate. Such points may severely compromise the accuracy of the DLT.
3. Some viewpoints may be degenerate with respect to the partial reconstruction. DLT requires at least 6 points, two of which must be non-coplanar with the remainder.

4. It is difficult to adapt the sequential approach to non-sequential image data. Given the likely inaccuracy of the partial solution and possible degeneracy of successive viewpoints with respect to that partial reconstruction, it is not clear in which order successive viewpoints should be registered within our euclidean (or projective) frame.

The second problem associated with few and noisy feature correspondences is that of obtaining a sufficiently accurate reconstruction. Architectural scenes typically contain a large number of parallel and perpendicular elements and it is subjectively very important that these relationships should be preserved as far as possible in the final model. However, small errors associated with the registration of nearby viewpoints may accumulate throughout a large set of images such that absolute errors become large. This problem may be particularly severe in cases where it is impossible to obtain images from a suitably wide range of viewpoints, e.g. a city street.

In order to address this problem, some interactive systems constrain the reconstruction process by exploiting the user's higher-level knowledge about parallelism and orthogonality. In [17] for example, scene structure is determined subject to constraints on (known) line directions and plane normals. This type of approach has the considerable disadvantage that it is not directly extensible to data sets comprising images of buildings with unknown and different orientations. Debevec et al [6] describe a system that allows the user to parameterise the scene in terms of primitives: simple geometric building blocks such as cuboids and prisms that can be combined to make more complex models. Such systems may produce excellent results but not all scenes can be expressed so simply in terms of a few geometric building blocks. Furthermore it is not always possible to find viewpoints such that a sufficiently large proportion of each primitive is visible in any one image.

## 1.2 Approach

We are concerned with modelling large architectural scenes. In this context, it is not always possible to assume that all buildings have known or at least similar orientation or that a single plane will be visible in all views. Nor will it always be possible to obtain photographs containing three vanishing points associated with non-degenerate orthogonal directions. We proceed by making only the following assumptions: firstly that the vertical direction can be identified in all views and secondly that we have a map of the scene.

Whilst a number of previous works have explored the possibility of using a map as an affine view of a scene in combination with one [22] or two [13] perspective views (and additional scene constraints [3]), using a map in combination with many perspective views has not been considered.

We use a map along with the user's prior knowledge of parallelism in order to determine camera orientation and calibration. This allows us to formulate the *uncalibrated* structure from motion problem as a simple *linear* equation without the problem of a possibly degenerate approach to calibration or the need for

restrictive assumptions about the scene (such as that reference points are visible in all images). In addition the map provides a strong *global* constraint on structure from motion, allowing high quality reconstruction from a few, noisy feature correspondences.

We describe a complete interactive system for architectural modelling. In comparison with existing systems, our system allows us to build much larger-scale models more quickly.

### 1.3 Structure of This Paper

This paper is arranged as follows. Section 2 reviews briefly the theory of perspective and affine projection. Section 3 describes how parallelism and map constraints may be used to determine camera calibration and camera registration in the map-based frame. Section 4 explains how these techniques are implemented in a working system. Finally Section 5 presents some experiments to demonstrate the efficacy of these ideas when applied to building large-scale architectural models.

## 2 Theory and Notation

A 3D point  $\mathbf{X}_j = [X \ Y \ Z \ 1]^t$  projects into an image plane according to a general  $3 \times 4$  projection matrix  $\mathbf{P}_i$ :

$$\mathbf{x}_{ij} \sim \mathbf{P}_i \mathbf{X}_j \quad (1)$$

where  $\mathbf{x}_{ij} = [u \ v \ 1]^t$  is an image coordinate and  $\sim$  means equality up to scale.

A projection matrix corresponding with a perspective camera may be decomposed as:

$$\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i \ -\mathbf{R}_i^t \mathbf{T}_i] \quad (2)$$

where  $\mathbf{R}_i$  is the  $3 \times 3$  rotation matrix describing the orientation of the camera and  $\mathbf{T}_i$  is the position of the camera. Camera calibration matrix  $\mathbf{K}_i$  is of the form:

$$\mathbf{K}_i = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where  $\alpha_u$  and  $\alpha_v$  are scale factors,  $s$  is skew, and  $[u_0 \ v_0 \ 1]^t$  is the principal point.

The map coordinate of a 3D point is dependent on scene  $X, Y$  position but not on  $Z$ -axis height. Thus, a map may be modelled as an affine (or orthographic) view of the scene with projection matrix:

$$\mathbf{P}_{\text{map}} \sim \begin{bmatrix} \sigma & 0 & 0 & X_0 \\ 0 & \sigma & 0 & Y_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where  $\sigma$  is the map's scale and  $[X_0 \ Y_0]^t$  is the world  $X, Y$  coordinate of the map's origin.



### 3 Map Constraints

#### 3.1 Single View Constraints

The image coordinate  $\mathbf{e}_i$  of the vertical vanishing point in image  $i$  may be determined from the image of two or more vertical lines. This vanishing point is the projection of the point  $[0 \ 0 \ 1 \ 0]^t$  at infinity and is the epipole corresponding with the affine map camera.

We begin by rectifying our images such that vertical lines in the world map to vertical lines in the image plane. We seek a  $3 \times 3$  homography such that:

$$\mathbf{H}_i [e_1 \ e_2 \ e_3]^t = [0 \ 1 \ 0]^t \quad (5)$$

where  $\mathbf{e}_i = [e_1 \ e_2 \ e_3]^t$  is the image coordinate of the vanishing point corresponding with the vertical direction in the world and  $|\mathbf{e}_i| = 1$ . It is convenient to choose  $\mathbf{H}_i$  such that  $\mathbf{H}_i$  is a rotation matrix and:

$$\mathbf{H}_i [e_2 \ -e_1 \ 0]^t = [1 \ 0 \ 0]^t \quad (6)$$

This transformation preserves scale along the line in the image plane that is parallel with the image of the horizon and passes through the point  $[0 \ 0 \ 1]^t$ . Figure 1 illustrates transformation of an image plane by such a homography.



(i)



(ii)

**Fig. 1.** (i) Vertical lines marked in an image. (ii) The image warped by a homography  $\mathbf{H}$  such that vertical lines project to vertical lines in the transformed image plane

3D points  $\mathbf{X}_j$  project into our transformed image plane according to the following equation:

$$\mathbf{x}'_{ij} \sim \mathbf{H}_i \mathbf{K}_i [\mathbf{R}_i \ -\mathbf{R}_i^t \mathbf{T}_i] \mathbf{X}_j \sim \hat{\mathbf{P}}_i \mathbf{X}_j \quad (7)$$

where  $\mathbf{x}'_{ij} = [u' \ v' \ 1]$  and  $\hat{\mathbf{P}}_i$  has the general form:

$$\hat{\mathbf{P}}_i = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (8)$$

In the rectified image,  $u'$  coordinates depend only on world  $X, Y$  coordinates and are independent of world  $Z$  coordinates. Thus  $p_{13} = p_{33} = 0$  and transformed image coordinates  $\mathbf{X}'$  are related to map  $X, Y$  coordinates by the simple 1D projection relationship:

$$\begin{bmatrix} u' \\ 1 \end{bmatrix} \sim \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{p}_i \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (9)$$

where  $\mathbf{p}_i$  is a 1D projection matrix. Given five or more correspondences between map  $X, Y$  coordinates and image  $u'$  coordinates (for points or vertical lines), it is possible to solve for all the elements of  $\mathbf{p}_i$  as the solution of a linear equation. We can also use as correspondences horizontal vanishing points in the image corresponding with known directions  $\mathbf{d}_k$  on the map. Map directions can be estimated from a single line or two or more parallel lines. Thus, we may solve for  $\mathbf{p}_i$  given five point correspondences or four point correspondences plus one horizontal vanishing point or three point correspondences plus two horizontal vanishing points.

In a similar manner to the  $3 \times 4$  projection matrix in equation (1), the projection matrix  $\mathbf{p}_i$  for a 1D camera may be decomposed as:

$$\mathbf{p}_i = [\mathbf{k}_i \mathbf{r}_i \quad -\mathbf{r}_i^t \mathbf{t}_i] \quad (10)$$

where  $\mathbf{k}_i$  is upper triangular,  $\mathbf{r}_i$  is  $2 \times 2$  rotation matrix describing the 1D camera's orientation about the vertical axis, and  $\mathbf{t}_i$  is the camera's  $X, Y$  map position. Note that there is an ambiguity associated with our solution for the elements of  $\mathbf{p}_i$  relating to whether 3D points are in front of or behind the camera. If  $\mathbf{r}_i$  is such that  $X, Y$  coordinates are behind the 1D camera then we should replace it with  $-\mathbf{r}_i$ .

The 1D projection matrix  $\mathbf{p}_i$  describes vertical axis orientation, map position, and calibration for the 1D camera. Figure (2) illustrates the registration of 1D cameras on the map using five map coordinates, four map coordinates plus one map direction, and three map coordinates plus two map directions.

Given the 1D projection matrix  $\mathbf{p}_i$ , we may determine  $\mathbf{K}_i$  and  $\mathbf{R}_i$  for the original view. From (7), and considering only the first  $3 \times 3$  sub matrix:

$$\mathbf{H}_i \mathbf{K}_i \mathbf{R}_i \sim \hat{\mathbf{P}}_i \quad (11)$$

Since  $\mathbf{R}_i$  is a rotation matrix,  $\mathbf{R}_i \mathbf{R}_i^t = \mathbf{I}$ . Thus:

$$\mathbf{H}_i \mathbf{K}_i \mathbf{K}_i^t \mathbf{H}_i^t = \lambda \hat{\mathbf{P}}_i \hat{\mathbf{P}}_i^t \quad (12)$$

where  $\lambda$  is an unknown scale factor. This relationship contains three equations in the known elements of  $\hat{\mathbf{P}}_i$  and the unknown elements of  $\mathbf{K}_i \mathbf{K}_i^t$  and  $\lambda$ . By assuming that pixels are square ( $s = 0$  and  $\alpha_u = \alpha_v$ ) and that the principal point  $[u_0 \ v_0 \ 1]^t$  lies at the image centre, we are able to solve this equation for  $\alpha^2$  ( $= \alpha_u^2 = \alpha_v^2$ ). This set of assumptions is at least sufficiently good to allow the approach to succeed for a wide range of cameras. In any case they may be relaxed during the subsequent multi-camera bundle adjustment stage.

Finally we determine  $\mathbf{R}_i$ . The epipole  $\mathbf{e}_i$  is the projection of the point  $[0 \ 0 \ 1 \ 0]^t$ :

$$\mathbf{e}_i \sim \mathbf{K}_i \mathbf{R}_i [0 \ 0 \ 1 \ 0]^t \quad (13)$$

Thus the third column of  $\mathbf{R}_i$  is simply  $\pm \mathbf{K}_i^{-1} \mathbf{e}_i$ . This sign ambiguity arises because the epipole may correspond with the projection of the 'up' or the 'down' direction. We resolve this ambiguity by assuming that photographs are incorporated into our system 'right way up', i.e. the sign of  $v$  coordinate of  $\mathbf{e}_i$  indicates whether the epipole is 'up' or 'down'. Equation (13) provides two constraints on the three parameters of  $\mathbf{R}_i$ . Equation (11) allows us to fix the remaining parameter.

### 3.2 Multiple View Constraints

Using the single view constraints described in the previous section, we can determine camera calibration, orientation, and map  $X, Y$  position. However, the  $Z$  coordinate of each camera (height) and 3D structure is unknown.

Having determined camera calibration and orientation, we exploit the linear constraint provided by the following equation (as in [6,17]):

$$\mathbf{K}_i^{-1} \mathbf{x}_{ij} \sim [\mathbf{R}_i \ -\mathbf{R}_i^t \mathbf{T}_i] \mathbf{X}_j \quad (14)$$

This relationship provides two independent linear equations in the elements of unknown structure  $\mathbf{X}_j$  and 3D camera positions  $\mathbf{T}_i$ .

Optionally, we may wish to employ the constraint that some lines have a known direction  $\mathbf{d}_k$ :

$$\mathbf{d}_k \times \mathbf{L}_l = 0 \quad (15)$$

where  $\times$  denotes the cross product,  $\mathbf{L}_l = \mathbf{X}_t - \mathbf{X}_u$ , and  $\mathbf{X}_t$  and  $\mathbf{X}_u$  are two 3D points connected by a line. We set  $\mathbf{d}_0 = [0 \ 0 \ 1]^t$  for the vertical direction and  $\mathbf{d}_k = [x \ y \ 0]^t$  for horizontal directions estimated from the map.

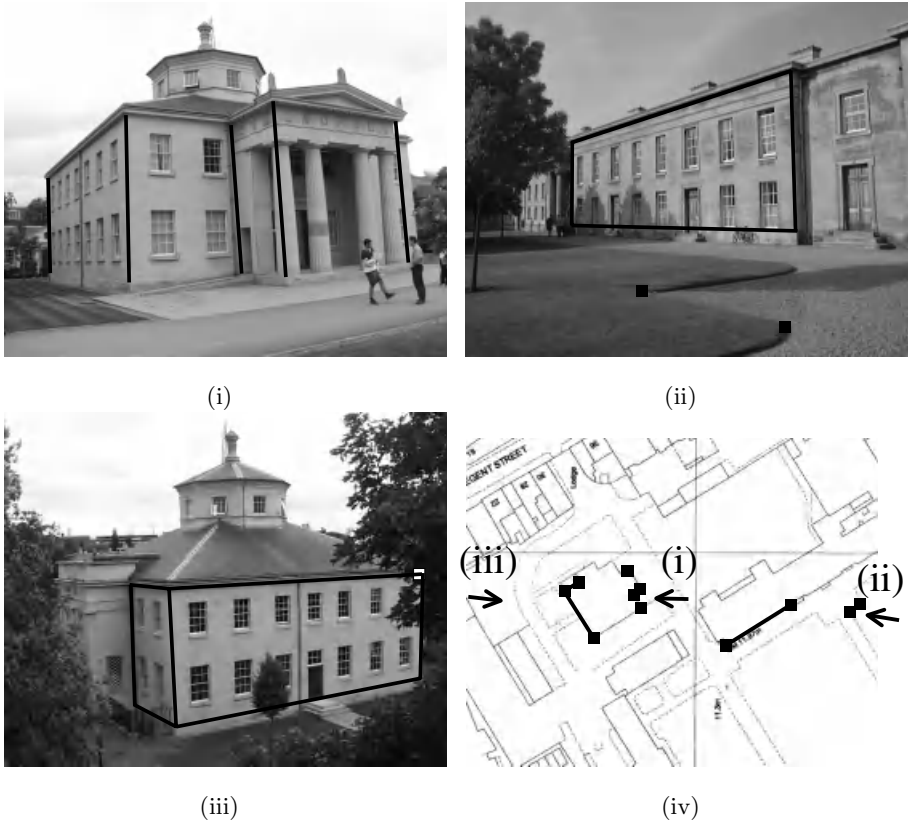
We may assemble all such equations (as 14 and 15) into a matrix equation of the form:

$$\mathbf{A} \mathbf{X} = 0 \quad (16)$$

where  $\mathbf{X}$  comprises all unknown structure  $\mathbf{X}_j$  and camera positions  $\mathbf{T}_i$ .

This equation can be solved easily for hundreds of vertices using the singular value decomposition. For more vertices, we should resort to an appropriate sparse matrix technique.

Note that there is a sign ambiguity associated with the solution for  $\mathbf{X}$  in (16). This ambiguity may be resolved by ensuring all points (or at least the majority



**Fig. 2.** A view is registered in the map coordinate frame using the vertical vanishing point plus (i) five correspondences between points or vertical lines in the image and points on the map, or (ii) four correspondences plus one horizontal direction, or (iii) three correspondences plus two horizontal directions.

of points in case of noisy data) are reconstructed in front of the cameras in which they are visible. In addition, we must fix the height of one point, e.g. the height of the first camera can be set to 0.

### 3.3 Optimization

From an initial guess at projection matrices and structure we can optimise camera parameters and structure by bundle adjustment (see Section 4.2).

## 4 Implementation

### 4.1 Algorithm

Our approach to modelling is as follows:

1. We transform each image by a homography such that vertical lines in the world project to vertical lines in the image plane
2. For each camera we estimate absolute orientation and camera calibration using the single-view constraints described in Section 3.1.
3. Given camera calibration and orientation and (optionally) extra scene constraints, we compute camera positions and scene structure as the solution of a linear equation as described in Section 3.2.
4. We optimise scene structure and camera parameters using bundle adjustment.

## 4.2 Bundle Adjustment

We wish to optimise camera parameters and structure subject to the constraint that parallel lines are parallel (both vertical and horizontal) and our knowledge of the affine projection matrix for the map.

We adjust the parameters of projection matrices  $\mathbf{P}_i$ , structure  $\mathbf{X}_j$  and directions  $\mathbf{d}_k$  in order to minimise back-projection error  $\epsilon$ :

$$\epsilon = \sum_{ij} |\mathbf{P}_i \mathbf{X}_j - \mathbf{X}_{ij}|^2 + \sum_{lk} |\mathbf{d}_k \times \mathbf{L}_l|^2 \quad (17)$$

An initial guess at horizontal directions is obtained from the map. The affine projection matrix for the map camera and the vertical direction are fixed.

We have implemented the fast bundle adjustment algorithm in [18]. We extend this algorithm by including extra parameters corresponding with the unknown focal length of each camera and line directions  $\mathbf{d}_k$ . In addition we provide the facility to incorporate a (fixed) affine camera (the map).

This algorithm allows us to introduce covariance matrices describing the error p.d.f. associated with feature coordinate measurements. In practice, this allows us to account for the fact that map data may be substantially less accurate than image data.

## 5 Results

### 5.1 Camera Registration

Figures 3(i, ii, iii) show representative images from a 16-image sequence. This sequence was obtained on level ground using a digital camera mounted on a tripod, which was positioned at regular intervals along a straight line. Using a 1:500 scale Ordnance Survey map, camera registration was determined by the approach described in this paper.

Figure 3(iv) compares estimated camera  $X, Y$  positions with ground truth data. Most of the errors associated with estimated camera positions are in the direction parallel to the viewing direction. This is due to the inevitable ambiguity between depth and focal length in views of scenes that do not occupy a substantial range of depth (RMS error associated with estimated focal lengths was

12.7% of the true value). Recovered camera heights are much more consistent because there is no such ambiguity in the estimate (see Figure 3(v)).

Note that in this sequence the sequential approach to camera registration failed after incorporating only the first few images due to the failure of bundle adjustment to converge to the global minimum. Defining only 10 feature correspondences on the map has allowed registration of all images simultaneously and accurately as the solution of a linear equation. The map-based approach obtains a good reconstruction more quickly with fewer feature correspondences - this is important for an interactive system.

## 5.2 Large-Scale Models

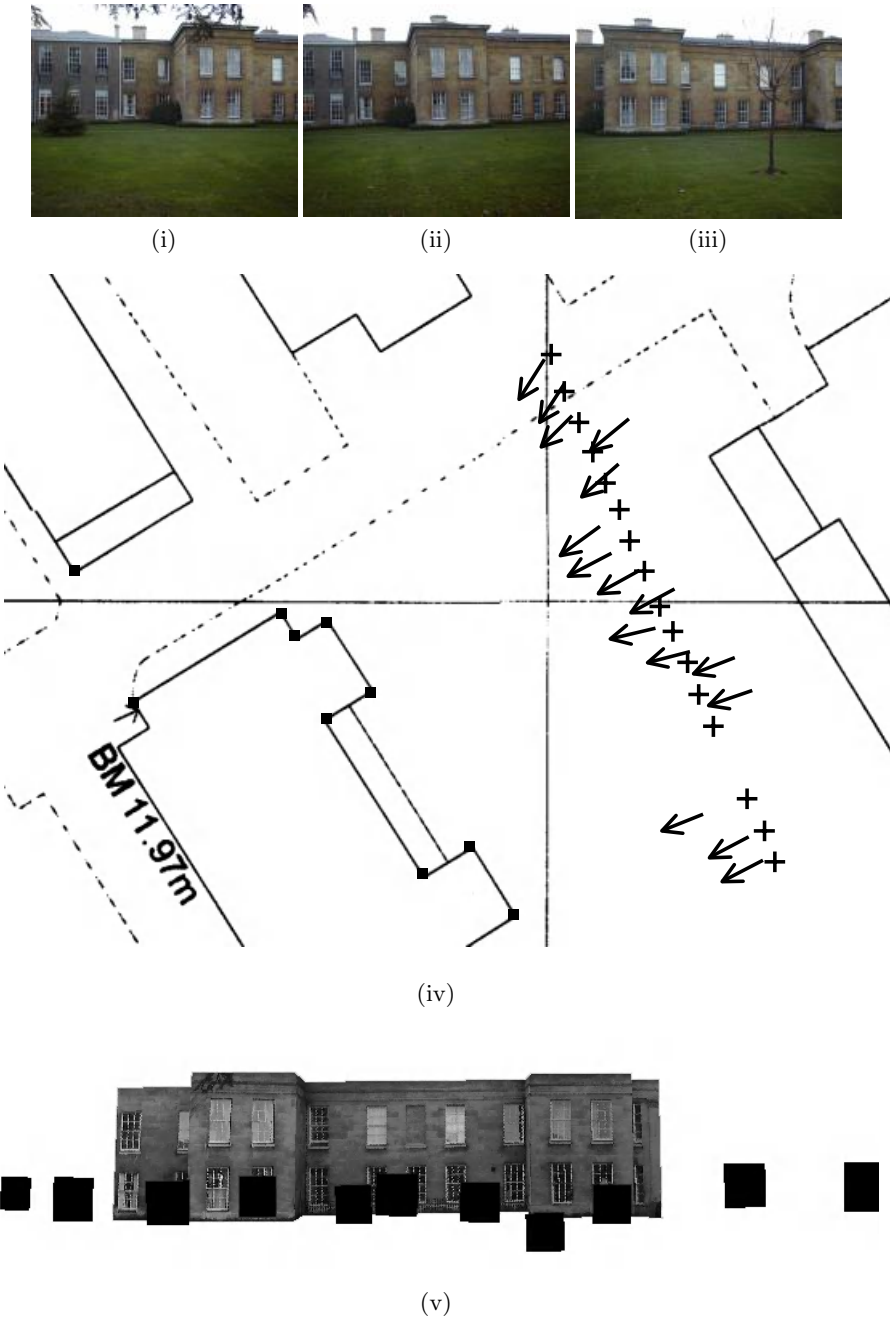
Using the approach described in this paper, we have been able quickly to create large-scale models of a variety of city scenes. Figure 4, for example, shows a view of part of a model of Downing College (before optimisation). Note that since projection matrices are obtained as well as structure, we can also reconstruct features that are not visible on the map.

Compared with existing interactive modelling strategies, the use of the map as an affine view means that more accurate models can be produced more quickly using fewer feature correspondences. Because all camera positions and scene structure are determined simultaneously as the solution of a linear equation, failure of the algorithm is far less common and time-consuming than in systems relying on a sequential DLT plus bundle adjustment approach (like that in [1]). Where the sequential approach fails it is necessary to repeat multiple time-consuming bundle adjustment steps.

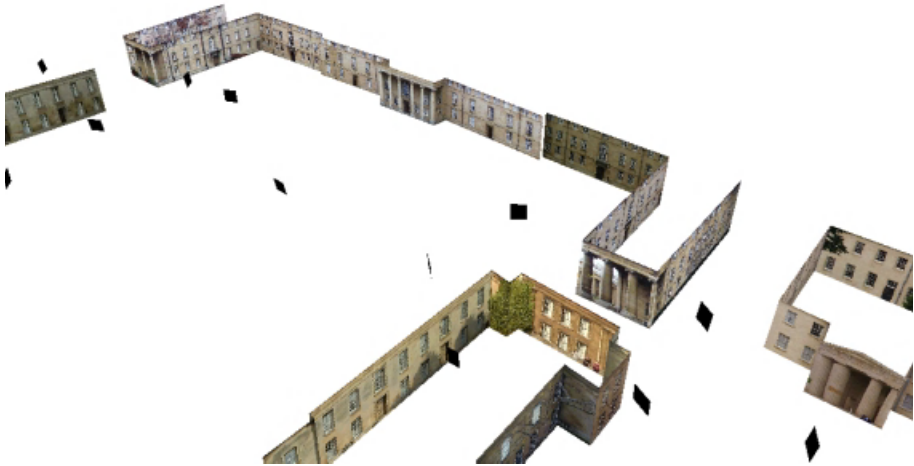
## 5.3 Plane Rectification

Many indoor scenes have planar ceilings and floors, and walls arranged at right angles to each other. This allows us to use the plane rectification technique in [12] to obtain a plan view from photographs of a floor or ceiling such as Figure 5(i). The ceiling plan in Figure 5(ii) is obtained (up to scale) from three rectified photographs without any knowledge of camera focal length. This plan is used like a map view in order to obtain the reconstruction in Figure 5(iii, iv) by the approach described in this paper.

In general modelling indoor scenes is extremely difficult without calibrated wide-angle (or panoramic) cameras. This difficulty arises because only narrow baseline views can be obtained in cramped indoor conditions and degenerate points are common (so that sequential camera registration approaches often fail). By first obtaining a ceiling (or floor) plan we are able to model indoor scenes with ease using a camera with unknown and varying focal length. Our results are comparable with those produced using calibrated panoramas in [17].



**Fig. 3.** (i, ii, iii) Three representative images from a sequence of 16. (iv)  $X, Y$  component of recovered camera positions using the sequential approach compared with ground truth. (v) A synthesised view showing camera image planes at consistent height.



**Fig. 4.** Part of a large model of Downing College obtained using the approach presented in this paper (before optimisation). This model was reconstructed from 30 photographs obtained from sparse viewpoints using a camera with unknown and varying focal length and a readily available 1:500 scale Ordnance Survey map. Projection matrices were obtained as well as structure allowing reconstruction of points not shown on the map (camera image planes are shown in black).

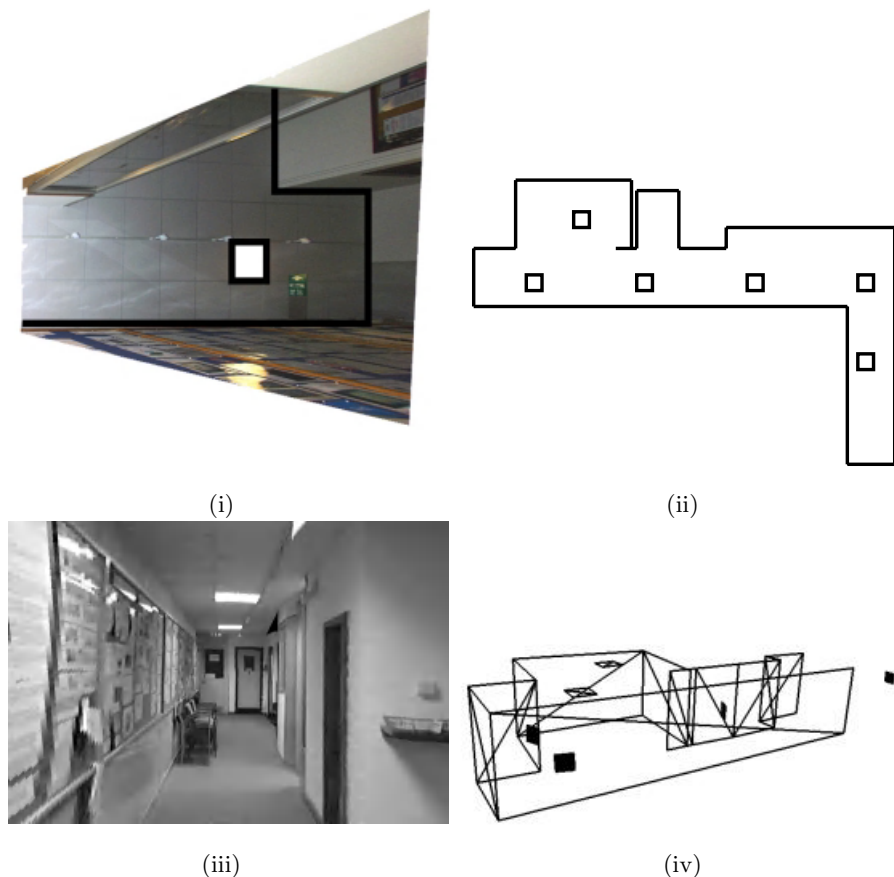
## 6 Conclusion

We have developed a practical system for making large-scale architectural models from many uncalibrated images. By using a map along with prior knowledge of which lines are vertical, we have shown that the *uncalibrated* structure from motion problem can be formulated as a simple *linear* equation. In the context of an interactive system, this reconstruction approach succeeds where the sequential approach fails and does not rely on overly restrictive assumptions about the scene.

Although map information may be *locally* much less accurate than image data (e.g. on the scale of a single building), it does provide a strong constraint on *absolute* geometry. Thus, map information may be used in combination with image data interactively to build much larger models than can be obtained using images alone. This approach makes possible to build models of whole city streets rather than simply a few buildings. An additional benefit is that models are registered in an absolute (map) coordinate system.

The principal limitation of all interactive approaches to model building is the amount of time required of the user. However, a number of techniques may be used automatically to improve coarse models produced interactively (e.g. voxel





**Fig. 5.** (i) One of three rectified photographs of the ceiling from which a complete map (ii) was assembled. The map allows recovery of structure and camera position as the solution of a linear equation. (iii) A synthesised novel view. (iv) Recovered structure and camera positions (before optimisation).

carving [15] and template-based model fitting [7]). Critical to the success of these techniques is a good initial guess at camera registration and structure. Present work concerns supplementing fast interactive modelling techniques with automatic ones.

## References

1. P. Beardsley, P. Torr, and A. Zisserman. "3D Model Acquisition from Extended Image Sequences." In *Proc. 4th European Conference on Computer Vision*, Cambridge (April 1996); LNCS 1065, volume II, pages 683-695, Springer-Verlag, 1996.

2. P. Beardsley, A. Zisserman, and D. Murray. "Sequential Updating of Projective and Affine Structure from Motion." *International Journal of Computer Vision* (23), No. 3, Jun-Jul 1997, pages 235-259.
3. D. Bondyfalat, T. Papadopoulos, B. Mourrain. "Using Scene Constraints during the Calibration Procedure." In Proc. *ICCV'01*, volume II, pages 124-130.
4. B. Caprile and V. Torre. "Using vanishing points for camera calibration." *International Journal of Computer Vision*, 4(2):127-140, March 1990.
5. R. Cipolla, T. Drummond and D. Robertson. "Camera calibration from vanishing points in images of architectural scenes." In Proc. *British Machine Vision Conf.*, volume II, pages 382-392, 1999.
6. P. E. Debevec, C.J. Taylor, and J. Malik. "Modelling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach." In *A CM Computer Graphics (Proceedings SIGGRAPH)*, pages 11-20, 1996.
7. A. Dick, P. Torr, S. Ruffe, R. Cipolla. "Combining Single View Recognition and Multiple View Stereo for Architectural Scenes." In /em Proc. 8th IEEE International Conference on Computer Vision (ICCV'01), pages 268-274, July 2001.
8. A. W. Fitzgibbon and A. Zisserman. "Automatic Camera Recovery for Closed or Open Image Sequences." In Proc. *ECCV*, 1998.
9. R. I. Hartley. "Euclidean reconstruction from uncalibrated views." In J.L. Mundy, A. Zisserman, and D. Forsythe, editors, *Applications of Invariance in Computer Vision*, volume 825 of Lecture notes in Computer Science, pages 237-256, Springer-Verlag, 1994.
10. R. I. Hartley and P. Sturm. "Triangulation." In *American Image Understanding Workshop*, pages 957-966, 1994.
11. R. I. Hartley. "In defence of the 8-point algorithm." In Proc. *International Conference on Computer Vision*, pages 1064-1070, 1995.
12. D. Liebowitz and A. Zisserman. "Combining Scene and Auto-calibration Constraints." In Proc. *ICCV*, volume I, pages 293-300, 1999.
13. N. Navab, Y. Genc, and M. Appel. "Lines in one orthographic and two perspective views." In Proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, Hilton Head Island, South Carolina, Vol. 2, 607-616, June 2000.
14. M. Pollefeys, R. Koch, and L. Van Gool. "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters." In Proc. *6th Int. Conf. on Computer Vision*, Mumbai, India, 1998.
15. K. N. Kutulakos and S.M. Seitz. "A theory of shape by space carving." In *Seventh International Conference on Computer Vision (ICCV'99)*, Greece, September 1999.
16. C. Rother and S. Carlsson: "Linear Multi View Reconstruction and Camera Recovery." In Proc. *8th International Conference on Computer Vision (ICCV'01)*, July 2001, Vancouver, Canada, pp. 42-49
17. H-Y. Shum, M. Han, and R. Szeliski. "Interactive Construction of 3D Models from Panoramic Mosaics." In Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, pages 427-433, Santa Barbara, (June) 1998.
18. C. Slama. "Manual of Photogrammetry." American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.
19. P. F. Sturm and S. J. Maybank. "A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images." In Proc. *British Machine Vision Conference*, volume I, pages 265-274, 1999.
20. I. E. Sutherland. "Three dimensional data input by tablet." Proc. *IEEE*, Vol 62, No. 4:453-461, April 1974.

21. M. Antone and S. Teller. "Automatic Recovery of Relative Camera Rotations for Urban Scenes." Proc. *CVPR 2000*, Volume II, June 2000, pp. 282-289.
22. Z. Zhang, P. Anandan, and H.-Y. Shum. "What can be determined from a full and a weak perspective image?" *International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, pages 680-687, September 1999.

# Motion – Stereo Integration for Depth Estimation

Christoph Strecha<sup>1</sup> and Luc Van Gool<sup>1</sup>

KU Leuven ESAT/PSI,  
B-3001 Leuven, Belgium  
{Christoph.Strecha, Luc.VanGool}@esat.kuleuven.ac.be  
<http://www.esat.kuleuven.ac.be/psi/visics>

**Abstract.** Depth extraction with a mobile stereo system is described. The stereo setup is precalibrated, but the system extracts its own motion. Emphasis lies on the integration of the motion and stereo cues. It is guided by the relative confidence that the system has in these cues. This weighing is fine-grained in that it is determined for every pixel at every iteration. Reliable information spreads fast at the expense of less reliable data, both in terms of spatial communication and in terms of exchange between cues. The resulting system can handle large displacements, depth discontinuities and occlusions. Experimental results corroborate the viability of the approach.

## 1 Introduction

Stereo and shape-from-motion are among the dominant methods for 3D shape reconstruction. Both have been studied extensively in computer vision, but mostly separately. Nevertheless, the two methods have much in common and an integration could follow rather naturally while bringing added value, such as less sensitivity to occlusions and more robust convergence.

Similar as in the case of separate stereo or motion analysis, integrated approaches have used discrete features (e.g. points [1,2] or line segments [3]) or dense correspondences (e.g. based on correlation windows [4], spatio-temporal image gradients [5] or MRFs [6]). The method proposed here belongs to the latter category, as our extension to the PDE approach for optic flow by Proesmans *et al.* [7] yields dense correspondences. This approach was an interesting point of departure as it can handle large disparities and as it detects occlusions and flow discontinuities. ‘Large disparities’ may also be relatively small in absolute terms in that traditional optic flow methods tend to fail as soon as motions are large compared to the granularity of scene texture. Our integrated approach reinforces these advantages further, in contrast to earlier integrated approaches which either neglect the detection of occlusions or are limited to small motion displacements.

In our approach the occlusion detection is in fact part of a correspondence quality estimation scheme, that determines the relative influences of the stereo and motion cues. For instance, in cases where one but not the other suffers from

occlusion, the one least affected will take the upper hand. But this weighing scheme reaches farther than depth discontinuity and occlusion detection, in that it guides the relative influences of both cues at every iteration and at every pixel during the evolution towards the solution. This sets our approach apart from earlier motion-stereo integration work.

Our method assumes a calibrated stereo rig (both the relative position of the cameras and their internal parameters are known beforehand), that is moved around with an unknown motion in a static environment. This motion is determined by the system. The parameters of the stereo rig are assumed to remain known during the motion. In our experiments, they have been kept fixed. If needed, one can do away with the rig calibration by using self-calibration methods for mobile stereo rigs [8]. The paper describes the basic integration procedure, where two subsequent stereo views are taken, i.e. four images in total. The extension to a whole stereo video can easily be made.

Of course, the combined use of multiple views for 3D reconstruction is not new. Bundle adjustment is a well-established technique to achieve exactly that. It improves both the 3D reconstruction and the camera calibration by exploiting the data provided by multiple cameras. However, whereas bundle adjustment basically takes the given feature correspondences for granted and tries to explain away visible deviations through adaptations to both camera parameters and 3D structure, the proposed approach is geared towards updating and coupling the correspondences themselves based on all the image data. It therefore plays a complementary role and acts at an earlier stage of the 3D reconstruction process.

The paper is organized as follows. Section 2 introduces some notations and explain the calibration of the stereo rig. Section 3 describes preparatory steps for the search of corresponding points within the single cues. Emphasis is on the introduction of a depth related parameterisation that facilitates the combination of information over multiple images and between cues. Section 4 discusses the integration of stereo and motion into a single scheme to extract depth and rig motion. Section 5 gives experimental results. Finally, section 6 concludes the paper.

## 2 The Stereo Setup

Our experimental setting consists of two video cameras mounted on a stereo rig at a distance of approximately 0.2 m. We choose the left camera center to be the Euclidean coordinate center. A 3D point denoted by  $\mathbf{X} = (X, Y, Z, 1)^T$  is projected to left image coordinates  $\mathbf{x}_l = (x_l, y_l, 1)^T$  and right image coordinates  $\mathbf{x}_r = (x_r, y_r, 1)^T$  through<sup>1</sup> :

$$\lambda_l \mathbf{x}_l = \mathbf{K}_l [\mathbf{I} | 0] \mathbf{X} \quad (1)$$

$$\lambda_r \mathbf{x}_r = \mathbf{K}_r [\mathbf{R}^T | -\mathbf{R}^T \mathbf{t}] \mathbf{X} \quad (2)$$

---

<sup>1</sup> In the following we will use the vector sign to describe non-homogeneous pixel coordinates  $\mathbf{x}_{l,r} = (x_{l,r}, y_{l,r})$

where  $\mathbf{K}_l$  and  $\mathbf{K}_r$  denote the left and right camera matrices,  $\mathbf{R}$  is the  $3 \times 3$  rotation matrix specifying the relative orientation of the cameras and  $\mathbf{t} = (t_x, t_y, t_z)^T$  is the translation vector between the two cameras. The camera calibration matrices  $\mathbf{K}$  are described by:

$$\mathbf{K} = \begin{pmatrix} f & s & x_0 \\ 0 & af & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where  $f$  denotes the focal length,  $(x_0, y_0)$  is the principal point,  $s$  is related to the pixel skew and  $a$  is the aspect ratio. When dealing with real cameras the above projection equation (1) will be perturbed by radial distortion of the lens. This can be modeled by the following equation [9]:

$$\begin{aligned} \mathbf{x}_u &= \mathbf{x}_d + \hat{\mathbf{x}}_d(k_1\|\hat{\mathbf{x}}_d\|^2 + k_2\|\hat{\mathbf{x}}_d\|^4) \\ \hat{\mathbf{x}}_d &= \mathbf{x}_d - \mathbf{c} \end{aligned} \quad (4)$$

where  $\mathbf{x}_u$  is the unmeasurable undisturbed 2D point on the image plane and  $\mathbf{x}_d$  the measured distorted point. We allow in this model the center of distortion  $\mathbf{c}$  to be different from the image center. In our experiments the radial distortion was estimated and corrected ( $\mathbf{x} = \mathbf{x}_u$ ).

Calibrating our system consists of extracting the internal camera parameters for each camera and the geometric relation between the two cameras. For the calibration we used a calibration box with 140 circles with known relative 3D coordinates. The center of each circle was computed by an ellipse fit to the Canny edge map. Assuming ellipses and using their centers is reasonable, since the circles are small compared to the image size (so that the ellipse centers correspond well to the projected circle centers and the radial distortion can be neglected). We used a stereo pair of the calibration box as well as one image of the box for each camera separately, with the box completely filling the whole image in the latter case. The last two images were added to get a better result for the radial distortion. For these calibration images we have 42 unknown parameters: the rotation and translation of the calibration box with respect to the first camera and the rotation + translation between the two cameras (12), the rotations and translations of the box when taking the separate, per camera images (12), 4  $\lambda$  parameters, one calibration matrix per camera (10), and finally 2 radial distortion parameters per camera (4). We solve this nonlinear system by a Levenberg-Marquardt minimization.

### 3 Single Cue Correspondence Search

PDE based methods have been shown to give good results for stereo and optical flow correspondence search [10,7,11,12,13]. They are in general not dependent on preprocessing stages (feature point or line segment extraction) and provide directly a dense correspondence map. We use the PDE based approach proposed

by Proesmans *et al.* [7]. They propose a system of 6 coupled, non-linear diffusion equations that in effect yield not only the disparities but also discontinuity maps indicating depth discontinuities as well as parts visible to only one of the cameras (occlusions). The high number of equations is due to the symmetric exploitation of the two images: the system embarks on a simultaneous 1st-to-2nd and 2nd-to-1st image correspondence search (in the sequel referred to as forward and backward schemes). Another feature is the ‘bi-local’ nature of the differential computations, i.e. spatial and temporal derivatives at two different positions in the two images are combined. This has to do with the division of the disparity or motion displacement into a current estimate plus a residue, which gradually declines during subsequent iterations. The current estimate yields an offset between the points at which derivatives are taken in the two images. Working with residues allows to better linearize the problem for large disparities, an argument with the assumptions underlying the optical flow constraint equation. A similar strategy is followed here. In that respect our approach differs from others [5,6].

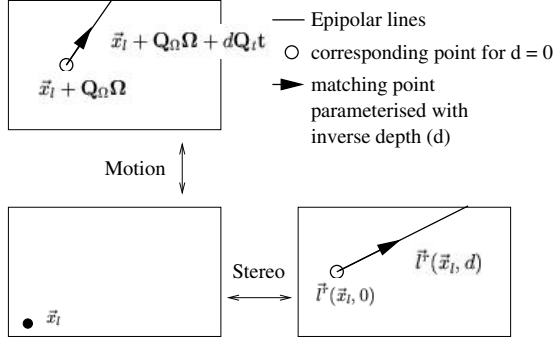
PDE methods often are expensive in terms of computation time, and may get stuck in some local minimum. Both problems can be reduced. Weickert *et al.* [14] used semi-implicit discretisation schemes to lower the computational effort. Multi-resolution techniques can help to find the global optimum [15,14]. The change to a semi-implicit and multi-resolution implementation of the original equations by Proesmans *et al.* [7] is a first modification to their system that we propose. A second is the use of epipolar geometry. In the original system epipolar geometry was not used. It is only along these lines that correspondence search proceeds, reducing the number of PDE’s for correspondence search between single image pairs from 6 to 4.

The choice of a common parameter value for all corresponding points in the four images, where the parameterisation runs along the epipolar lines and is directly related to depth, facilitates our integration of stereo and motion.

**Depth parameterisation for stereo related views.** We assume that the translation and rotation between the stereo cameras is known from precalibration. From eqs. (1, 2) the epipolar lines for the two cameras can be derived. It follows for corresponding image points  $\mathbf{x}_l = (x_l, y_l, 1)^T$  and  $\mathbf{x}_r = (x_r, y_r, 1)^T$ :

$$\frac{\lambda_r}{Z} \mathbf{x}_r = \mathbf{K}_r \tilde{\mathbf{R}} \mathbf{K}_l^{-1} \mathbf{x}_l + \frac{\mathbf{K}_r \tilde{\mathbf{t}}}{Z}, \quad (5)$$

with  $\tilde{\mathbf{t}} = -\mathbf{R}^T \mathbf{t}$  and  $\tilde{\mathbf{R}} = \mathbf{R}^T$ . The stereo correspondence is divided into a component that depends on the rotation and pixel coordinate (according to the homography  $\mathbf{H} = \mathbf{K}_r \tilde{\mathbf{R}} \mathbf{K}_l^{-1}$ ) and a depth dependent part that scales with the amount of translation between the cameras. For the left image the parameterisation of the corresponding point along the epipolar line in the right image is realized by  $l^r(\mathbf{x}_l, d)$  starting from the point  $Z = \infty$  and going in the direction of the epipole  $\mathbf{K}_r \tilde{\mathbf{t}}$  (see fig. 1). Points along the epipolar lines will be parameterised



**Fig. 1.** Parameterisation of the correspondence for a point  $\mathbf{x}_l$  in the left image at the first time instance for stereo and motion related views

with  $d = \frac{1}{Z}$ , i.e. a *depth related parameter*. In pixel coordinates this gives:

$$\mathbf{l}^r(\mathbf{x}_l, d) = \frac{\begin{pmatrix} \mathbf{H}[1]\mathbf{x}_l \\ \mathbf{H}[2]\mathbf{x}_l \end{pmatrix} + d \begin{pmatrix} \mathbf{K}_r[1]\tilde{\mathbf{t}} \\ \mathbf{K}_r[2]\tilde{\mathbf{t}} \end{pmatrix}}{\mathbf{H}[3]\mathbf{x}_l + d t_z} \quad (6)$$

$\mathbf{H}[i]$  is the 3-vector for the  $i$  th row of the homography  $\mathbf{H}$  and similarly for  $\mathbf{K}_r[i]$ . This parameterisation differs from the one advocated by Alvarez *et al.* [12], which is less directly coupled to depth.

**Depth parameterisation for motion related views.** To make use of the epipolar constraint that is also present in motion related views (we assume a static environment) the rig motion has to be estimated. For that purpose we have extended the instantaneous motion model described in [16,17] to the general case of non-square and skewed pixels. Similar calculations provide us with the 6 motion parameters  $\mathbf{p} = (\boldsymbol{\Omega}, \mathbf{t})^T = (\Omega_x, \Omega_y, \Omega_z, t_x, t_y, t_z)^T$  where the first 3 represent the rotation angles about the corresponding axes and the last 3 the translation components along these axes. The extraction of these motion parameters from motion correspondences is discussed in the next paragraph. The displacement for each pixel can then be expressed as a function of the depth related parameter  $d$  and the motion parameters:

$$\mathbf{u} = \mathbf{Q}_\Omega \boldsymbol{\Omega} + d \mathbf{Q}_t \mathbf{t} = \mathbf{Q} \mathbf{p} \quad (7)$$

$$\mathbf{Q}_\Omega = \begin{pmatrix} \frac{\tilde{x}\tilde{y}}{af} + s & -f - \frac{\tilde{x}^2}{f} + \frac{s\tilde{x}\tilde{y}}{af^2} & \frac{\tilde{y}}{a} - \frac{s\tilde{x}}{f} + \frac{s^2\tilde{y}}{af^2} \\ af + \frac{\tilde{y}^2}{af} & -\frac{\tilde{x}\tilde{y}}{f} + \frac{s\tilde{y}^2}{af^2} & -\tilde{x} + \frac{s\tilde{y}}{f} \end{pmatrix}, \mathbf{Q}_t = \begin{pmatrix} -f & -s & \tilde{x} \\ 0 & -af & \tilde{y} \end{pmatrix} \quad (8)$$

where  $\tilde{x} = x - x_0$  and  $\tilde{y} = y - y_0$  are the centered image coordinates. Fig. 1 shows this parameterisation, which is very similar to that of stereo. In the following we describe the motion correspondence by  $\mathbf{m}(\mathbf{x}, d)$ . For a pixel  $\mathbf{x}_l \in I_1^l$  the corresponding pixel in image  $I_2^l$  (see fig. 1,2) parameterised by  $d$  is:

$$\mathbf{m}_2^l(\mathbf{x}_l, d) = \mathbf{x}_l + \mathbf{Q}_\Omega \boldsymbol{\Omega} + d \mathbf{Q}_t \mathbf{t} \quad (9)$$



**Camera motion estimation.** Following eq. (7), the motion displacements (correspondences) of the pixels and the motion parameters  $\mathbf{p}$  can be extracted by an iterative process. Suppose we already have an estimate of the displacement vector  $\mathbf{u}_0$ . In the same vein as the offset + residual description of optical flow by Proesmans *et al.* [7], we split the displacement in a current estimate and a residual  $\mathbf{u}_r = \mathbf{u} - \mathbf{u}_0$ . The introduction of  $\mathbf{u}_0$  is especially important for large displacements, for which we could not otherwise truncate the  $O(\mathbf{u}^2)$  terms from the Taylor expansion of  $\mathbf{u}$ . Assuming Lambertian surfaces and hence identical intensities for corresponding pixels, we have

$$I_1(\mathbf{x}) = I_2(\mathbf{x} + \mathbf{u}_0 + \mathbf{u}_r) = I_2(\mathbf{x} + \mathbf{u}_0) + \frac{\partial I_2(\mathbf{x} + \mathbf{u}_0)}{\partial \mathbf{x}} \mathbf{u}_r \quad (10)$$

Setting  $\mathbf{u}_r = \mathbf{Q}\mathbf{p}_r = \mathbf{Q}\mathbf{p} - \mathbf{u}_0$  and assuming that  $\mathbf{Q}$  is known (it is because the depths at each pixel ( $d_0$ ) obtain a value at each iteration in our final system), we look for those  $\mathbf{p}$  that minimize the integral:

$$\int_{\mathbf{x} \in \Omega} \left( I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}_0) - \frac{\partial I_2(\mathbf{x} + \mathbf{u}_0)}{\partial \mathbf{x}} (\mathbf{Q}\mathbf{p} - \mathbf{u}_0) \right)^2 d\mathbf{x}$$

This yields

$$\begin{aligned} \mathbf{A}\mathbf{p} &= \mathbf{b} \\ \mathbf{A} &= \sum_{\mathbf{x} \in \Omega} \mathbf{Q}^T I_{2x}^T I_{2x} \mathbf{Q}, \quad \mathbf{b} = \sum_{\mathbf{x} \in \Omega} \mathbf{Q}^T I_{2x}^T (I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}_0) + I_{2x} \mathbf{u}_0) \end{aligned} \quad (11)$$

where

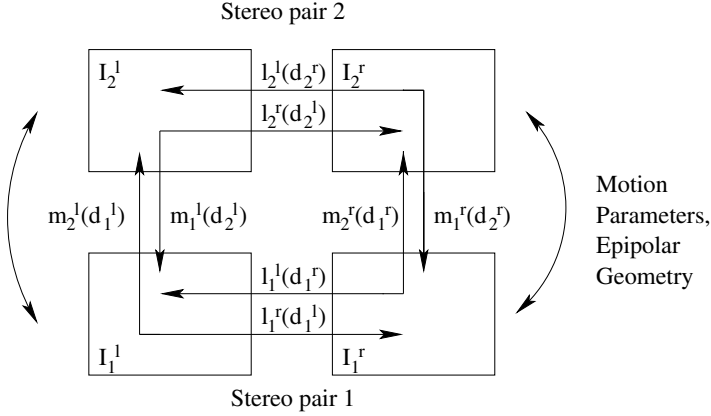
$$I_{2x} = \frac{\partial I_2(\mathbf{x} + \mathbf{u}_0)}{\partial \mathbf{x}} \quad (12)$$

and where we sum over some image domain  $\Omega$ . This will later include all pixels with sufficiently good confidence scores for their correspondences.

## 4 Integration of Motion and Stereo

Our integration of stereo and motion takes the form of a single system of equations. This system yields dense scene depths, their discontinuities and occlusions, and the motion of the stereo rig. The evolving functions for inverse depth and motion are initialized to zero.

For the given image configuration (see figure 2) pixels in an image can typically obtain a depth through both stereo and motion based reconstruction. Exceptions are those that fall victim to occlusions in one or both of these cues. These two depths ought to be the same, and hence corresponding pixels in the two other images are expected to obtain equal epipolar parameter values ( $d$  for stereo =  $d$  for motion). Armed with these expected equalities (see fig 2) and exploiting the stereo and the motion companion views, the inverse depth  $d$  is



**Fig. 2.** Stereo disparity relations:  $l(d)$ ,  $m(d)$  are the stereo and motion correspondences eq. 6 and 9

solved through the iteration of a system of coupled, non-linear diffusion equations, à la Proesmans *et al.* [7]. As a matter of fact, such a system is solved for each of the four images, but their systems are coupled. In the case of image  $I_1^l$  the system takes the following form (at each pixel):

$$\begin{aligned}
 \frac{\partial d_1^l}{\partial t} &= \text{div}(\delta(c^{s,m}) \nabla d_1^l) \\
 &\quad - \frac{\gamma(c^s)}{\gamma(c^s) + \gamma(c^m)} \lambda I_{\mathbf{x}}^s (I_{\mathbf{x}}^s (d_1^l - d_{10}^l) + I_t^s) \\
 &\quad - \frac{\gamma(c^m)}{\gamma(c^s) + \gamma(c^m)} \lambda I_{\mathbf{x}}^m (I_{\mathbf{x}}^m (d_1^l - d_{10}^l) + I_t^m) \\
 \frac{\partial c^s}{\partial t} &= \rho \nabla^2 c^s + 2\alpha(1 - c^s) |C^s| \\
 \frac{\partial c^m}{\partial t} &= \rho \nabla^2 c^m + 2\alpha(1 - c^m) |C^m|
 \end{aligned} \tag{13}$$

As mentioned, these are variations on Proesmans *et al.*'s [7] equations and the reader is referred to that reference for a detailed description. The superscripts  $()^{m,s}$  are related to the motion or stereo pair, resp. The expressions  $I_t^m$ ,  $I_t^s$  replace the temporal derivative of intensity in the traditional optical flow constraint, an adaptation due to the formulation in terms of residual motions,  $I_{\mathbf{x}}^m$ ,  $I_{\mathbf{x}}^s$  are spatial derivatives of intensity, taken here along the epipolar lines. The definitions for  $\mathbf{x}_l \in I_1^l$  are:

$$\begin{aligned}
 I_t^m &= I_1^l(\mathbf{x}_l) - I_2^l(\mathbf{m}_2^l(\mathbf{x}_l, d_{10}^l)), I_t^s = I_1^l(\mathbf{x}_l) - I_1^r(I_1^r(\mathbf{x}_l, d_{10}^l)) \\
 I_{\mathbf{x}}^m &= \frac{\partial I_2^l(\mathbf{m}_2^l(\mathbf{x}_l, d_{10}^l))}{\partial d}, I_{\mathbf{x}}^s = \frac{\partial I_1^r(I_1^r(\mathbf{x}_l, d_{10}^l))}{\partial d},
 \end{aligned} \tag{14}$$



**Fig. 3.** Two stereo pairs (left and right images) at the first (bottom ( $I_1^l, I_1^r$ )) and the second (top ( $I_2^l, I_2^r$ )) time instance

where  $\mathbf{l}_1^l(\mathbf{x}_l, d_{10}^l)$  and  $\mathbf{m}_2^l(\mathbf{x}_l, d_{10}^l)$  are the positions of the corresponding points for stereo and motion using the previous estimate  $d_{10}^l$  of the parameter value. Every pixel in image  $I_1^l$  also has confidence measures  $c^s$  and  $c^m$  for its stereo and motion correspondences, resp.. Values close to 0 mean high confidence, values close to 1 low confidence.

The first equation governs the evolution of the depth related parameter  $d_1^l$ . The first term is an anisotropic diffusion term. It blocks diffusion (smoothing) from places with a lower confidence in their correspondences. In a typical iteration process most places start out with low confidences, but at the end low confidence values ( $c^s$  close to 1) tend to cluster near discontinuities and occlusions. More explanation is given later. The second and third terms impose optical flow constraints on the stereo and motion correspondences, resp. Similarly as with Proesmans *et al.*'s system, the formulation is in terms of residual displacements rather than complete displacements. The rationale is as said: the correspondence search becomes more amenable to linearization. The two terms are weighted with factors. The weight of the second term increases with the confidence in the stereo correspondence of that pixel, giving more importance to the depth ( $1/d$ ) suggested by the stereo cue. The third term acts similarly, but for the motion cue. The function  $\gamma$  in these weighting factors is given by:

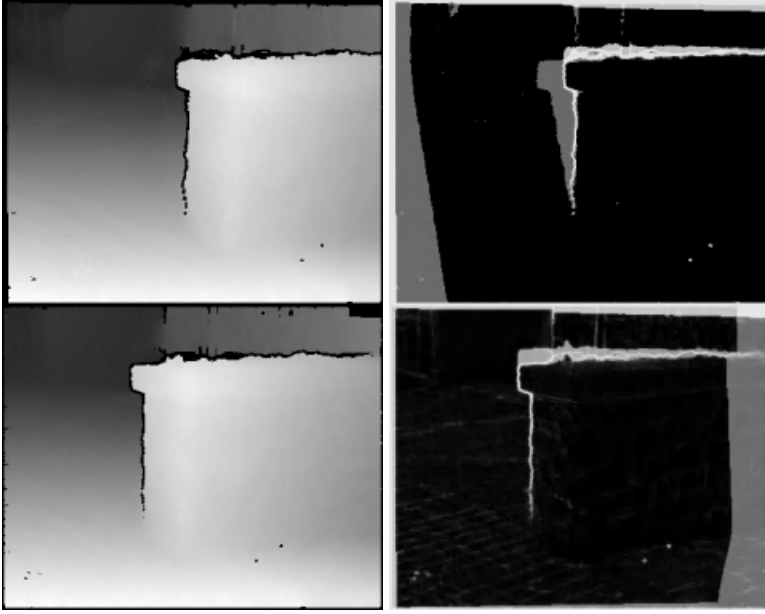
$$\gamma(c) = \exp\left(-\frac{c^2}{k}\right). \quad (15)$$

They are high for high confidences (i.e. low  $c^{m,s}$ ). The relative confidences put in the stereo vs. the motion correspondence guide the relative influence that these cues get in the correspondence search. These relative values can change over time (over iterations) and from one pixel to the next.

The evolution of the confidence measures  $c^{m,s}$  in the second and third equations is driven by the vectors  $C^m$  and  $C^s$ . They measure the difference between forward and backward flow in the stereo and motion direction, respectively. In the normal case of a pair of corresponding points, the extracted displacements for the first with respect to the second – forward flow – and the second with respect to the first – backward flow – are equal but of opposite sign. Hence, when summed the two displacements will cancel each other out. The  $C$ 's represent these vector sums. Large  $|C|$  yield values of  $c$  that are close to one – its maximal value – whereas at other places  $c$  tends to zero. Hence,  $c$  actually quantifies the inconsistency between the forward and backward flows. Its restriction to the interval  $c \in [0, 1]$  is realized by the factor  $(1 - c)$ . The interested reader may note that we actually simplified the 2nd and 3rd equation with respect to those proposed by Proesmans *et al.* [7] without loss of performance. A separate inconsistency measure  $c^m$  and  $c^s$  is extracted for the motion and the stereo pair, resp. The use of this forward-backward regularization scheme is - different from usual optical flow techniques - important here. Since it is not only used to block smoothing but also to weight stereo and motion according to it's ability to extract depth especially near occlusion. However, other regularizers could be added with small contribution.

Although inconsistency values are calculated for all pixels separately, their surroundings matter as the 2nd and 3rd equations also contain a diffusion term. Referring to the first equation, its anisotropic diffusion coefficient  $\delta(c^{m,s})$  is controlled by the inconsistencies  $(c^m, c^s)$ . The role of this coefficient is to prevent smoothing across depth edges or incorrect displacements being spread towards neighbouring pixels. Since the corresponding pixel can be occluded in one neighboring image but not in the other, information exchange for the image pair with good correspondences should persist. This is realized by taking the maximum of  $\gamma(c^m)$  and  $\gamma(c^s)$ . Subsequently, these maximal values are normalized as to sum to one over the 4 neighbors of a pixel and the result is called  $\delta(c^{s,m})$ . The reason behind this normalization is to make sure that diffusion does not stop completely in all directions simultaneously, as this can keep the system from evolving towards the solution at the early stages.

The overall system of equations that is solved, consists of the forementioned 3 equations for each of the four images. Additional to this an update of the motion parameters  $\mathbf{p}_{l,r}$  for the left and right camera is done after each iteration. For the calculation we combine forward and backward motion since they are related by  $(\mathbf{R}^{12}, \mathbf{t}^{12}) = (\mathbf{R}^{21^T}, -\mathbf{R}^{21^T} \mathbf{t}^{21})$ . This is reflected by the following relation



**Fig. 4.** Left: Depth map (black pixels on the edge of the little wall indicate sharp discontinuities); Right: Inconsistency, high confidence pixels in black, low confidence pixels in white (discontinuities). Dark gray corresponds to low confidence regions for stereo only (occlusions), light gray for motion only

between the instantaneous motion vectors:

$$\begin{pmatrix} \Omega_x^{12} \\ \Omega_y^{12} \\ \Omega_z^{12} \\ t_x^{12} \\ t_y^{12} \\ t_z^{12} \end{pmatrix} = \mathbf{G} \begin{pmatrix} \Omega_x^{21} \\ \Omega_y^{21} \\ \Omega_z^{21} \\ t_x^{21} \\ t_y^{21} \\ t_z^{21} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -\Omega_z^{12} & \Omega_y^{12} \\ 0 & 0 & 0 & \Omega_z^{12} & -1 & -\Omega_x^{12} \\ 0 & 0 & 0 & -\Omega_y^{12} & \Omega_x^{12} & -1 \end{pmatrix}. \quad (16)$$

Using eq. (11), the common motion vector  $\mathbf{p}^{12}$  is estimated from:

$$\begin{bmatrix} \mathbf{A}^{12} \\ \mathbf{A}^{21}\mathbf{G} \end{bmatrix}_{6 \times 12} \mathbf{p}^{12} = \begin{bmatrix} \mathbf{b}^{12} \\ \mathbf{b}^{21} \end{bmatrix}_{1 \times 12} \quad (17)$$

The input to this (remember eq. 12 and eq. 8) contains the current estimate of the inverse depth. As described by eq. 11 we sum over appropriate domains. These contain all pixels with confidence values  $c^s$  and  $c^m$  that are below some threshold (0.15 in our experiments). In other words, only pixels with high confidence for both (motion and stereo) correspondences contribute to the extraction of the rig's motion.

A coupling between the left and right motion parameters is therefor realised by the coupling of the depth values. Additional constraints between left and



**Fig. 5.** Pure stereo results; Left: depth map for the dynamic path algorithm; Right: depth map for left the algorithm described in section 3, black pixels have low confidence (e.g. occlusion).

right camera motions can be imposed in the case of a fixed stereo rig. We intend to relax the condition that the stereo rig has to be fixed, only requiring it to be calibrated at the start and, hence, have not used this additional constraint at this point. The total system consists of 14 equations. First, for the four images the 3 equations (13) yielding the depths and inconsistency measures (discontinuities and occlusions) are iterated one in turn (12 equation). Then, with the resulting update for the depth values, a new estimate of the cameras motion is calculated (2 equations of type 17). This results in new epipolar constraints, that are fed back to the 12 equations for the next iteration. As mentioned we used Weickert's [14] semi-implicit discretisation for the sake of fast convergence.

## 5 Experimental Results

We tested the method on real images taken by two Sony DCR-TRV900E cameras with an image size of  $(360 \times 288)$ . We discovered a serious radial distortion and corrected for it in a preprocessing stage. The inverse depths and the motion vector ( $d$  and  $\mathbf{p}$ ) were initialized with zero and estimated in a coarse-to-fine manner over 6 pyramid levels. Figure 3 shows the 4 input images.

For the sake of comparison, fig 5 gives the results for the determination of depth based solely on the initial stereo pair. The image on the left is the depth map obtained from correspondences obtained with our dynamic path search algorithm [18]. The middle and right images show the results of the forward and backward schemes of our modified Proesmans *et al.* [7] algorithm. It yields better results than the other method, but still shows some gaps and part of the depth discontinuities remain undetected.

The results for the left and right cameras of the integrated system are shown in figure 4. One can see on the left the inverse depth maps ( $d_1^l$  (top),  $d_1^r$  (bottom)) for the images  $I_1^l$  and  $I_1^r$  and on the right the corresponding inconsistency maps  $c^s$ ,  $c^m$  for these images (confidences, with bright meaning low and dark high). Comparing these results with the results from single stereo (see figure 5), we can

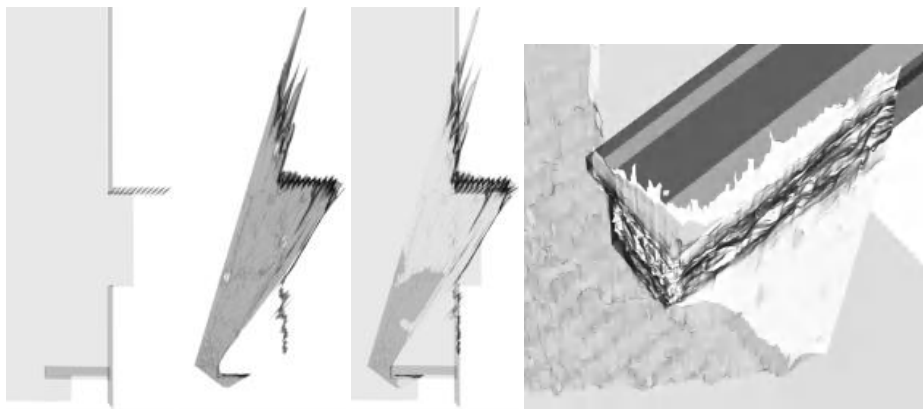


**Fig. 6.** 3D model from image  $I_1^l$ ; left: model right with added texture

conclude that the system is better able to assign good depth values to pixels that have no correspondence in the stereo partner image. The left part of the scene and the occluding parts behind the wall got their depth from the upward motion of the left camera. The occlusions to the motion / stereo partner are marked (light gray / dark gray) in the inconsistency maps of fig. 4. The same can be seen in the depth map for the right image  $I_1^r$ : the occluded part on the right of fig. 3 got its depths from the motion system. In conclusion, depth is recovered of more points than in the single stereo case. This is not surprising as we used more images. Nevertheless, this result shows the success of this motion-stereo integration scheme, as a system that is able to decide whether stereo or motion cues should be used for 3D estimation.

A second observation is that the integrated system yields better defined depth discontinuities (bright values in the inconsistency map on the right). Moreover, where disparities get large, pure stereo (fig. 5) had problems with local minima. This can be seen at the lower left part of the image, where the recurrent stone texture contains a lot of possibilities for wrong matches, and also near the top right part of the wall, with hardly any texture in the direction of the epipolar lines (nearly horizontal). These problems are alleviated by the integrated approach since the epipolar lines for both cues are typically not parallel and therefore contain complementary information. Pixels in white in fig. 4 are those where we have discontinuities with respect to both (stereo and motion) correspondences. These points form the anisotropic diffusion coefficient  $\delta(c^{m,s})$ .

Fig. 6 and 7 show views of the 3D model. In the last figures we added the simplified ground truth for the scene (measured with a ruler). The depth discontinuities in the 3D model correspond to the discontinuities  $\delta(c^{m,s})$ . The discontinuities are an explicit part of the output and are not extracted *post factum* through edge detection in the depth map, as e.g. in [5]. They are of a better quality that way, certainly in scenes like this one with several planes almost orthogonal to the image planes. All these planes yield high gradients in the depth map and tend to yield many spurious ‘discontinuities’ through depth edge detection. The 3D reconstruction is quite precise. For instance, the stone structures of the small wall close to the cameras come out well. This precision points at



**Fig. 7.** Views of the 3D model from image  $I_1^l$ ; ground truth top view, model top view, combined ground truth and data; right: zoom on the wall in front with added ground truth

the subpixel accuracy of the correspondences. For this stereo rig at a distance of 2m a match  $\pm$  one pixel corresponds to a depth change of  $\pm$  2.5cm. We did not retrieve such detail when all correspondences were retrieved with our dynamic path search algorithm [18]. The whole scene also lines up well with the ground truth. However, there are still some mistakes in the reconstruction. Some pixels at the back wall appear to be more to the front than they should. These pixels didn't have a correspondence in the stereo partner. Another mistake is the connection between the wall on the right and the back wall. There the motion inconsistency measure  $c^m$  was not able to find the discontinuity (see also fig. 4). The result was obtained after 10 minutes (image size  $360 \times 288$ , PC 700 MHz).

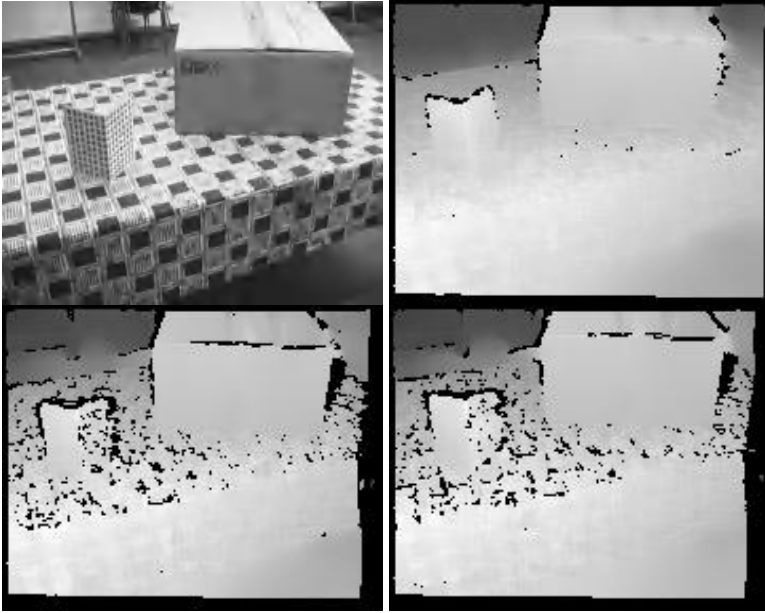
In fig. 8 we show the result for an indoor scene. By comparing the result with the result from single stereo in fig. 8(bottom) for both time instances one can see that it would not be possible to achieve the result of the combined approach by combining the two depth maps from single stereo. Especially the part behind the table is wrong in both single stereo depth maps. This said, in this case also the combined approach could not prevent that some errors occur. The depth discontinuity between the table cloth and the floor remains largely undetected. This is due to the fundamental problem that the floor is essentially untextured.

The computation time for this example was less than 4 minutes. This time again included the computation on all pyramid levels and the extraction of the full depth, inconsistencies and motion parameters for all 4 images up to image size  $160 \times 120$ .

## 6 Summary and Conclusions

We have proposed a scheme to integrate depth extraction from stereo and motion. A precalibrated stereo rig was moved with an unknown motion. The point





**Fig. 8.** Top right: Depth map with the proposed approach; Bottom: Depth maps for the indoor scene from single stereo at first (left) and second (right) time instance

of departure of our integration was a PDE scheme for the extraction of correspondence between pairs of images, as introduced by Proesmans *et al.* [7]. It has the advantage that it detects discontinuities in the disparities or motion fields as well as occlusions. It can also deal with large displacements and it yields good precision.

We have modified this scheme in a number of important ways. First, we have shortened the processing time substantially by using Weickert *et al.*'s [14] semi-implicit discretisation, by restricting the search along epipolar lines, and by applying a multi-resolution approach. This reduces the time from hours to minutes. The latter alteration also resulted in higher robustness, e.g. against ambiguities in the case of periodic textures. Secondly, we have adapted this scheme for the integration of motion and stereo. On the one hand, we have introduced a very direct coupling that was guided by the dynamic, relative weighing of both schemes at every pixel and at every iteration. This allows our method to really get the best from both cues. On the other hand, we have used inverse depth as a common parameter for all correspondences, which facilitated the direct combination of matches coming from the different cues and which directly yields the valuable depth of points.

We plan to extend this work in a number of directions. Stereo videos rather than two subsequent stereo pairs will be processed. Static scenes will be replaced by scenes with independently moving objects. It is there that having stereo vision really pays off, as the scene is always static as far as the stereo image pairs are

concerned (one moment in time). The stereo rig will be made variable, so that the vergence and focal lengths can be changed. Finally, we plan to improve the correspondence search under variable lighting conditions and in the presence of specular reflections.

**Acknowledgment.** The authors gratefully acknowledge support by K.U.Leuven GOA project ‘VHS+’ and EU IST project ‘CogViSys’.

## References

- [1] F.Dornaika and R.Chung: Stereo correspondence from motion correspondence. *CVPR*, vol. 1, pp. 70–75, 1999.
- [2] P.K.Ho and R.Chung: Stereo-motion with stereo and motion in complement? *PAMI*, vol. 22, no. 2, pp. 215–220, 2000.
- [3] Z.Zhang, Q.T.Luong, and O.Faugeras: Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Trans. Robotics and Automation*, vol. 12, no. 1, pp. 103–113, feb 1996.
- [4] R.Mandelbaum, G.Salgian, and H.Sawhney: Correlationbased estimation of ego-motion and structure from motion and stereo. *ICCV*, pp. 544–550, 1999.
- [5] G.P.Stein and A.Shashua: Direct estimation of motion and extended scene structure from a moving stereo rig. *CVPR*, pp. 211–218, 1998.
- [6] G.Sudhir, S.Banerjee, R.Bahl, and K.Biswas: A cooperative integration of stereopsis and optic flow computation. *J. Opt. Soc. Am. A*, vol. 12, pp. 2564, 1995.
- [7] M.Proesmans, L.Van Gool, E.Pauwels, and A.Oosterlinck: Determination of optical flow and its discontinuities using non-linear diffusion. *ECCV*, vol. 2, pp. 295–304, 1994.
- [8] A.Zisserman, P.A.Beardsley, and I.D.Reid: Metric calibration of a stereo rig. *In Proc. IEEE Workshop on Representation of Visual Scenes*, pp. 93–100, 1995.
- [9] G.P.Stein: Lens distortion calibration using point correspondences. *CVPR*, pp. 143–148, 1997.
- [10] J.Shah: A nonlinear diffusion model for discontinuous disparity and half-occlusions in stereo. *CVPR*, pp. 34–40, 1993.
- [11] M.Proesmans, E.Pauwels, and L.Van Gool: *in Geometry-driven diffusion in computer vision*, ed. Bart M. ter Haar Romeny Kluwer, 1994.
- [12] L.Alvarez, R.Deriche, J.Sanchez, and J.Weickert: Dense disparity map estimation respecting image discontinuities: pde and scalespace based approach. Tech. Rep. RR-3874, INRIA, 2000.
- [13] J.Weickert and C.Schnörr: Variational optic flow computation with a spatio-temporal smoothness constraint. Tech. Rep. 15, University of Mannheim, 2000.
- [14] J.Weickert, B.M.ter Haar Romeny, and M.A.Viergever: Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 398–410, 1998.
- [15] D.Marr and T.Poggio: A theory of human stereopsis. *Proc. Royal Soc. B*, vol. 204, pp. 301–328, 1979.
- [16] D.J.Heeger and A.D.Jepson: Subspace methods for recovering rigid motion I: algorithm and implemenation. *IJCV*, vol. 7, no. 2, pp. 95–117, 1992.
- [17] M.Irani: Multi-frame optical flow estimation using subspace constraints. *ICCV*, pp. 626–633, 1999.

- [18] G.Van Meerbergen, M.Vergauwen, M.Pollefeys, and L.Van Gool: A hierarchical stereo algorithm using dynamic programming. *IEEE Workshop on Stereo and Multi-Baseline Vision*, pp. 166–174, 2001.

# Lens Distortion Recovery for Accurate Sequential Structure and Motion Recovery

Kurt Cornelis\*, Marc Pollefeys\*, and Luc Van Gool

K. U. Leuven, ESAT-PSI

Kasteelpark Arenberg 10

B-3001 Leuven-Heverlee, Belgium

{kurt.cornelis, marc.pollefeys, luc.gool}@esat.kuleuven.ac.be

<http://www.esat.kuleuven.ac.be/~kcorneli>

<http://www.esat.kuleuven.ac.be/~pollefeys>

**Abstract.** Lens distortions in off-the-shelf or wide-angle cameras block the road to high accuracy Structure and Motion Recovery (SMR) from video sequences. Neglecting lens distortions introduces a systematic error buildup which causes recovered structure and motion to bend and inhibits turntable or other loop sequences to close perfectly. Locking back onto previously reconstructed structure can become impossible due to the large drift caused by the error buildup. Bundle adjustments are widely used to perform an ultimate post-minimization of the total reprojection error. However, the initial recovered structure and motion needs to be close to optimal to avoid local minima. We found that bundle adjustments cannot remedy the error buildup caused by ignoring lens distortions. The classical approach to distortion removal involves a preliminary distortion estimation using a calibration pattern, known geometric properties of perspective projections or only 2D feature correspondences. Often the distortion is assumed constant during camera usage and removed from the images before applying SMR algorithms. However, lens distortions can change by zooming, focusing and temperature variations. Moreover, when only the video sequence is available preliminary calibration is often not an option. This paper addresses all fore-mentioned problems by sequentially recovering lens distortions together with structure and motion from video sequences without tedious pre-calibrations and allowing lens distortions to change over time. The devised algorithms are fairly simple as they only use linear least squares techniques. The unprocessed video sequence forms the only input and no severe restrictions are placed on viewed scene geometry. Therefore, the accurate recovery of structure and motion is fully automated and widely applicable. The experiments demonstrate the necessity of modeling lens distortions to achieve high accuracy in recovered structure and motion.

**Keywords:** Structure from motion, calibration, lens distortion recovery, high accuracy, sequential.

---

\* Kurt Cornelis and Marc Pollefeys are respectively research assistant and postdoctoral fellow of the Fund for Scientific Research - Flanders(Belgium)(F.W.O. - Vlaanderen)

# 1 Introduction

## 1.1 Previous Work

Much research exists that acknowledges the importance of modeling lens distortions. Most papers determine lens distortions using calibration patterns [2,3,7,13,14,15], known geometric properties of perspective projections [2,4,8,9,11,12] or 2D feature correspondences [5,9,10,17]. After this pre-calibration, the distortions are often assumed constant during the remainder of the camera usage, a valid assumption if no zooming or focusing is performed. However, when only the video sequence is available a preliminary calibration is often impossible and other ways to recover lens distortions are needed.

Lens distortions are mostly considered after the application of an ideal pinhole projection model. Some work also exists to include distortions implicitly in projection equations using intermediate parameters without physical meaning [14]. The extraction of the distortion parameters from the latter, e.g. to undo or add the same distortion to computer generated graphics which need to be incorporated in Augmented Reality, is often difficult and not well conditioned. This is due to the strong coupling between intrinsic, extrinsic and distortion parameters. As several authors [2,9,10,15] stated, this coupling can result in unacceptable variance of the recovered parameters. Therefore, a decoupling of the projection equations in a part modeling ideal pinhole projection and a part modeling lens distortions is used by [4,15].

This paper is most closely related to [15] but presents a new way to sequentially determine lens distortions together with structure and motion from a video sequence without preliminary calibration using calibration patterns or specific geometric scene properties. Due to this sequential nature, the lens distortions are also allowed to change over time as can happen in reality.

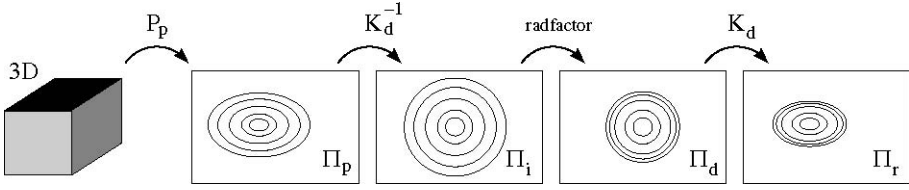
## 1.2 Overview

A first section will describe the camera projection model. Subsequently, the estimation of all parameters given 3D-2D correspondences is explained. The model consists of a pinhole projection part and a lens distortion part. The estimation of both are realized iteratively, each described in a separate section. This estimation needs initialization which is considered in a following section. Experiments demonstrate the importance of modeling distortions, we finish with a short summary and propose future research topics.

# 2 The Camera Model

The model describing the projection process from 3D scene points to 2D image coordinates consists of several sequential steps, shown in Figure 1. First, a projection takes place according to the classical pinhole model.

$$m_p \sim \mathbf{P}_p M_r \quad (1)$$



**Fig. 1.** The camera model from left to right: First, the 3D space is projected onto an image plane  $\Pi_p$  with a pinhole camera. Next, this plane is transformed with  $\mathbf{K}_d^{-1}$  to yield the ideal image plane  $\Pi_i$ . In the ideal image plane the radial distortion factor is applied to give coordinates in the distorted image plane  $\Pi_d$ . Finally, this plane is transformed back using  $\mathbf{K}_d$  to end up in the real image plane  $\Pi_r$ .

$$\mathbf{P}_p = \mathbf{K}_p [\mathbf{R}^T | -\mathbf{R}^T t]$$

$$\mathbf{K}_p = \begin{bmatrix} f_p & s_p & u_p \\ 0 & r_p f_p & v_p \\ 0 & 0 & 1 \end{bmatrix}$$

in which  $M_r = (X_r, Y_r, Z_r, W_r)^T$  and  $m_p = (x_p, y_p, w_p)^T$  are the homogeneous coordinates of the 3D point and its projection onto the pinhole image plane  $\Pi_p$ .  $\mathbf{P}_p$  is a  $3 \times 4$  matrix and ‘ $\sim$ ’ denotes that (1) is valid up to a scale factor.  $\mathbf{P}_p$  can be decomposed in internal and external calibration parameters.  $\mathbf{K}_p$  is called the calibration matrix which contains the focal length  $f_p$ , the pixel aspect ratio  $r_p$ , the skew  $s_p$  and the principal point  $(u_p, v_p)$ .  $\mathbf{R}$  and  $t$  determine the rotation and translation in world coordinates.

Next, the distortions transforming the ideal pinhole image into an image that conforms more to real images are modeled. Various distortions exist [16] but radial distortion, described by the following model, is the most prominent.

1. Using a distortion calibration matrix similar to  $\mathbf{K}_p$  transform the pinhole image coordinates  $m_p$  to the ideal image plane  $\Pi_i$  where the radial distortion center equals  $(0, 0)$ .

$$m_i = \mathbf{K}_d^{-1} m_p \tag{2}$$

$$\mathbf{K}_d = \begin{bmatrix} f_d & s_d & u_d \\ 0 & r_d f_d & v_d \\ 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{K}_d$  can differ from  $\mathbf{K}_p$  as cameras can be recovered in a projective framework in which  $\mathbf{K}_p$  has no physical meaning. While radial distortions take place around the physical optical axis, the principal point  $(u_p, v_p)$  cannot be used as the distortion center.  $\mathbf{K}_d$  is therefore considered the Euclidean version of  $\mathbf{K}_p$ . As in modern cameras the skew is negligible and aspect ratio is practically 1, we can fix  $s_d$  and  $r_d$  to 0 and 1 respectively while keeping the distortion center  $(u_d, v_d)$  and focal length  $f_d$  variable. As shown in [4, 9], the latter allows to model de-centering distortions together with radial

distortions. Note, however, that all parameters could be made variable if desired.

2. Next, a radial distortion is applied to  $m_i$  to yield the distorted image coordinates  $m_d$  in the distorted image plane  $\Pi_d$ .

$$\begin{aligned} m_d &= m_i \times \text{radfactor} \\ \text{radfactor} &= \sum_{j=0}^n k_j r_i^{2j} \\ \text{with } r_i &= \sqrt{x_i^2 + y_i^2} \end{aligned} \quad (3)$$

where  $k_j$  are the radial parameters. The model for *radfactor* is common in literature except that we take  $k_0$  not to equal 1 but make it act as a parameter for a reason later explained. A decreasing *radfactor* for increasing  $r_i$  introduces barrel distortion; otherwise it introduces pincushion distortion.

3. Finally, the distorted coordinates  $m_d$  are transformed with  $\mathbf{K}_d$  to yield the final real image coordinates  $m_r$  in the real image plane  $\Pi_r$ .

$$m_r = \mathbf{K}_d m_d \quad (4)$$

An ambiguity exists between the radial parameters  $k_j$  and  $\mathbf{K}_d$ . Scaling  $f_d$  and  $s_d$  with  $sc$  results in scaling the ideal image plane  $\Pi_i$  around the distortion center with  $sc^{-1}$ . Therefore, *radfactor* has to remain unchanged to obtain identical  $m_r$  after distortion. In equation (3) only  $r_i$  is scaled, but an appropriate change of  $k_j$  can compensate this, hence the ambiguity. To resolve the ambiguity  $f_d$  and  $s_d$  will be scaled such that every  $m_i$  will lie in the range  $[-1, 1]$  in  $\Pi_i$ . This will condition algorithms for numerical stability.

Above we stated that strong parameter coupling can result in unacceptable variance of the recovered parameters. This has to be clarified as an ambiguity is a perfect coupling. Suppose two parameters describe an ambiguity, as in  $a \times b = c$  where  $c$  is a constant an increase in  $a$  can be compensated by a decrease in  $b$ . The camera model estimation, explained in the following sections, consists of separate steps in which some parameters are held constant at each step. Therefore, fixing one parameter during a step automatically determines the value of the other parameter with which it forms the ambiguity, avoiding any danger of large parameter variances.

### 3 Camera Model Estimation

During sequential Structure and Motion Recovery from video sequences we advance each time by calculating the camera pose for the current frame given 3D-2D correspondences  $(M_r, m_r)$ , as explained in [1]. Given the 3D-2D correspondences, we wish to minimize the following error:

$$\mathbf{P}_{p, \mathbf{K}_d, k_j} \min \sum_l (m_r^l - \text{proj}(M_r^l))^2$$

in which  $proj()$  denotes the total projection, described in section 2, and  $\mathbf{P}_p, \mathbf{K}_d$  and  $k_j$  are the parameters to be optimized. The error is the residual reprojection error in the real image which forms the natural goal for minimization as this is the only error visible to human observers.

The camera model consists of two parts; a first part models a pinhole projection; a second part describes lens distortions. We therefore opted to perform the error minimization as an iterative process where each part is minimized while the other is kept constant. As stated above, the coupling between all parameters is strong if solved for in a single global optimization. The decoupling in this multi-step iterative procedure reduces this problem and allows to use simple linear least squares techniques. The iterative procedure has the following lay-out:

1. Initialize the distortion parameters  $\mathbf{K}_d$  and  $k_j$ .
2. Given the distortion parameters and  $m_r$ , one can compute  $m_p$ . In Figure 1 this corresponds to a motion from right to left towards the plane  $\Pi_p$ .
3. Given  $(M_r, m_p)$  correspondences estimate  $\mathbf{P}_p$  minimizing the residual reprojection error.
4. Given the projection matrix  $\mathbf{P}_p$  and  $M_r$ , one can compute  $m_p$ . In Figure 1 this corresponds to a motion from left to right towards the plane  $\Pi_p$ .
5. Given  $(m_p, m_r)$  correspondences determine  $\mathbf{K}_d$  and  $k_j$  minimizing the residual reprojection error.
6. Return to step 2 until convergence.

The initialization step 1 will be explained in section 4. The following sections clarify step 5 and step 3 respectively.

### 3.1 Lens Distortion Estimation

At this point a previous best pinhole camera matrix  $\mathbf{P}_p$  has been determined. Given  $(m_p, m_r)$  correspondences, we look for the distortion parameters  $\mathbf{K}_d$  and  $k_j$  minimizing the following residual reprojection error:

$$\min_{\mathbf{K}_d, k_j} \sum_l (m_r^l - distort(m_p^l))^2 \quad (5)$$

in which  $distort()$  determines the second part (lens distortions) of the camera model. The residual is expressed in the real image plane  $\Pi_r$ , the only plane in which we can finally see the errors as all other image planes  $\Pi_p, \Pi_i$  and  $\Pi_d$  are virtual. Substituting equations (2) and (3) in (4) we get nonlinear equations in the distortion parameters. However, given constant  $m_i$ , we note that fixing  $\mathbf{K}_d$  gives linear equations in  $k_j$ . Vice versa, taking  $k_j$  constant yields linear equations in the elements of  $\mathbf{K}_d$ . Therefore, another iterative solution surfaces:

1. Use the current  $\mathbf{K}_d$  and  $\mathbf{P}_p$  to form a compound camera matrix which projects a 3D point  $M_r$  directly onto the ideal image plane  $\Pi_i$ :  $\mathbf{P}_i = \mathbf{K}_d^{-1} \mathbf{P}_p$ . Use  $\mathbf{P}_i$  to project all  $M_r$  to their corresponding  $m_i$  which from now on are assumed constant. Note that assuming  $m_i$  constant is equal to fixing  $\mathbf{P}_i$ . In the following steps  $\mathbf{K}_d$  will change, requiring  $\mathbf{P}_p$  to compensate for this change to keep  $\mathbf{P}_i$  constant. The altered  $\mathbf{P}_p$  will be determined in step 6.



2. Given  $\mathbf{K}_d$  and equations (3) and (4) we can determine  $k_j$  minimizing (5) with linear least squares techniques.
3. Using  $k_j$  and equation (3) we can distort all  $m_i$  to their corresponding  $m_d$  in the distorted image plane  $\Pi_d$ .
4. Given  $m_d$  and equation (4) we can determine  $\mathbf{K}_d$  minimizing (5) with linear least squares techniques.
5. Return to step 2 until convergence.
6. Because  $m_i$  and therefore  $\mathbf{P}_i$  were assumed constant and  $\mathbf{K}_d$  changed during iteration, we update  $\mathbf{P}_p$  by extracting the new  $\mathbf{K}_d$  from  $\mathbf{P}_i$ :  $\mathbf{P}_p = \mathbf{K}_d \mathbf{P}_i$

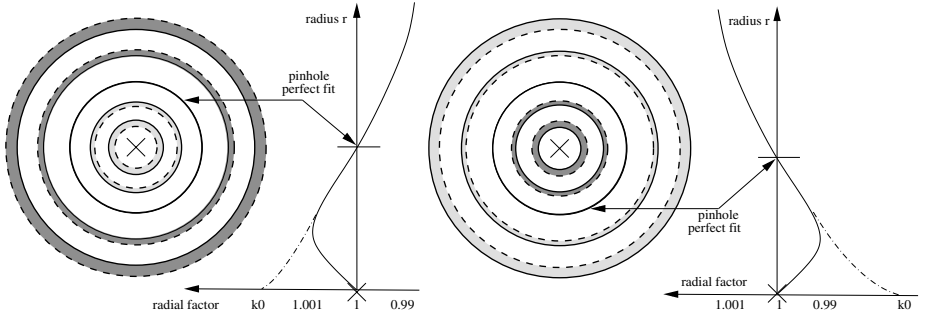
The use of parameter  $k_0$ , in conventional radial distortion modeling taken to be 1, will now be explained. The camera matrix  $\mathbf{P}_p$  used in step 1 will have been estimated before the lens distortions are updated in step 2–5. Therefore,  $\mathbf{P}_p$  will have been estimated based on an image where radial distortion can be under- or overestimated. This leads to a  $\mathbf{P}_p$  predicting well the 3D-2D correspondences ( $M_r, m_r$ ) on a certain circle around the distortion center in  $\Pi_r$ . But for barrel lens distortions it will overestimate the predicted positions of  $m_r$  outside this circle and underestimate them inside it and vice versa for pincushion distortions, as shown in Figure 2. These over- and underestimations will be compensated for by the estimation of the distortion parameters in steps 2–5. However, note that the estimated *radfactor* should almost equal 1 on the circle for which  $\mathbf{P}_p$  already fits best. By fixing  $k_0$  to 1 we also demand *radfactor* to equal 1 at the distortion center. As Figure 2 shows, this constrains *radfactor* (3) to be a non-monotone function. However, we know that radial distortions in real lenses have a more or less monotone function. Therefore, it is better to allow  $k_0$  to be a parameter, representing a scaling of the pinhole image plane  $\Pi_p$  and the camera matrix  $\mathbf{P}_p$  around the distortion center to achieve a monotone *radfactor* function as shown in Figure 3.

### 3.2 Pinhole Camera Estimation

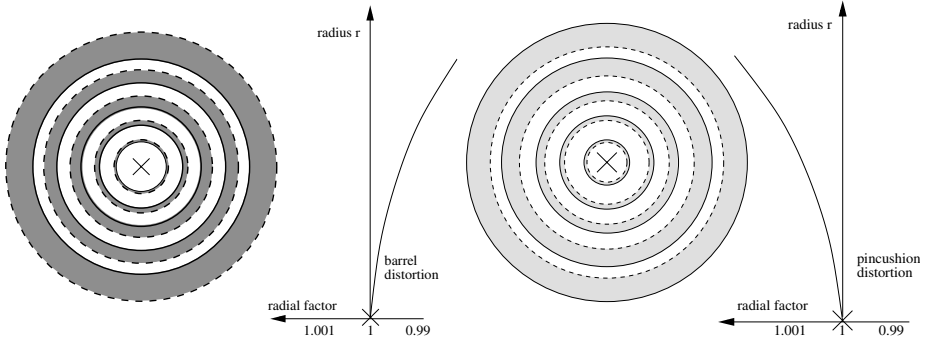
In this section the distortion parameters  $\mathbf{K}_d$  and  $k_j$  are kept constant and the estimation of camera matrix  $\mathbf{P}_p$  using 3D-2D correspondences ( $M_r, m_p$ ) is considered. Given  $m_r$  and the estimated distortion parameters we can undo the distortion to calculate their corresponding  $m_p$ . The camera matrix  $\mathbf{P}_p$  which minimizes

$$\min_{\mathbf{P}_p} \sum_l (m_p^l - \text{pinhole}(M_r^l))^2 \quad (6)$$

can be found by iteratively re-weighted least squares minimization [6] given the previously estimated camera matrix  $\mathbf{P}_p$  as initialization. *pinhole()* represents the pinhole projection model, described in (1), and the residual error (6) is an error living in the pinhole image plane  $\Pi_p$ . Since the only image plane visible to human observers is the real image plane  $\Pi_r$ , the residuals should be transferred to the latter as also noted by [12]. Equation (3) shows that in the neighborhood of  $m_p$  the ideal image plane and the pinhole image plane is scaled with *radfactor*  $\sum_{j=0}^n k_j r_i^{2j}$  to get to the real image plane. Therefore the residual of



**Fig. 2.** Left: If the barrel distortion is underestimated the pinhole camera model fits perfectly on a certain circle (thick) around the distortion center but underestimates the distortion outside this circle (dark gray) and overestimates it inside the circle (light gray). The dotted circles are predicted by the pinhole model, the solid circles are the real distorted ones. The estimated radial factor will try to compensate these over- and underestimations. Right: Same but now for pincushion distortion. In both examples the radial factor is forced to create a bump in its curvature (solid curve) while the best match would be the dash-dotted radial factor curve.



**Fig. 3.** Left: Same situation as in Figure 2 but now the circles predicted by the pinhole model are up-scaled in the image plane  $\Pi_p$  so that the distortion is an overestimation everywhere. This makes it possible for the radial factor to be a monotone function of the radius which is more natural for real lenses. Right: The same but now the dotted circles predicted by the pinhole camera model are down-scaled.

$m_p$  is scaled with exactly this factor to calculate its counterpart in the real image plane. The new error measure then becomes:

$$\min_{\mathbf{P}} \sum_l [\text{radfactor}(m_p^l) * (m_p^l - \text{pinhole}(M_r^l))]^2 \quad (7)$$

## 4 Initialization

The iterative multi-step estimation of all projection parameters needs an initial starting point. Using a sequential Structure and Motion Recovery methodology the projection parameters of each frame are estimated while running through the video. Considering the time-continuity, we can take the previous frame's distortion parameters as a starting point for the current frame. To initialize the sequential recovery of structure, motion and distortion we use the same method as described in [1] where two initial camera matrices  $\mathbf{P}_p$  are determined by decomposing a Fundamental Matrix. This decomposition assumed the Fundamental Matrix to be estimated between two images with zero lens distortion and therefore the initial distortion parameters for these cameras can be taken as:

$$\mathbf{K}_d = \begin{bmatrix} f_d & 0 & u_d \\ 0 & f_d & v_d \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

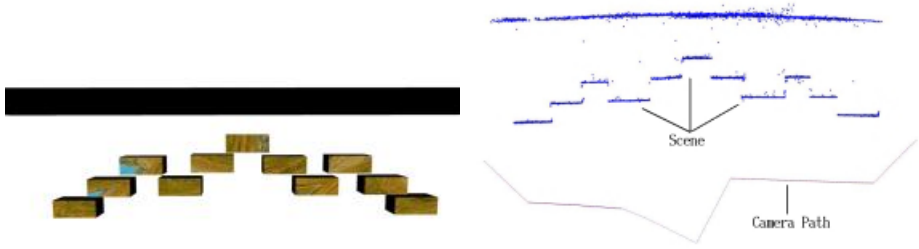
$$k_0 = 1 \text{ and } \forall j \neq 0 : k_j = 0 \quad (9)$$

The initial aspect ratio  $r_d$  and skew  $s_d$  are respectively taken 1 and 0 which is reasonable for real cameras. The initial distortion center  $(u_d, v_d)$  equals the image center and the initial focal length  $f_d$  is chosen such that the image corners form a bounding box in the ideal image plane  $\Pi_i$  which lies in the  $[-1, 1]$  range for conditioning numerical algorithms.

Given a real camera with lens distortions, we therefore start from an initialization which is not correct as both initial cameras are distortion free. However, minimizing the reprojection error for each frame using the supplied distortion parameters to remove any systematic error, the distortion parameters will most often converge to the real ones. At this point we cannot provide any mathematical proof but the statement is backed up with extensive simulations on artificial and real video sequences which show the relevance of modeling lens distortions. When we dispose of the video camera we could use any available off-line technique that uses calibration patterns to find better initial values for the distortion parameters.

## 5 Experimental Results

We conducted experiments on artificial and real-life video sequences to test the usefulness of sequential modeling lens distortions during Structure and Motion Recovery. First, an artificial sequence (500 frames, image resolution  $720 \times 576$ ) without lens distortions was created. The scene consisted of boxes positioned at different depths. Figure 4 shows the recovered Structure and Motion when no radial distortion was estimated. It corresponds very well with the ground truth. Next, we artificially added barrel lens distortions, which moved points in the image corners with 25 pixels from their original position. At first we did not try to recover the distortion which introduced a systematic error buildup in the recovered Structure and Motion. Figure 5 shows how the unmodeled radial distortion bends the recovered structure and motion. When we also tried to recover



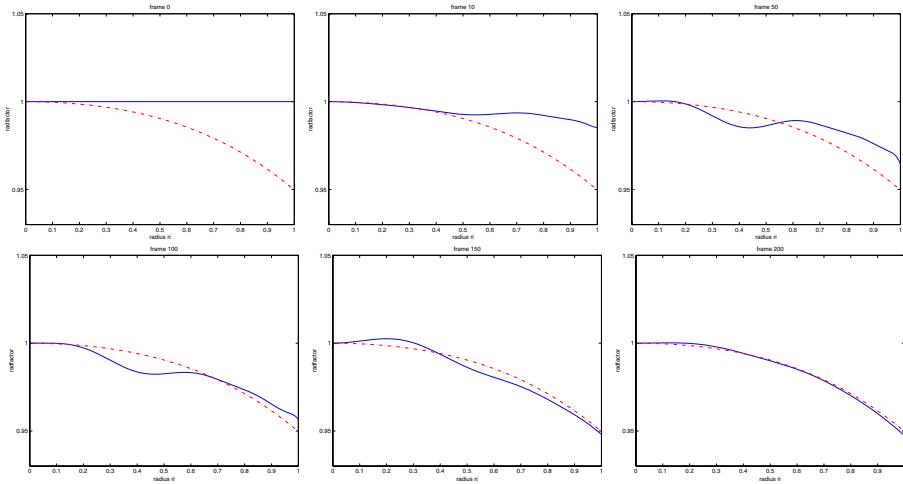
**Fig. 4.** Left: Top view of the scene structure. Several boxes are placed at different depths. Right: The recovered structure (upper dots) and motion (lower line) from an artificial video sequence without radial distortions.



**Fig. 5.** Left: The recovered structure and scene when radial distortion was present but not modeled by the program. Right: The recovered structure and motion when the radial distortion was also modeled as described in this paper. The camera moved from right to left. As the system only converged after 200 frames the part of the structure on the far right still shows some residual bending.

the lens distortions the distortion parameters converged after approximately 200 frames (Figure 6) and the recovered structure and motion resembles more the ground truth as also shown in Figure 5.

Next, a second artificial sequence (image resolution  $720 \times 576$ ) was made to investigate the error buildup with or without modeling lens distortions. It consisted of a turntable sequence of a teapot. Figure 7 shows the first frame of the original and radial distorted versions. A single round trip counts 100 frames. We performed ten round trips and therefore the total sequence consisted of 1000 frames. Due to periodicity the frames whose frame number are equal modulo 100 are the same. The error buildup during sequential Structure and Motion Recovery is measured as the projection errors in cameras that are supposed to be the same due to this periodicity. Equal cameras are supposed to project 3D scene points onto the same image locations. However, this is not the case when error buildup is present which can therefore be measured as the average projection error between ‘equal’ cameras. This projection error, expressed in pixels, is calculated between cameras separated by one round trip (relative error buildup).



**Fig. 6.** From left to right [frames 0,10,50,100,150 and 200]: The convergence of the estimated radial factor (solid curve) towards the ground truth (dash-dot). The abscissa represents the radius in the ideal image plane  $\Pi_i$ . The ordinate represents the radial distortion factor *radfactor*.

Artificial sequences were made having no distortion, barrel distortion and pin-cushion distortion. For each different kind of distortion the influence of the number of estimated radial distortion parameters was investigated (the number of parameters are counted excluding parameter  $k_0$  to correspond to more common conventions). Table 1 displays the results from which several conclusions can be drawn. First, when radial distortion (barrel or pincushion) is present, not modeling it leads to a large error buildup. When modeling the radial distortion it converges except when 1 or 3 parameters are used to model the lens distortions. Next, when no radial distortion was present the results also show that the best error buildup was obtained if no radial distortion was modeled. However, modeling it anyway the results did not completely deteriorate. Another important conclusion is that two radial parameters are often sufficient for convergence as the higher order terms only have a small contribution but using a higher number, e.g. 6, does not necessarily introduce divergence. Actually, we suspect a trade-off to exist. The exact ground truth radial distortion might be sufficiently modeled using only a few parameters. In this framework, however, the radial distortion needs to converge from an initial erroneous estimate to its final correct value. This convergence is a dynamic event in which more degrees of freedom increase the convergence rate. Once converged, a high number of parameters might lead to modeling the residual noise instead of the physical underlying lens distortions. The strange phenomenon that odd number of parameters 1 and 3 do not converge still has to be investigated. Because of the complex interactions that are involved in this sequential structure, motion and lens distortion recovery a theoretical proof of convergence is very difficult. From the results, however, it is clear that not modeling present lens distortions always leads to bad results

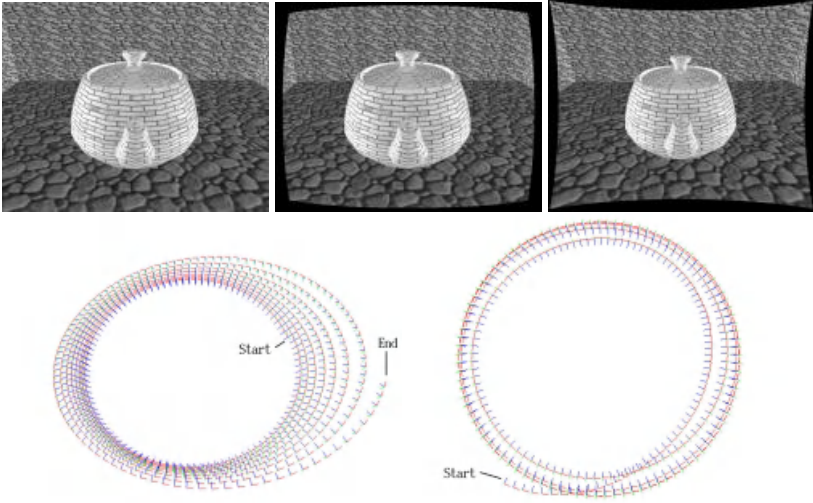
**Table 1.** relative error buildup in pixels: frame 100x(i-1) - frame 100xi Radial calibration matrix  $\mathbf{K}_d$  with  $f_d = 461.025$ ,  $r_d = 1$ ,  $s_d = 0$  and  $(u_d, v_d) = (360, 288)$  on images of resolution  $720 \times 576$ . The number of radial parameters are counted excluding  $k_0$ .

Scenario	rad. param	i = 1	2	3	4	5	6	7	8	9
barrel	0	16.55	16.91	16.87	17.13	17.49	17.47	18.15	18.57	18.62
ground truth	1	12.09	12.87	13.87	14.83	15.66	16.95	18.71	20.76	22.19
	2	6.55	4.05	2.56	0.28	0.08	0.08	0.07	0.07	0.06
$k_0 = 1$ .	3	26.73	46.34	54.46	87.63	130.52	197.76	242.24	79.49	35.18
$k_1 = -0.01$	4	6.86	5.67	5.85	5.42	3.52	3.57	4.67	5.06	7.45
$k_2 = -0.01$	6	3.04	0.90	0.12	0.11	0.08	0.06	0.09	0.07	0.07
$k_3 = -0.01$	8	5.03	4.58	3.79	1.30	0.18	0.08	0.08	0.08	0.08
	0	0.63	0.06	0.05	0.05	0.05	0.04	0.03	0.05	0.05
	1	5.92	8.59	10.52	12.09	13.39	13.06	10.64	10.21	8.93
	2	1.15	0.14	0.07	0.09	0.08	0.06	0.06	0.05	0.05
	3	5.23	11.92	24.28	37.68	36.70	42.63	57.89	61.17	69.63
	4	1.18	0.30	0.08	0.08	0.08	0.06	0.09	0.11	0.09
	6	0.50	0.07	0.07	0.05	0.06	0.06	0.06	0.07	0.07
	8	3.86	1.63	0.08	0.07	0.07	0.08	0.06	0.07	0.06
pincushion	0	17.69	17.02	16.63	15.97	15.73	15.48	15.27	14.85	15.18
ground truth	1	15.92	15.24	14.88	15.29	16.38	19.41	21.24	20.63	19.41
	2	4.46	1.62	0.56	0.08	0.08	0.07	0.07	0.07	0.06
$k_0 = 1$ .	3	13.33	21.10	34.93	45.30	42.28	47.66	51.76	54.16	56.45
$k_1 = 0.01$	4	3.05	0.79	0.10	0.09	0.08	0.08	0.06	0.07	0.07
$k_2 = 0.01$	6	3.30	4.60	2.56	1.03	0.14	0.12	0.09	0.08	0.07
$k_3 = 0.01$	8	5.00	4.88	6.16	6.11	6.30	6.05	6.07	7.18	6.31

while better outcomes can be achieved by modeling the lens distortions. Figure 7 shows the difference in recovered camera positions of the turntable sequence with and without modeling of lens distortions. Without the consideration of distortions a systematic error is incorporated, clearly shown by the diverging camera path. When lens distortions are considered, they are captured and after an initial transient behavior the camera follows a periodic cyclic path as was the case in reality.

Figure 8 shows real-life footage (1200 frames, image resolution  $720 \times 576$ ) representing the roof of an ancient fountain. The reconstruction without modeled lens distortions is shown in Figure 9. Clearly the roof which in reality is straight bends backwards in the reconstruction because of the present barrel distortion. When lens distortions were considered, modeling six radial parameters  $k_j$ , the reconstructions as shown in Figure 10 could be achieved. The first reconstruction in Figure 10 took two frames with zero radial distortion as a starting point. The second reconstruction took the final radial distortion values obtained by the previous run as a starting point and therefore could achieve a structure which closely resembles the real structure as no initial transients were present.

Because the strategy of sequentially modeling lens distortions allows for varying distortions, we tested another artificial video sequence similar to the teapot sequence. A round trip of the turntable sequence consists of 100 frames. We var-



**Fig. 7.** Top: First frame of the original turntable sequence, the barrel distorted version and the pincushion distorted version respectively. Bottom Left: Divergence of the estimated camera path due to lack of radial distortion modeling when a barrel distorted version of the original turntable sequence was used. Bottom Right: Convergence of the camera path when the actual radial distortion is modeled. The camera path does not diverge but converges onto a circular path.

ied the radial distortion from one extreme at frame 0 to another extreme at frame 50, returning to the first extreme at frame 100. Again ten round trips were made and the error buildup assessed as shown in Table 2. Figure 11 shows how the estimated radial distortion moves between the two ground truth extreme radial distortions, proving that variable distortions pose no immediate problems.

**Table 2.** relative error buildup in pixels: frame  $100 \times (i-1)$  - frame  $100 \times i$  radial calibration matrix  $\mathbf{K}_d$  with  $f_d = 461.025$ ,  $r_d = 1$ ,  $s_d = 0$  and  $(u_d, v_d) = (360, 288)$  on images of resolution  $720 \times 576$ . Variation of radial distortion parameters between  $(k_0, k_1, k_2, k_3) = (1., -0.01, -0.01, -0.01)$  and  $(k_0, k_1, k_2, k_3) = (1., -0.03, -0.03, -0.03)$

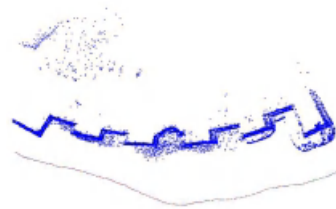
Scenario	rad. param	i = 1	2	3	4	5	6	7	8	9
barrel	6	3.05	0.83	0.08	0.07	0.08	0.08	0.08	0.07	0.08

## 6 Summary

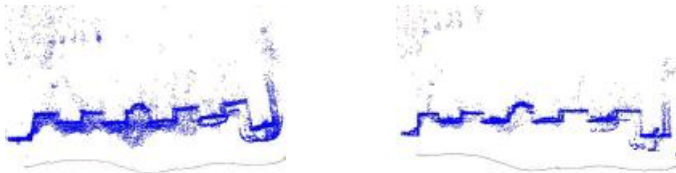
In this paper we described how lens distortions could be recovered sequentially from a video sequence together with structure and motion without the need for preliminary distortion calibration or specific scene geometry. In a first section



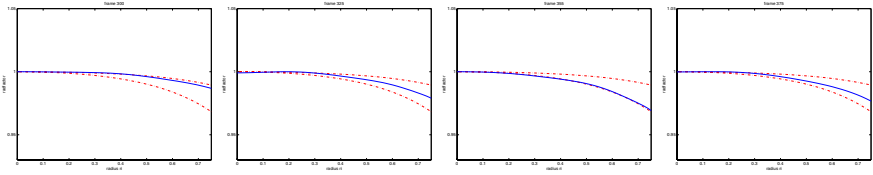
**Fig. 8.** A couple of frames from a video sequence showing an ancient fountain roof. When pasted together, these give an idea of what the real roof looks like.



**Fig. 9.** Top view of the reconstruction of the fountain roof without modeling radial distortions. While the roof is straight in real life the curvature of the scene structure is clearly visible.



**Fig. 10.** Left: Top view of the reconstruction if the radial distortion was modeled with an initialization of 2 frames without radial distortion. Right: Result if the final radial distortion of a first run was taken as initialization for the program.



**Fig. 11.** From left to right [frames 300,325,355,375]: The variation of the estimated radial factor (solid curve) between the two ground truth extremes(dash-dot). The abscissa represents the radius in the ideal image plane  $\Pi_i$ . The ordinate represents the radial distortion factor *radfactor*.

the camera model that approximates the real projection process was discussed. It consisted of two parts, an ideal pinhole camera model and a subsequent distortion model which could model radial and de-centering distortions. In a fol-



lowing section it was shown how the parameters of this camera model could be estimated in a multi-step iterative way given 3D-2D correspondences. The multi-step algorithm consisted of a separate estimation of the ideal pinhole camera model and the distortion parameters. This separation diminished the strong coupling between all the camera parameters which exists in a single global minimization formulation. The following sections explained how the pinhole camera parameters and the distortion parameters could be estimated by applying only linear least squares techniques. Subsequently, as the sequential nature of Structure, Motion and Distortion Recovery needs initialization it was shown that this consisted of 2 cameras with zero distortion or any off-line pre-calibration of the distortion if the video camera was still available. Experiments identified the advantages of taking lens distortions into account by demonstrating results on artificial and real video sequences. It showed that modeling lens distortion was crucial to the minimization of the error buildup. The negligence of lens distortions would introduce a systematic error which causes severe error buildup in recovered scene structure and cameras which could inhibit loop video sequences to close perfectly. Accurate camera retrieval enables the recognition and tracking of 3D scene points which were reconstructed at an earlier stage of the video processing. As demonstrated in [1] this recovery of scene points reduces drift in sequential algorithms by a large amount. This benefit would be lost if camera retrieval suffers from error buildup due to unmodeled lens distortions.

## 7 Future Work

This work introduced the modeling of lens distortions in a sequential Structure and Motion Recovery framework. The lens distortions were modeled using the radial distortion formulation with a moving distortion center so that de-centering distortions could also be modeled. In [15] tangential and thin prism distortions are also modeled and these formulations could be used to even further optimize the camera model to best fit reality. However, too many free parameters could turn the process unstable and unreliable as one may be modeling noise instead of the physical projection process. This has to be investigated.

**Acknowledgments.** We would like to gratefully acknowledge the financial support of the FWO project G.0223.01 and the IST projects ATTEST and INVIEW.

## References

1. K. Cornelis, M. Pollefeys, and L. Van Gool. Tracking based structure and motion recovery for augmented video productions. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology VRST2001*, pages 17–24, November 2001.
2. J. Batista, H. Araújo, and A. Almeida. Iterative multi-step explicit camera calibration. *IEEE Transactions on Robotics and Automation*, 15(5), October 1999.
3. H. A. Beyer. Accurate calibration of ccd cameras. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition, Urbana Champaign, Illinois, USA*, pages 96–101, June 1992.

4. F. Devernay and O. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. *In Proceedings of SPIE Conference, San Diego, CA*, 2567:62–72, July 1995.
5. Andrew W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *CVPR, Kauai, Hawaii*, 1:125–132, December 2001.
6. R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, November/December 1989.
7. E. Marchand and F. Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. *Rapport de Recherche IRISA, No 1366*, January 2001.
8. M. A. Penna. Camera calibration : A quick and easy way to determine the scale factor. *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1240–1245, December 1991.
9. G. Stein. Internal camera calibration using rotation and geometric shapes. *Master's Thesis, Massachussets Institute of Technology. Artificial Intelligence Laboratory*, 1993.
10. G. Stein. Lens distortion calibration using point correspondences. *In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 143–148, June 1997.
11. Stevenson, Daniel, and M. Fleck. Nonparametric correction of distortion. *TR 95-07, Comp. Sci., University of Iowa*, 1995.
12. R. Swaminathan and S. Nayar. Non-metric calibration of wide angle lenses. *In Proceedings of the 1998 DARPA Image Understanding Workshop, Monterey, California*, November 1998.
13. R. Y. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.
14. G.-Q. Wei and S. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, May 1994.
15. J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, October 1992.
16. R. Willson. Modeling and Calibration of Automated Zoom Lenses. *Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University*, January 1994.
17. Z. Zhang. On the epipolar geometry between two images with lens distortion. *In the Proceedings of the International Conference on Pattern Recognition (ICPR)*, 1:407–411, August 1996.

# Generalized Rank Conditions in Multiple View Geometry with Applications to Dynamical Scenes<sup>\*</sup>

Kun Huang<sup>1</sup>, Robert Fossum<sup>2</sup>, and Yi Ma<sup>1</sup>

<sup>1</sup> Electrical & Computer Engineering Dept., and Coordinated Science Lab.

<sup>2</sup> Mathematics Department, and Beckman Institute

University of Illinois at Urbana-Champaign, Urbana IL 61801, USA

{kunhuang, r-fossum, yima}@uiuc.edu

**Abstract.** In this paper, the geometry of a general class of projections from  $\mathbb{R}^n$  to  $\mathbb{R}^k$  ( $k < n$ ) is examined, as a generalization of classic multiple view geometry in computer vision. It is shown that geometric constraints that govern multiple images of hyperplanes in  $\mathbb{R}^n$ , as well as any incidence conditions among these hyperplanes (such as inclusion, intersection, and restriction), can be systematically captured through certain rank conditions on the so-called multiple view matrix. All constraints known or unknown in computer vision for the projection from  $\mathbb{R}^3$  to  $\mathbb{R}^2$  are simply instances of this result. It certainly simplifies current efforts to extending classic multiple view geometry to dynamical scenes. It also reveals that since most new constraints in spaces of higher dimension are *nonlinear*, the rank conditions are a natural replacement for the traditional multilinear analysis. We also demonstrate that the rank conditions encode extremely rich information about dynamical scenes and they give rise to fundamental criteria for purposes such as stereopsis in  $n$ -dimensional space, segmentation of dynamical features, detection of spatial and temporal formations, and rejection of occluding T-junctions.

**Keywords:** multiple view geometry, rank condition, multiple view matrix, dynamical scenes, segmentation, formation detection, occlusion, structure from motion.

## 1 Introduction

Conventional multiple view geometry typically applies to the case that the scene is static and only the camera is allowed to move. Nonetheless, it is easy to show that, if a scene contains independently moving objects – referred to as a *dynamical scene*, we usually can embed (by a certain formal process) the problem into a space of higher dimension, with a point in the high-dimensional space now representing such as the location and velocity of a moving point in the physical three-dimensional world [13]. However, results and understanding regarding multiple view geometry in such high dimensional spaces are rather sporadic or incomplete at best. This motivates us to seek a systematic generalization of multiple view geometry to higher dimensional spaces.

---

<sup>\*</sup> This material is based upon work partially supported by the U.S. Army Research Office under Contract DAAD19-00-1-0466 and UIUC ECE department startup fund. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

In the computer vision literature, geometric relationships between image features and camera displacements are traditionally described by the so-called *multilinear* matching constraints [8,7,11]. Most of the work since has focused on the algebraic aspects of these multilinear constraints, along with the algorithms which followed from the same formulation. In this framework, different geometric primitives, i.e. points, lines, and planes are treated separately, and analysis of multiple views must rely on a reduction to pairwise [8], triple-wise [11,4,1,6] or quadruple-wise views [12].<sup>1</sup> This line of work culminated in the publication of two monographs on this topic [5,2]. However, it has recently been discovered that a unifying and yet simplifying tool for characterizing geometric relationships among multiple views is a so-called *rank condition* [10]. It gives rise to a *global* constraint for multiple images of multiple features and incidence relations among them. Consequently, certain *nonlinear* relationships among multiple (up to four) images are revealed. As we will also see in this paper, in fact the majority of constraints among multiple images in high dimensional spaces are going to be *nonlinear*. Therefore, the rank condition currently seems to be a reasonable tool left which allows us to systematically generalize multiple view geometry to higher dimensional spaces.

In this paper, our focus is on a *complete* characterization of intrinsic algebraic and geometric constraints that govern multiple  $k$ -dimensional images of hyperplanes in an  $n$ -dimensional space. We show that these constraints can be uniformly expressed in terms of certain rank conditions, which also simultaneously capture geometric relationships among the hyperplanes themselves, such as inclusion, intersection, and restriction. The importance of this study is at least two-fold: 1. In many applications, objects involved are indeed multi-faceted (polygonal) and their shape can be well modeled (or approximated) as a combination of hyperplanes;<sup>2</sup> 2. In some cases, there is not enough information or it is not necessary to locate the exact location of points in a high-dimensional space and instead, we may still be interested in identifying them up to some hyperplane (e.g., in the case of segmentation). As we will point out later, for the special case  $n = 3$  and  $k = 2$ , our results naturally reduce to what is known in computer vision for points, lines, and planes. For the cases  $n > 3$  and  $k = 2$ , our results provide a simpler explanation to extant study on dynamical scenes (e.g., [13]) based on tensor algebra. Since reconstruction is not the main focus of this paper, the reader is referred to [10] for how to use such constraints to develop algorithms for various reconstruction purposes. Nonetheless, since the rank conditions encode extremely rich information about dynamical scenes, we will show how to use it as a basic tool to conduct multiple view analysis in high dimensional spaces, including *stereopsis* in  $n$ -dimensional spaces, *segmentation* of independently moving feature points, detection of spatial and temporal *formations*, and rejection of *occluding T-junctions*.

**Outline of this paper.** Section 2 provides a general formulation of multiple view geometry from  $\mathbb{R}^n$  to  $\mathbb{R}^k$ , including the concepts of camera, camera motion, image, coimage, and preimage. Section 3 fully generalizes classical rank conditions to hyperplanes of arbitrary dimension in  $\mathbb{R}^n$ . Rank conditions for various incidence conditions (i.e. inclu-

<sup>1</sup> Although we now know quadruple wise constraints are completely redundant in the point case.

<sup>2</sup> In case the object consists of smooth curves and surfaces, it is not hard to show that the rank conditions can be easily generalized (see [9]).

sion, intersection, and restriction) among hyperplanes are presented. In Section 4, we discuss numerous potential applications of the rank conditions through a few concrete examples and simulation results.

## 2 Problem Formulation

### 2.1 Euclidean Embedding of Dynamical Scenes

In classic multiple view geometry, we typically consider projecting a three-dimensional *static* scene onto a two-dimensional image plane with respect to multiple camera frames. The standard mathematical model for such a projection is

$$\lambda(t)\mathbf{x}(t) = \Pi(t)\mathbf{X}, \quad (1)$$

where  $\mathbf{X} \in \mathbb{R}^4$  is the homogeneous representation of the 3-D coordinates of a feature point  $\mathbf{p}$  (relative to a world coordinate frame),  $\mathbf{x}(t) \in \mathbb{R}^3$  is its image at time  $t$  (also in homogeneous coordinates),  $\Pi(t) \in \mathbb{R}^{3 \times 4}$  is the projection matrix, and  $\lambda(t)$  is the depth scale of the point  $\mathbf{p}$  with respect to the current camera frame. Typically  $\Pi(t) = [R(t), T(t)]$  where  $R \in \mathbb{R}^{3 \times 3}$ ,  $T \in \mathbb{R}^3$  respectively are the rotation and translation of the camera frame relative to a pre-fixed world frame.<sup>3</sup>

Now suppose the scene contains independently moving (relative to the world frame) feature points. Then the above equation must be modified to

$$\lambda(t)\mathbf{x}(t) = \Pi(t)\mathbf{X}(t). \quad (2)$$

Since now  $\mathbf{X}(t)$  is time-dependent, methods from classic multiple view geometry no longer apply. However, suppose we can find a *time-base* for  $\mathbf{X}(t)$ , i.e. we can express the 3-D coordinates  $\mathbf{X}(t)$  of  $\mathbf{p}$  in terms of a linear combination of some time-varying functions  $b_i(t) \in \mathbb{R}^4$  of time  $t$

$$\mathbf{X}(t) = [b_1(t), b_2(t), \dots, b_k(t)]\bar{\mathbf{X}} \in \mathbb{R}^4, \quad (3)$$

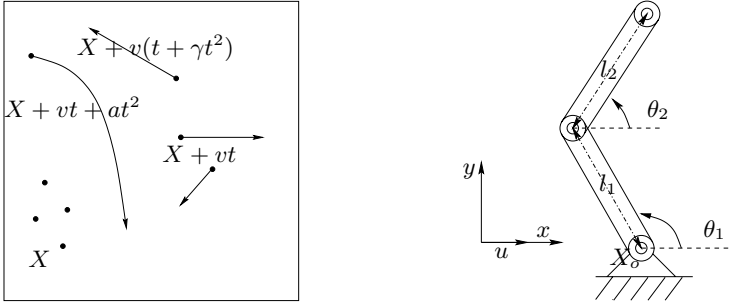
where  $\bar{\mathbf{X}} \in \mathbb{R}^{n+1}$  is a time-independent vector of coefficients. As an important example,  $\mathbf{X}(t)$  is described by linear dynamics. Substitute this back to the above equation, to get

$$\lambda(t)\mathbf{x}(t) = \bar{\Pi}(t)\bar{\mathbf{X}}, \quad (4)$$

where  $\bar{\Pi}(t) \doteq \Pi(t)[b_1(t), b_2(t), \dots, b_k(t)] \in \mathbb{R}^{3 \times (n+1)}$ . This equation is of the same form as (1) except that it represents a (perspective) projection from the space  $\mathbb{R}^n$  to  $\mathbb{R}^2$ . It is then natural to expect that results in classic multiple view geometry should generalize to this class of projections as well.

If a dynamical scene allows such a time-base, we say that the scene allows a *Euclidean embedding*. To avoid redundancy, the time-base functions  $b_i(\cdot)$  should be chosen to be independent functions. The two examples shown by Figure 1 are scenes which do allow such an embedding. In the first case, since the coordinates of all points in the scene can

<sup>3</sup> In case the camera is not calibrated, simply pre-multiply  $R$  and  $T$  by a calibration matrix  $A \in \mathbb{R}^{3 \times 3}$



**Fig. 1.** a) Independent moving features; b) Two resolute joints.

be uniformly described by  $X(t) = X + vt + at^2$  for some  $X, v, a \in \mathbb{R}^3$ , we can simply choose the embedded coordinates of a point to be  $\bar{X} = [X^T, v^T, a^T, 1]^T \in \mathbb{R}^{10}$ . Then we have the following projection equation

$$\lambda(t)x(t) = [R(t), R(t)t, R(t)t^2, T(t)]\bar{X}. \quad (5)$$

Such an embedding would allow us to consider points with trajectories such as parabolic curves. In general, using similar techniques, one may also embed points on any rigid body as shown in Figure 1 b) of multiple links and joints (maybe of other types) into a high dimensional Euclidean space [3].

The above examples have shown the need for generalizing classic multiple view geometry to higher dimensional spaces. Although they both fall into the category of projection from  $\mathbb{R}^n$  to  $\mathbb{R}^2$  for some  $n$ , we will try to bring multiple view geometry into its full potential. That is, we will study the most general case by considering projections from  $\mathbb{R}^n$  to  $\mathbb{R}^k$  with arbitrary  $k < n$ .

## 2.2 Generalized Multiple View Geometry Formulation

**Homogeneous coordinates.** In this paper we will use homogeneous coordinates for both points in  $\mathbb{R}^n$  and its image in  $\mathbb{R}^k$ . Hence  $\bar{X} = [X_1, X_2, \dots, X_n, 1] \in \mathbb{R}^{n+1}$  is the coordinate for a point  $p \in \mathbb{R}^n$  and  $\bar{x} = [x_1, x_2, \dots, x_k, 1] \in \mathbb{R}^{k+1}$  is its image. However by abuse of language, we usually use  $x$  to denote as well the entire ray (1-dimensional subspace) spanned by  $\bar{x}$ , since any vector on this ray gives an equivalent (homogeneous) representation for the image of  $p$ .

**Image formation in high dimensional space.** As a natural generalization of the perspective projection from  $\mathbb{R}^3$  to  $\mathbb{R}^2$ , a perspective projection from  $\mathbb{R}^n$  to  $\mathbb{R}^k$  (with  $k < n$ ) is described by the equation

$$\lambda(t)x(t) = \Pi(t)\bar{X}, \quad (6)$$

where  $x(t) \in \mathbb{R}^{k+1}$  is the (homogeneous) image at time  $t$  of the point  $\bar{X}$ ,  $\lambda(t) \in \mathbb{R}$  is the missing depth scale and  $\Pi(t) \in \mathbb{R}^{(k+1) \times (n+1)}$  is the *projection matrix* of full rank

$k + 1$ .<sup>4</sup> For a dynamical scene,  $\Pi(t)$  may depend on both the (relative) motion of the camera and the scene dynamics. Suppose multiple views of the same point (now in the high-dimensional space) are captured at time  $t_1, \dots, t_m$ . The images of  $\mathbf{X}$  then satisfy the equations

$$\lambda_i \mathbf{x}_i = \Pi_i \mathbf{X}, \quad i = 1, \dots, m, \quad (7)$$

where  $\lambda_i \doteq \lambda(t_i)$ ,  $\mathbf{x}_i \doteq \mathbf{x}(t_i)$ , and  $\Pi_i \doteq \Pi(t_i) \in \mathbb{R}^{(k+1) \times (n+1)}$ .<sup>5</sup> For the rest of the paper, we typically split  $\Pi_i \doteq [\bar{R}_i \ \bar{T}_i]$  with  $\bar{R}_i \in \mathbb{R}^{(k+1) \times (k+1)}$  and  $\bar{T}_i \in \mathbb{R}^{(k+1) \times (n-k)}$ . Note that here  $\bar{R}_i, \bar{T}_i$  are not necessarily the motion (rotation and translation) of the moving camera, although they do depend on the motion.

Since  $\Pi$  is full rank, there always exists a matrix  $g \in \mathbb{R}^{(n+1) \times (n+1)}$  in the general linear group  $GL(n+1, \mathbb{R})$  such that  $\Pi_1 g$  is in the standard form

$$\Pi_1 g = \begin{bmatrix} I_{(k+1) \times (k+1)} & 0_{(k+1) \times (n-k)} \end{bmatrix}, \quad (8)$$

hence for simplicity, we will always assume  $\Pi_1$  is itself in the above form already.<sup>6</sup> The reader should be aware that algebraically we do not lose any generality in doing so.

The following two assumptions make the future study well conditioned:

1. Motion of the camera is generic, i.e. for any  $p$ -dimensional hyperplane in  $\mathbb{R}^n$ , the its image in  $\mathbb{R}^k$  is a hyperplane whose dimension is  $q = \min\{p, k\}$ .
2. Any hyperplane in  $\mathbb{R}^n$  whose image is to be studied has a dimension  $p < k$ . If  $p \geq k$ , then its image will occupy the whole image plane for a generic motion and hence does not provide any useful information.)

The two assumptions above guarantee that we always have  $q = p$  in this paper.

*Remark 1 (Degenerate motions).* Note that motions which violate the first assumption correspond to degenerate configurations which comprises just a zero-measure set of the overall configuration space of the camera and object. In addition, they would only induce minor changes in the results of this paper. A detailed analysis for these degenerate cases can be found in [3] and is omitted here.

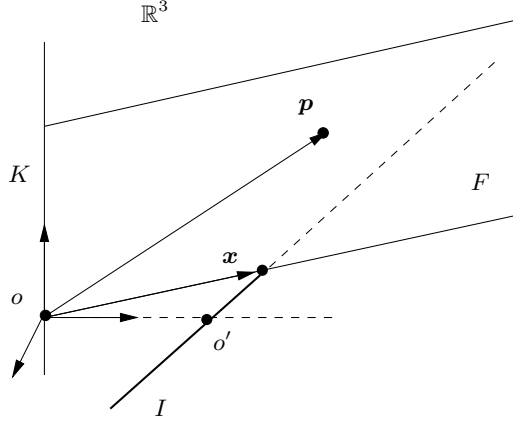
**Image, coimage, and preimage.** For a  $p$ -dimensional hyperplane  $H^p \subseteq \mathbb{R}^n$  whose points satisfy the equation  $\Lambda \mathbf{X} = 0$  (where  $\Lambda \in \mathbb{R}^{(n-p) \times (n+1)}$  is of rank  $n - p$ ), it corresponds to a  $(p + 1)$ -dimensional subspace (i.e. a hyperplane passing through the origin)  $G^{p+1} \subseteq \mathbb{R}^{n+1}$  w.r.t. the camera frame. The image of  $H^p$  is then a  $p$ -dimensional hyperplane  $S^p$  in the image space  $\mathbb{R}^k$  (since  $p < k$ ), it corresponds to a  $(p + 1)$ -dimensional subspace  $U^{p+1} \subset \mathbb{R}^{k+1}$ . Hence the image can be described by the span of a matrix  $\mathbf{s} = [u_1, u_2, \dots, u_{p+1}] \in \mathbb{R}^{(k+1) \times (p+1)}$  or by its maximum complementary

<sup>4</sup> if  $\text{rank}(\Pi) = k' < k + 1$ , then the problem simply becomes the projection from  $\mathbb{R}^n$  to  $\mathbb{R}^{k'-1}$ .

<sup>5</sup> Usually for a static scene with a moving camera, we have  $\Pi_i = \Pi_1 g_i = \Pi_1 \begin{bmatrix} R_i & T_i \\ 0 & 1 \end{bmatrix}$ , where  $R_i \in \mathbb{R}^{n \times n}$  and  $T_i \in \mathbb{R}^n$  are usually the rotation and translation of the camera in  $\mathbb{R}^n$ .

<sup>6</sup> If  $g$  is an affine transformation, it simply corresponds to a different choice of the world reference frame. If  $g$  has to be a projective transformation, then it distinguishes the perspective and orthographic projections. For details, see [3].

orthogonal space (with respect to  $\mathbb{R}^{k+1}$ ), which is spanned by  $\mathbf{s}^\perp = [v_1, v_2, \dots, v_{k-p}] \in \mathbb{R}^{(k+1) \times (k-p)}$  such that  $(\mathbf{s}^\perp)^T \mathbf{s} = 0$ . For clarity, we then call  $\mathbf{s}$  the *image* of the hyperplane and call  $\mathbf{s}^\perp$  its *coimage*. The reader must notice that they are equivalent ways of expressing the same geometric entity on the image plane. In the rest of this



**Fig. 2.** An image  $\mathbf{x}$  of a point  $\mathbf{p} \in \mathbb{R}^3$  under a perspective projection from  $\mathbb{R}^3$  to  $\mathbb{R}^1 (\doteq I)$ .  $K$  is the subspace orthogonal to the subspace spanned by  $o$  and the image plane  $I$ . The plane  $F$  corresponds to the preimage of  $\mathbf{x}$  which is the subspace spanned by  $\mathbf{p}$  and  $K$ .

paper, we will use  $H^p \subseteq \mathbb{R}^n$  and  $G^{p+1} \subseteq \mathbb{R}^{n+1}$  interchangeably to refer to the same object in  $\mathbb{R}^n$  and use  $S^p \subseteq \mathbb{R}^k$  and  $U^{p+1} \subseteq \mathbb{R}^{k+1}$  for the same image entity in  $\mathbb{R}^k$ .

One difference between the image formation in high dimensional space with the classical 3-D case is that the difference between the dimension of the ambient space and that of the image space might be larger than one. This leads to the notion of preimage. For any subspace  $U^{p+1} \subseteq \mathbb{R}^{k+1}$  in the image space, if its equation is  $(\mathbf{s}^\perp)^T \mathbf{x} = 0$ , then define its *preimage* to be the set  $\mathbf{F} = \{\mathbf{X} \in \mathbb{R}^{n+1} : (\mathbf{s}^\perp)^T \Pi \mathbf{X} = 0\}$ , where  $\Pi$  is the corresponding projection matrix. Geometrically, the preimage  $\mathbf{F}$  is the largest set in  $\mathbb{R}^{n+1}$  that can give rise to the same image  $U^{p+1}$ . Its dimension is

$$\dim(\mathbf{F}) = \dim(\mathbf{F} \cap \mathbb{R}^{k+1}) + \dim(\mathbf{F} + \mathbb{R}^{k+1}) - \dim(\mathbb{R}^{k+1}) = (n - k) + p + 1.$$

It corresponds to a  $(n - k + p)$ -dimensional subspace  $\mathbf{F}$  in  $\mathbb{R}^n$ . Figure 2 illustrates the notions of image and preimage for a special case when  $n = 3, k = 1$  and  $p = q = 0$ . The dimension of  $\mathbf{F}$  is  $3 - 1 + 0 = 2$ .

### 3 Generalized Rank Conditions on Multiple View Matrix

A typical problem in multiple view geometry is to systematically express all intrinsic constraints among multiple images of an object (in  $\mathbb{R}^n$ ). By intrinsic we mean that such



constraints should not explicitly depend on the location (or structure) of the object in  $\mathbb{R}^n$ . In this section, we will give a complete description of such constraints including those for various incidence relations among different objects. We will only present the theorems without giving proof. For the complete proof please refer to [10,3].

With the notation introduced in the preceding section, we may formally define a so-called *multiple view matrix* as following:

**Definition 1 (Formal multiple view matrix).** We define a multiple view matrix  $M$  as

$$M \doteq \begin{bmatrix} (D_2^\perp)^T \bar{R}_2 D_1 & (D_2^\perp)^T \bar{T}_2 \\ (D_3^\perp)^T \bar{R}_3 D_1 & (D_3^\perp)^T \bar{T}_3 \\ \vdots & \vdots \\ (D_m^\perp)^T \bar{R}_m D_1 & (D_m^\perp)^T \bar{T}_m \end{bmatrix} \quad (9)$$

where the  $D_i$ 's and  $D_i^\perp$ 's stand for images and coimages of some hyperplanes respectively. The actual values of  $D_i$ 's and  $D_i^\perp$ 's are to be determined in context.

**Theorem 1 (Rank condition for multiple images of one hyperplane).** Given  $m$  images  $s_1, \dots, s_m$  and coimages  $s_1^\perp, \dots, s_m^\perp$  of a  $p$ -dimensional hyperplane  $H^p$  in  $\mathbb{R}^n$ , choose in the above multiple view matrix  $D_1 = s_1$  and  $D_i^\perp = s_i^\perp, i = 2, \dots, m$ , then the resulting matrix  $M$  satisfies

$$\boxed{0 \leq \text{rank}(M) \leq (n - k).} \quad (10)$$

If the hyperplane happens to be a point (i.e.  $p = 0$ ), the theorem easily implies the following result:

**Corollary 1 (Multilinear constraints for  $\mathbb{R}^n \rightarrow \mathbb{R}^k$ ).** For multiple  $(k-D)$  images of a point in  $\mathbb{R}^n$ , non-trivial algebraic constraints involve up to  $(n - k + 2)$ -wise views. These constraints happen to be multilinear and the tensor associated to the  $(n - k + 2)$ -view relationship in fact induces all the other types of tensors associated to smaller numbers of views.

In the classic case  $n = 3, k = 2$ , this corollary reduces to the well-known fact in computer vision that irreducible constraints exist up to triple-wise views, and furthermore the associated (tri-focal) tensor induces all (bi-focal) tensors (i.e. the essential matrix) associated to pairwise views. Of course, besides the examples given in Section 2.1, extra knowledge on the motion of features sometime allows us to embed the problem in a lower dimensional space. These special motions have been studied in [13], but only incomplete lists of constraints among multiple images were given. Our results here clearly *complete* such efforts and imply a much richer set of constraints, not just for point features but also for hyperplanes of any dimension. One must notice that for hyperplanes with dimension higher than 0, most algebraic constraints (as result of the above theorem) will be however *nonlinear*, especially when  $n > 3$ . Hence traditional multilinear analysis will no longer apply. Our approach can also be applied to much more general scenarios and capture all kinds of incidence relations among objects in a high dimensional spaces.

If there are two hyperplanes with one including the other, they give rise to the following theorem which further generalizes Theorem 1:

**Theorem 2 (Rank condition with inclusion).** Consider a  $p_2$ -dimensional hyperplane  $H^{p_2}$  belonging to a  $p_1$ -dimensional hyperplane  $H^{p_1}$  in  $\mathbb{R}^n$ .  $m$  images  $\mathbf{x}_i \in \mathbb{R}^{(k+1) \times (p_2+1)}$  of the  $H^{p_2}$  and  $m$  images  $\mathbf{s}_i \in \mathbb{R}^{(k+1) \times (p_1+1)}$  of the  $H^{p_1}$  relative to the  $i^{\text{th}}$  camera frame are given ( $i = 1, \dots, m$ ). Let the  $D_i$ 's and  $D_i^\perp$ 's in the multiple view matrix  $M$  have the following values

$$\begin{cases} D_i^\perp \doteq \mathbf{x}_i^\perp \in \mathbb{R}^{(k+1) \times (k-p_2)} & \text{or } \mathbf{s}_i^\perp \in \mathbb{R}^{(k+1) \times (k-p_1)}, \\ D_i \doteq \mathbf{x}_i \in \mathbb{R}^{(k+1) \times (p_2+1)} & \text{or } \mathbf{s}_i \in \mathbb{R}^{(k+1) \times (p_1+1)}. \end{cases} \quad (11)$$

Then for all possible instances of the matrix  $M$ , we have the two cases:

1. case one: If  $D_1 = \mathbf{s}_1$  and  $D_i^\perp = \mathbf{x}_i^\perp$  for some  $i \geq 2$ , then

$$\boxed{\text{rank}(M) \leq (n - k) + (p_1 - p_2),}$$

2. case two: Otherwise,

$$\boxed{0 \leq \text{rank}(M) \leq n - k.}$$

Since  $\text{rank}(AB) \geq (\text{rank}(A) + \text{rank}(B) - n)$  for all  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times k}$ , we have  $\text{rank}[(D_i^\perp)^T \bar{R}_i D_1] \geq (p_1 - p_2)$  if the matrix  $\bar{R}_i \in \mathbb{R}^{(k+1) \times (k+1)}$  is full rank for some  $i \geq 2$ . So the rank condition for the case one can be improved with a tight lower bound

$$\boxed{(p_1 - p_2) \leq \text{rank}(M) \leq (n - k) + (p_1 - p_2).}$$

This theorem can be easily generalized to any set of cascading hyperplanes

$$H^{p_l} \subseteq H^{p_{l-1}} \subseteq \dots \subseteq H^{p_1},$$

for some  $l \in \mathbb{Z}_+$ . We omit the details for simplicity.

For two hyperplanes intersecting at a third, we have

**Theorem 3 (Rank condition with intersection).** Consider hyperplanes  $H^{p_1}$ ,  $H^{p_2}$ , and  $H^{p_3}$  with  $H^{p_3} \subseteq H^{p_1} \cap H^{p_2}$ . Given their  $m$  images relative to  $m$  camera frames as above, let the  $D_i$ 's and  $D_i^\perp$ 's in the multiple view matrix  $M$  have the following values:  $D_1 \doteq \mathbf{x}_1$ , and  $D_i^\perp \doteq \mathbf{x}_i^\perp, \mathbf{r}_i^\perp, \mathbf{s}_i^\perp$  being the coimages of  $H^{p_3}, H^{p_1}, H^{p_2}$  respectively. Then we have

$$\boxed{0 \leq \text{rank}(M) \leq (n - k).}$$

This theorem can be easily generalized to a family of intersecting hyperplanes

$$H^p \subseteq H^{p_l} \cap H^{p_{l-1}} \cap \dots \cap H^{p_1}, \quad (12)$$

for some  $l \in \mathbb{Z}_+$ . We here omit the details for simplicity.

In practice, there are situations when all hyperplanes being observed belong to a  $p$ -dimensional ambient hyperplane in  $\mathbb{R}^n$ ,<sup>7</sup> and the location of this ambient hyperplane

<sup>7</sup> Here we no longer require that  $p$  is less than  $k$  since the image of this ambient hyperplane is not of interest.

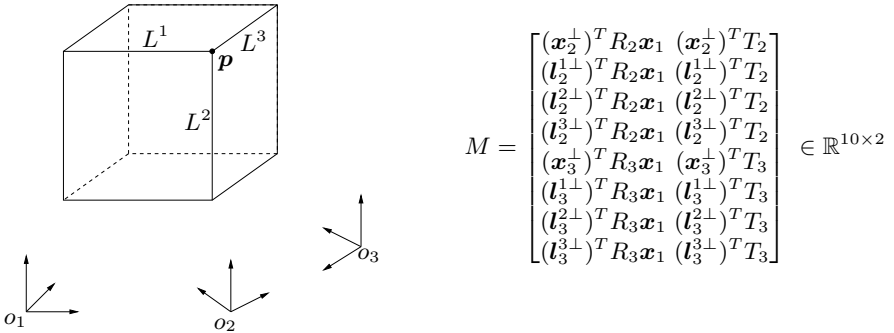
relative to the world reference frame is fixed. In general, a  $p$ -dimensional hyperplane  $H^p$  in  $\mathbb{R}^n$  can be described by a full-rank  $(n-p) \times (n+1)$  matrix  $A \in \mathbb{R}^{(n-p) \times (n+1)}$  such that any point  $p \in H^p$  satisfies:  $AX = 0$ , where  $X \in \mathbb{R}^{n+1}$  is the homogeneous coordinate of the point  $p$ . We call the matrix  $A$  the *homogeneous representation* for  $H^p$ . Of course, such a representation is not unique – any two matrices  $A_1$  and  $A_2$  with the same kernel give rise to the same hyperplane. For convenience, we usually divide the  $(n-k) \times (n+1)$  matrix  $A$  into two parts

$$A = [A^1, A^2], \quad \text{with } A^1 \in \mathbb{R}^{(n-p) \times (k+1)}, \quad A^2 \in \mathbb{R}^{(n-p) \times (n-k)}. \quad (13)$$

Then we have

**Theorem 4 (Rank condition with restriction).** *If all the feature hyperplanes belong to an ambient hyperplane homogeneously described by the matrix  $A \in \mathbb{R}^{(n-p) \times (n+1)}$ , then by appending a block of rows  $[A^1 D_1 \ A^2]$  to the multiple view matrix  $M$ , all the rank conditions in the above three theorems remain the same for the new matrix.*

*Example 1 (Multiple views of a cube).* The above theorems allow us to express all incidental constraints associated to a particular object. For instance, as shown in Figure 3, there are three edges  $L^1, L^2$  and  $L^3$  intersecting at each vertex  $p$  of a cube. Then from three images of the cube from three vantage points, we get the following multiple view matrix  $M$  (see Figure 3) associated to the vertex  $p$ , where  $x_i \in \mathbb{R}^3$  is the image of the vertex  $p$  in the  $i^{th}$  view, and  $l_i^j \in \mathbb{R}^{3 \times 2}$  is the image of the  $j^{th}$  edge  $L^j$  in the  $i^{th}$  view.<sup>8</sup>



**Fig. 3.** Images of a standard cube from three vantage points.

The condition  $\text{rank}(M) \leq 1$  then expresses the incidence condition among the vertex and three edges in terms of their three images, without explicitly referring to their 3-D location. However still more can be said. The above matrix  $M$  only captures the fact that the edges all pass the vertex in all views, it does not captures the incidence condition for the edges themselves and also not capture the fact that the four vertices are on the same plane. These constraints can be described by other (types of) multiple view matrices which are also just instances of above theorems.

<sup>8</sup> Traditionally the symbol  $l$  is used to describe the *coimage* of a line. In our case, that becomes  $l^\perp$ , which is indeed a three dimensional vector.

**Remark 2 (Rank values).** In the above rank conditions, the highest rank value typically corresponds to a generic configuration, every value in between the highest and lowest corresponds to a different class (or type) of degenerate configurations of the object w.r.t. the camera positions (see [3]).

**Remark 3 (Fixed camera).** Note that in many practical situations, as we will see from examples in the following section, one typically has a *fixed* camera recording a dynamical scene. In this case, the projection matrix  $\Pi$  might have a column with all zeros for all views/time. Then in above theorems, the rank of the multiple view matrix should drop 1.

## 4 Applications and Examples

In classic multiple view geometry, a primary purpose of deriving the above rank conditions or the constraints among multiple images is for a full reconstruction of camera motion as well as the location of the objects being observed. Technical conditions and algorithms for similar purposes in higher dimensional spaces are however still largely unknown. Nonetheless, in this section, we demonstrate through a few examples how information about the camera motion and scene structure is extensively encoded by the multiple view matrix and its associated rank conditions.

### 4.1 Multiple View Stereopsis in $n$ -Dimensional Space

For a  $p$ -dimensional hyperplane  $H^p$  in  $\mathbb{R}^n$ , if we can obtain its coimages  $s_i^\perp$ 's, with known projection matrices  $\Pi_i = [\bar{R}_i, \bar{T}_i]$ , the question is “What is the least number of images we need in order to determine  $H^p$ ?” In the classical 3-D space, this is known as *stereopsis*. To this end, we need to introduce another matrix

$$C = \begin{bmatrix} (s_1^\perp)^T \bar{R}_1 & (s_1^\perp)^T \bar{T}_1 \\ (s_2^\perp)^T \bar{R}_2 & (s_2^\perp)^T \bar{T}_2 \\ \vdots & \vdots \\ (s_m^\perp)^T \bar{R}_m & (s_m^\perp)^T \bar{T}_m \end{bmatrix} \in \mathbb{R}^{[m(k-p)] \times (n+1)},$$

which is related to  $M$  by

$$\text{rank}(C) = \text{rank}(M) + (k - p). \quad (14)$$

If  $G^{p+1}$  is the corresponding subspace in  $\mathbb{R}^{n+1}$  for  $H^p$ , then  $G^{p+1} \subseteq \ker(C)$ . Hence when the rank of  $M$  reaches its upper bound  $n - k$ , the rank of  $C$  reaches its upper bound  $n - p$ , which means that the kernel of  $C$  has dimension  $p + 1$ . This further implies that  $G^{p+1} = \ker(C)$  and we can reconstruct  $H^p$  uniquely by calculating the kernel of  $C$ . On the other hand, if  $\text{rank}(C) = l < n - p$ , then we can only recover the hyperplane up to an  $(n - l)$ -dimensional hyperplane in  $\mathbb{R}^n$ .

Note that  $C$  is a stack of  $m(k - p) \times (n + 1)$  matrices  $[(s_i^\perp)^T \bar{R}_i \ (s_i^\perp)^T \bar{T}_i]$ ,  $i = 1, \dots, m$ . The kernel of the  $i^{\text{th}}$  block is the preimage  $F_i$  of  $s_i$ . Hence, each view actually

contributes to a reduction in the dimension of the kernel of  $C$  and  $\ker(C) = \cap_{i=1}^m \mathbf{F}_i$ . In order to reconstruct the original subspace  $G^{p+1}$ , the dimension of the kernel should be reduced from  $\dim(\mathbf{F}_1) = n - k + p + 1$  to  $p + 1$ . The reduction in dimension is then  $n - k$ . If  $K$  is the kernel of matrix  $C$  composed of  $i - 1$  views, then the dimension reduction of the kernel contributed by the  $i^{th}$  view is

$$\begin{aligned} \dim(K) - \dim(K \cap \mathbf{F}_i) &= \dim(K + \mathbf{F}_i) - \dim(\mathbf{F}_i) \\ &\leq (n + 1) - (n - k + p + 1) = k - p. \end{aligned} \quad (15)$$

Thus, in order to uniquely determine  $H^p$ , we need at least  $m = \lceil \frac{n-k}{k-p} \rceil + 1$  views under general configuration. However, this is the “optimal” case such that each view can contribute maximum dimension reduction of the kernel of  $C$ . The maximum number of general views required is  $m = (n - k + 1)$  in which case each view only contributes to a one-dimensional reduction of the kernel of  $C$ . For example, in the special case for point features, we have  $p = 0$ . Hence the minimum number of independent views required is then  $\lceil \frac{n-k}{k} \rceil + 1$ . When  $n = 3, k = 2$ , this number reduces to 2 which is well-known for 3-D stereopsis. For the dynamical scene problem with  $n = 6, k = 2$  studied in [13], in general we need 5 views to reconstruct the point unless we have an optimal configuration for which 3 views suffice.

## 4.2 Segmentation and Formation Detection

**Segmentation by the rank.** Now we consider the dynamical scene example we introduced in the beginning of this paper. For any point moving with up to constant acceleration, it can be described by a homogeneous coordinate in the  $\mathbb{R}^{10}$ :  $\mathbf{X} = [X^T, v^T, a^T, 1]^T \in \mathbb{R}^{10}$ , where  $X \in \mathbb{R}^3, v \in \mathbb{R}^3$ , and  $a \in \mathbb{R}^3$  are the point’s initial location, velocity, and acceleration, respectively. With respect to a fixed camera, its image  $\mathbf{x}_i$  at time  $t_i$  then satisfies:  $\lambda_i \mathbf{x}_i = \Pi_i \mathbf{X}$ , where  $\lambda_i \in \mathbb{R}$  and

$$\Pi_i = [\bar{R}_i \ \bar{T}_i] \in \mathbb{R}^{3 \times 10}, \quad \bar{R}_i = I \in \mathbb{R}^{3 \times 3}, \quad \bar{T}_i = [It_i \ It_i^2/2 \ 0] \in \mathbb{R}^{3 \times 7}. \quad (16)$$

Hence the associated multiple view matrix is

$$M = \begin{bmatrix} (\mathbf{x}_2^\perp)^T & \mathbf{x}_1 & t_2 & (\mathbf{x}_2^\perp)^T & \frac{t_2^2}{2} & (\mathbf{x}_2^\perp)^T & 0 \\ (\mathbf{x}_3^\perp)^T & \mathbf{x}_1 & t_3 & (\mathbf{x}_3^\perp)^T & \frac{t_3^2}{2} & (\mathbf{x}_3^\perp)^T & 0 \\ \vdots & & & \vdots & & \vdots & \\ (\mathbf{x}_m^\perp)^T & \mathbf{x}_1 & t_m & (\mathbf{x}_m^\perp)^T & \frac{t_m^2}{2} & (\mathbf{x}_m^\perp)^T & 0 \end{bmatrix} \in \mathbb{R}^{[2(m-1)] \times 8}.$$

From the rank condition and remark 2 we know that  $\text{rank}(M) < 9 - 2 = 7$ .

By randomly choosing values for  $X$ ,  $v$ , and  $a$  for numerous points, simulation results showed that we always have:

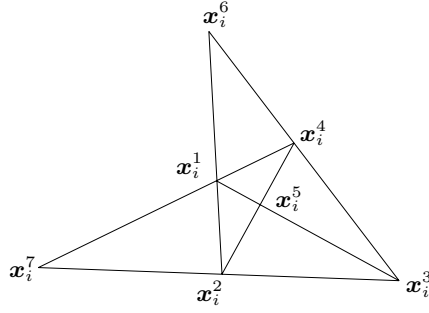
1.  $\text{rank}(M) = 4$ , if the point is static, i.e.  $a = 0$ ,  $v = 0$ .
2.  $\text{rank}(M) = 5$ , if the point is moving with constant velocity, i.e.  $a = 0$ .
3.  $\text{rank}(M) = 6$ , if the point is moving with constant acceleration, i.e.  $a \neq 0$ .

These results can be explained by studying the kernel of  $M$ . Note that the kernel of  $M$  always contains a trivial vector  $l = [0 \ 0_{3 \times 1} \ 0_{3 \times 1} \ 1]^T$ . For the case that  $v = 0$  and  $a = 0$ , we have  $x_i = x_1$ . So the first column of  $M$  is always 0. A basis of the kernel of  $M$  is then  $l$ ,  $[1 \ 0_{3 \times 1} \ 0_{3 \times 1} \ 0]^T$ ,  $[0 \ x_1^T \ 0_{3 \times 1} \ 0]^T$  and  $[0 \ 0_{3 \times 1} \ x_1^T \ 0]^T$ .<sup>9</sup> If we only have  $a = 0$ , then a basis for the kernel of  $M$  is  $l$ ,  $[0 \ \lambda_1 x_1^T \ 2v^T \ 0]^T$ , and  $[\lambda_1 \ v^T \ 0_{3 \times 1} \ 0]^T$ . Finally for the most general case, the basis for the kernel is  $l$  and  $[\lambda_1 \ v^T \ a^T \ 0]^T$ . The upper bound 7 for the rank is never achieved because the camera location is fixed. There will always be a one-parameter family of ambiguity for the location of a point in  $\mathbb{R}^{10}$ .

Hence rank conditions provide a simple method for doing segmentation or grouping of feature points in a dynamical scene. Given the correspondences of the images points at different time, we can easily differentiate background (static) points, points moving on a straight line with constant velocity and points moving with parabolic trajectory. Since we approximate the motion up to the second order, most practical scenarios studied in the literature so far [13] fall into this case.

**Formation detection by the rank.** Given the image correspondences for a set of points, sometimes we want to detect whether they move (relative to each other) in a similar fashion. If so, we say they move in a *formation*. As we will see, this is a more general notion than *grouping* of static points. In practice, moving together in “formation” often implies that features considered are more likely from the same object. Here we demonstrate a special case: images of four points with constant 3-D velocities, and show how the formation information can be encoded in the rank conditions. Denote the four points as  $p^1$ ,  $p^2$ ,  $p^3$  and  $p^4$ , and their velocities as  $v^1, v^2, v^3$  and  $v^4$ , respectively. The corresponding images in the  $i^{th}$  frame are  $x_i^1, x_i^2, x_i^3$  and  $x_i^4$ . In general, the four image points can be linked to generate six virtual lines. Intersections of these lines give three virtual points on the image plane, as shown in Figure 4, which are labeled by  $x_i^5, x_i^6$  and  $x_i^7$ . Then we can associate multiple view matrices  $M^5, M^6$  and  $M^7$  to these (virtual) images of points and lines. However, since these image points may not necessarily correspond to physical points (moving or not) in 3-D space, the rank of the corresponding matrix may vary according to the relative motion among the four points. Table 1 summarizes results from simulation performed in various cases. In the table, it is clear that when the four points are coplanar and moving in that plane with the same velocity, then the

<sup>9</sup> The kernel can be calculated in the following way, let the vector be  $V = [u_0, u_1^T, u_2^T, 0]^T \in \mathbb{R}^8$  where  $u_0 \in \mathbb{R}$  and  $u_1, u_2 \in \mathbb{R}^3$ . If  $V \in \ker(M)$ , then  $[(x_i^1)^T \ x_1 \ (t_i x_i^1)^T \ (t_i^2 x_i^1 / 2)^T \ 0]^T V = 0$  for all  $i = 2, \dots, m$ . Combined with the fact that  $\lambda_i x_i = \lambda_1 x_1 + t_i v + \frac{t_i^2}{2} a$ , we can write this into a “polynomial” of  $t_i$ . By setting the coefficients to zeros we can solve for  $u_0, u_1$  and  $u_2$ .



**Fig. 4.** Three apparent “intersections” ( $x_i^5$ ,  $x_i^6$  and  $x_i^7$ ) generated by the four image points.

**Table 1.** Relationship between the rank of the multiple view matrices associated with the virtual image points and the spatial & motion formations of the four points in 3-D space.

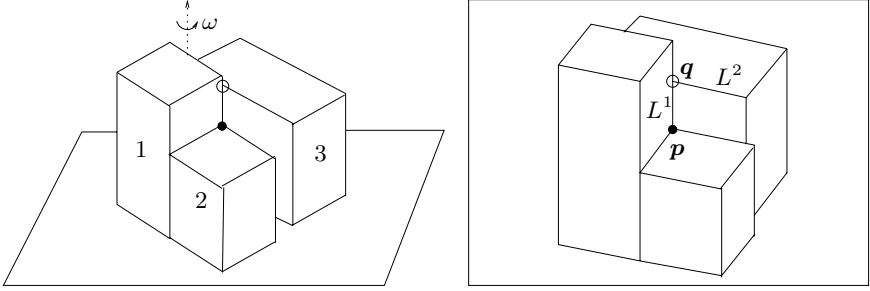
Motion formations	Spatial formations	$\text{rank}(M^5)$	$\text{rank}(M^6)$	$\text{rank}(M^7)$
$v^1 = v^2 = v^3 = v^4$	Coplanar	5	5	5
$v^1 = v^3, v^2 = v^4 = kv^1$ $k \in \mathbb{R}$ and $k \neq 1$	Coplanar	5	6	6
$v^1 = v^2 = v^3$	Coplanar	6	6	6
$v^1 = v^2 = v^3 = v^4$	Not coplanar	6	6	6

additional intersection points correspond to physical points in 3-D space which are just intersections of the lines connecting the four points in 3-D. They should move with the same velocity, which is why we always obtain rank 5 for the three matrices. For the other cases, virtual images typically do *not* correspond to any physical points. Still the rank of associated matrices can tell us information about the formation of the feature points under different scenarios. These results give us simply a glimpse of how rich 3-D information we may gain by merely playing with the rank of the multiple view matrix. Although our current results do not provide an analytical explanation to the relationship between the formation and the rank conditions, at least they give some necessary conditions which can be used at early stage of establishing correspondence, grouping, or segmentation. We believe a thorough study will lead to fruitful theoretical results.

### 4.3 Distinguish Corners and Occluding T-Junctions by the Rank

A fundamental problem which troubles structure from motion is that feature points, as “corners” or “T-junctions”, extracted from the image do not necessarily correspond to physical points in the 3-D world. For example, in Figure 5, three blocks are on a table. Blocks 1 and 2 are adjacent to each other hence the corner  $p$  can be treated as a real 3-D point. However, the “point”  $q$  appears in the image as the occluding T-junction of the two edges  $L^1$  and  $L^2$ . It does not correspond to any physical point in 3-D space. The question is: “Can we extract some visual information from a moving scene so that

these two types of feature points can be distinguished?” Suppose that the table is rotating around its normal, denoted as the  $X$ -axis, with angular velocity  $\omega$ . The axis intersects the table at a point  $[0, Y_0, Z_0]^T$  (where  $Y_0$  and  $Z_0$  are unknown). Then for any point with



**Fig. 5.** Three rectangular blocks on a table with blocks 1 and 2 adjacent to each other. Point  $p$  is the corner formed by blocks 1 and 2. Point  $q$  appears on image as the occluding T-junction of lines  $L^1$  and  $L^2$ . In 3-D space these two lines do *not* intersect.

initial coordinate  $[X, Y, Z]^T$ , its image  $x(t)$  at any time  $t$  satisfies

$$\lambda(t)x(t) = \Pi(t)\bar{X} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \cos(\omega t) & -\sin(\omega t) & 0 \\ 0 & 0 & 1 & \sin(\omega t) & \cos(\omega t) & 0 \end{bmatrix} \bar{X}, \quad (17)$$

where  $\bar{X} = [X, Y_0, Z_0, \tilde{Y}, \tilde{Z}, 1]^T \in \mathbb{R}^6$  with  $\tilde{Y} = Y - Y_0$ , and  $\tilde{Z} = Z - Z_0$ . Under this setup, the image can be viewed as a projection from  $\mathbb{R}^5$  to  $\mathbb{R}^2$ . Since the projection matrix always has the last column being 0, the rank for the multiple view matrix associated to any point (in  $\mathbb{R}^5$ ) should be strictly less than 3 (again, because the camera is not moving). Simulation results demonstrate that for a sufficient number of images we have:<sup>10</sup>

1.  $\text{rank}(M) = 2 < 3$  for images of the point  $p$ ,
2.  $\text{rank}(M) = 3$  for virtual images of the T-junction  $q$ .

This example shows that at least in some cases the rank condition can serve as a criterion for determining whether or not a “feature point” in the image actually corresponds to some physical point in 3-D (from their motion). It may provide a means to reject T-junctions which are the result of occlusion, like the point  $q$  in Figure 5.

<sup>10</sup> Note that at time  $t = 0$ , the projection matrix is not in the standard form  $[I, 0]$ , so we need to multiply another matrix to each  $\Pi(t)$  to obtain this form, and the multiple view matrix is also modified accordingly.



## 5 Summary

The main result in this paper is the presentation of generalized rank conditions associated to the multiple view matrix for perspective projection from  $\mathbb{R}^n$  to  $\mathbb{R}^k$ . These conditions provide a complete set of intrinsic constraints that govern multiple images of objects in high dimensional spaces. The theory is general enough to enable geometric analysis for many dynamical scenes (after embedded into a higher dimensional Euclidean space), as an extension to classic multiple view geometry. In addition to its potential for purposes such as recovering camera motion and scene structure and dynamics, many new problems and phenomena arise in the setting of dynamical scenes from broad applications of the multiple view matrix and its rank conditions. They include (but not limited to): stereopsis, segmentation and grouping, formation detection, and occlusion detection. A full geometric and algebraic characterization for these problems and phenomena remains largely open.

In this paper we did not address at all how to use such rank conditions to facilitate the recovery of camera motion and scene dynamics. But it provides a systematic way to eliminate redundant parameters and reduce constraints among image sequences to its (almost) minimum. Further estimation of unknown parameters using either tensorial techniques or direct minimization is a matter of algorithm design. In addition to such theoretical endeavor, we are currently conducting experiments on videos of multiple moving objects (mobile robots) as well as for the purpose of tracking and estimating human body movement.

## References

1. S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 4(4), pp.293-306, 1998.
2. O. Faugeras, Q.-T. Luong, and T. Papadopoulos. *Geometry of Multiple Images*. The MIT Press, 2001.
3. R. Fossum, K. Huang, Y. Ma. General rank conditions in multiple view geometry. *UIUC, CSL Technical Report, UILU-ENG 01-2222 (DC-203)*, October 8, 2001.
4. R. Hartley. Lines and points in three views - a unified approach. In *Proceedings of 1994 Image Understanding Workshop*, pp. 1006–1016, Monterey, CA USA, 1994. OMNIPRESS.
5. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.
6. A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in Applied Sciences*, 20(13), pp.1135-1162, 1997.
7. Y. Liu and T.S. Huang. Estimation of rigid body motion using straight line correspondences *IEEE Workshop on Motion: Representation and Analysis*, Kiawah Island, SC, May 1986.
8. H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, pp.133-135, 1981.
9. Y. Ma, K. Huang, and J. Košecká. New rank deficiency condition for multiple view geometry of line features. *UIUC, CSL Technical Report, UILU-ENG 01-2209 (DC-201)*, May 8, 2001.
10. Y. Ma, K. Huang, R. Vidal, J. Košecká, and S. Sastry. Rank conditions of multiple view matrix in multiple view geometry. *UIUC, CSL Technical Report, UILU-ENG 01-2214 (DC-220)*, submitted to IJCV, June 18, 2001.
11. M. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–184, 1990.

12. B. Triggs. Matching constraints and the joint image. In *Proceedings of Fifth International Conference on Computer Vision*, pp.338-343, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.
13. L. Wolf and A. Shashua. On projection matrices  $\mathcal{P}^k \rightarrow \mathcal{P}^2$ ,  $k = 3, \dots, 6$ , and their applications in computer vision. In *Proceedings of the Eighth International Conference on Computer Vision*, pp.412-419, Vancouver, Canada, 2001.
14. G. Sparr. A common framework for kinetic depth, reconstruction and motion for deformable objects. *Proceedings of the Fourth European Conference on Computer Vision*, pp.471-482, Cambridge, England, 1994.

# Dense Structure-from-Motion: An Approach Based on Segment Matching

Fabian Ernst, Piotr Wilinski, and Kees van Overveld

Philips Research, Prof. Holstlaan 4,  
NL 5656AA Eindhoven, The Netherlands,  
fabian.ernst@philips.com

**Abstract.** For 3-D video applications, dense depth maps are required. We present a segment-based structure-from-motion technique. After image segmentation, we estimate the motion of each segment. With knowledge of the camera motion, this can be translated into depth. The optimal depth is found by minimizing a suitable error norm, which can handle occlusions as well. This method combines the advantages of motion estimation on the one hand, and structure-from-motion algorithms on the other hand. The resulting depth maps are pixel-accurate due to the segmentation, and have a high accuracy: depth differences corresponding to motion differences of  $1/8^{th}$  of a pixel can be recovered.

## 1 Introduction

Video technology at the turn of the century is facing the dilemma of finding features which increase product competitiveness. The third dimension has the potential of revolutionizing future video and TV products. Recent progress in computer vision brings closer the possibility of extracting depth information from video sequences. For many 3-D video applications, this depth information should be *dense*, i.e., every pixel should have depth information.

Several methods to extract depth out of video exist. Structure-from-motion (SFM) techniques are based on the extraction of structure from apparent motion. Two kinds of motion can be found in video sequences. The first one is due to camera motion: Objects at different depths have different apparent motion. The second one is independent motion of the objects. Handling independent motion of the objects requires handling of static scenes as a necessary first step. Therefore we assume for the time being that our scenes are static. Alternatively, we can extract depth from stereo sequences, allowing us to apply the same algorithm for all kinds of dynamic and static scenes.

SFM techniques have a long history in computer vision, and several overviews of techniques are available (for instance [9]). Feature-based algorithms [3] first extract predefined features, such as corners, and then match these. The difficulty in correspondence estimation lies in the size of the search space, and the presence of ‘false correspondences’. Also features may disappear, or change their characteristics. Furthermore, the accuracy of feature estimation in images is often limited. The accuracy of the structure estimation can be improved using multiple frames, for instance through dynamic models [15]. Since the use of

feature points is limited to regions in the image where enough information is contained, they are, in general, of limited use for direct dense matching.

A possibility to obtain dense depth maps is to use a feature-based technique as a starting point, and subsequently to generate a dense depth map [11]. Dense stereo matching methods [12] are commonly devised specifically for horizontal motions. Optical flow techniques are useful for obtaining dense depth maps in the presence of small displacements, but require regularization to obtain smooth depth maps [2]. This problem is avoided when we compare small regions of the image with corresponding regions in the following images [1,16]. For each region in the first image, the region in the second image is sought which is most similar to it according to a certain criterium (the *match penalty*). For depth reconstruction, we need additionally an (assumed) camera transformation, that converts a motion vector into a depth value. We assume that this camera transformation is available or can be estimated using calibration algorithms.

Generally, all pixels within a given region are assumed to move uniformly, and hence have the same depth. Square blocks of pixels are most common, and have been used with success in video coding applications [8] for displacement estimation and motion compensation. However, with a fixed given region shape (such as a block), within one region several objects with different depth values could be present, which leads to artifacts. Dividing the blocks into smaller entities can avoid this [13]; however, regions of constant depth in an image are commonly (much) larger than blocks. Hence, we propose to create *segments* corresponding to underlying objects, or more accurate, to segment the image in regions containing only a single depth. This leads to a chicken-and-egg problem (for the segmentation we need the depth; however, for depth estimation we use the segments). For foreground-background segmentation or scenes containing a small number of well-defined objects, one could iteratively solve for scene structure and segmentation [22]. For general video scenes, we have to make a key assumption (which is valid for a large amount of video footage, but of course fails in some situations): *Discontinuities in depth coincide with discontinuities in color*.

In the remainder of this paper we discuss our dense structure-from-motion algorithm (DSFM), which is based on segment matching. In section 2, we discuss the choice of our error norm and how to minimize it in order to find the depth per segment. We also shortly explain the segmentation algorithm. Due to the presence of noise and regions with low texture, the depth map may contain artifacts. We apply a post-processing procedure for smoothing depth artifacts, taking into account depth uncertainty intervals. Results are shown in section 3. To improve the matching, we introduce a way of handling occlusions in section 4.

## 2 Dense Structure-from-Motion

### 2.1 General Overview

Our dense SFM algorithm consists of the following key components:

*Camera calibration.* We have to determine the internal geometric and optical characteristics of the camera (intrinsic parameters) and the 3-D position and

orientation of the camera's frame relative to a certain world coordinate system (extrinsic parameters). This is required to enable the conversion of an apparent motion to a depth value. For instance, if the camera motion consists of horizontal translation, the parallax is inversely proportional to depth. Several calibration methods have been described in the literature [4,18]. An overview of general algorithms is given in [6]. Camera calibration is not discussed in this paper.

*Image segmentation.* We assume that depth discontinuities coincide with color discontinuities. Segmentation divides the image into regions of more or less constant color, and hence, depth. The segmentation algorithm is discussed in section 2.4.

*Segment matching.* This is the key element of this paper and is discussed in sections 2.2 and 2.3. It results in a pixel-wise depth map, annotated with depth uncertainty intervals.

*Postprocessing.* This *relaxation* algorithm is discussed in section 2.5 and allows to improve the accuracy of the depth maps with information from neighboring segments.

## 2.2 Segment Matching

The core of our dense structure-from-motion algorithm (DSFM) is the matching of segments in the first image  $I_0$  to find their location in the next image  $I_1$ . To each segment  $S$ , a depth  $\hat{d}$  is assigned which minimizes the following error norm, the so-called *match penalty* [5]:

$$E(d) = \sum_{\mathbf{x} \in S} |I_1(\mathbf{x} + \Delta\mathbf{x}(d)) - I_0(\mathbf{x})|, \quad (1)$$

i.e., we estimate for all pixels  $\mathbf{x}$  in a segment  $S$  their location in  $I_1$  based on the proposed depth, and compute the absolute difference between the colors at their positions in  $I_0$ . The camera transformation relates the depth value  $d$  to a motion vector  $\Delta\mathbf{x}$ . Note that we do *not* match a segment with a segment of frame  $I_1$ , since this would put high requirements on the time-consistency of the segmentation.

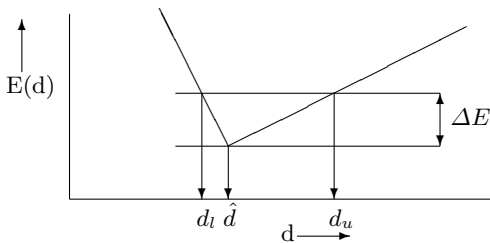
This matching approach comes from the domain of motion estimation, for instance for scan-rate conversion [5] or MPEG compression [7]. The key difference between our method and existing matching approaches is that in our case the region  $S$  is an arbitrarily shaped segment, instead of a square region. Another difference is that the minimization is 1-D (over depths  $d$ ), instead of 2-D (the two components  $(\Delta x, \Delta y)$  of a motion vector). In case of non-integer pixel displacements, we use bilinear interpolation to estimate color values at inter-pixel positions.

Solving equation (1) not only results in a depth per segment, but can give us also information on its accuracy by considering the error curve: the error as function of the depth (see figure 1 for an illustration). We define the boundaries of the depth uncertainty interval as:

$$d_l = \max\{d : d < \hat{d} \wedge E(d) > E(\hat{d}) + \Delta E\}, \quad (2)$$

$$d_u = \min\{d : d > \hat{d} \wedge E(d) > E(\hat{d}) + \Delta E\}, \quad (3)$$

i.e., the depth values closest to  $\hat{d}$  for which the match penalty is still above a predefined threshold.



**Fig. 1.** Illustration of the depth accuracy. The error as function of depth has a global minimum in  $\hat{d}$ . The lower and upper depth bounds are given by  $d_l$  and  $d_u$  according to expressions (2) and (3).

### 2.3 Solving the Minimization Problem: The “Candidates” Concept

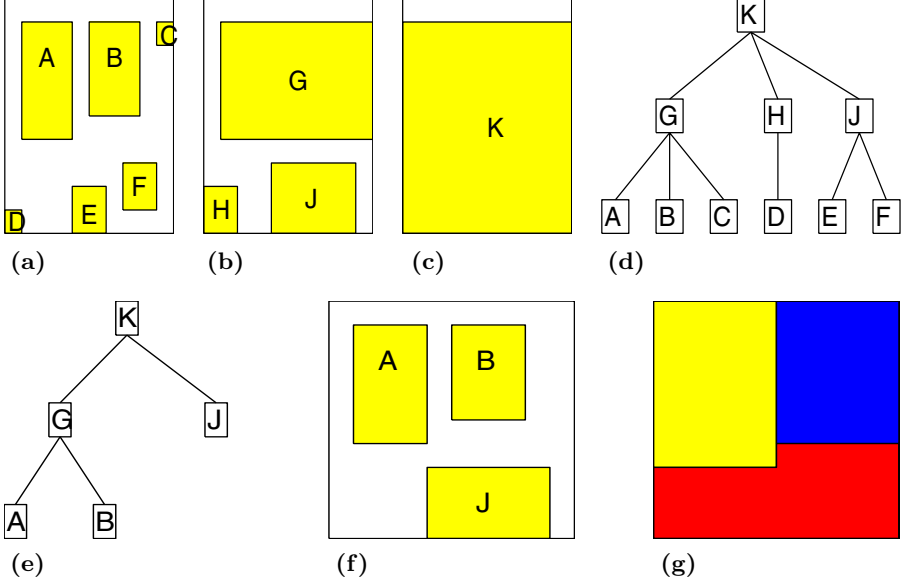
A naive version of matching would do a full search over all possible depths  $d$ . For applications requiring real-time performance, full search is computationally too expensive. Hence the number of depths to be evaluated has to be reduced, however, without affecting the quality of the result. For block-based motion estimation, a fast method for solving minimization problem (1) has been proposed [5] which uses for each block only motion vectors  $\Delta\mathbf{x}_i$  coming from a small candidate set (CS). This CS is composed of motion vectors of some well-selected neighbors of the block under consideration, and some random vectors. We use a modified version of this algorithm by assigning a depth value  $d$  to a segment as a result of minimizing the match error  $E(d)$  over a limited set of candidate depth values. It is assumed that the candidates sample the function  $E(d)$  sufficiently dense. Since we deal with arbitrarily shaped segments, the number of neighboring segments is a priori unknown and might be different for each segment. To keep the notion of a small CS, we might reduce the number of candidates by taking only neighboring segments into account with similar color or reliable depth.

Before the first iteration the depth values are initialized in a random way. During the iterations the candidate depth with the minimal value of the match penalty is taken as the new estimate. The iteration procedure is halted as soon as all the depths of the segments have converged to their final value; this typically takes about four iterations. An estimate of the depth accuracy (expressions (2) and (3)) is obtained by storing the results for the tested depth candidates.

### 2.4 Segmentation

Segmentation groups pixels into a number of (4-connected) segments, such that every pixel is part of exactly one segment. For our DSFM algorithm, membership of a pixel to a segment should be decided based on depth. However, before the matching we do not know the depth yet, but we assume that depth discontinuities coincide with color discontinuities. Since depth discontinuities can only occur at

segment boundaries (because of the matching process), our main requirement is that color discontinuities should coincide with segment boundaries, and that the positioning of these boundaries is *pixel-accurate*. Image segmentation methods can be in general divided in feature-based and region-based methods (see [14] for an overview of state-of-the-art techniques). The aim of feature-based methods is to detect and then combine features (edges, gradients, ...), to obtain structures. Region-based methods use region measures (texture, color, ...) to find zones where certain properties are (nearly) constant. In this paper, we use for stability reasons a region-based method, similar to a watershed method [20,21].



**Fig. 2.** Illustration of the hierarchical segmentation. (a),(b),(c): The resulting cores at three different levels of  $p$ :  $p_0, p_1, p_2$ . (d) The tree structure arising from the core segments. (e) The pruned tree. Note that different parts of the tree have a different value for  $p$ . (f) The final cores. (g) The segmentation after growing the cores.

Many segmentation techniques suffer from the fact that there is no clear connection between the parameters specified by the user (commonly some kind of threshold  $p$  on the level of homogeneity) and the final segmentation. For our application, a priori we do not know the complexity nor content of the scene, and it is hard to specify this threshold  $p$  in advance. Furthermore, to accommodate for variations in the image, the threshold should be region-dependent. This would increase the amount of freedom in the choice of the parameters, which is undesired. Therefore we study the behavior of the segmentation when  $p$  is increased continuously. For a given value of  $p$ , let us denote a segmentation by  $\mathcal{S}(p)$ , where

$$\mathcal{S}(p) = \{S_i(p), i = 1, \dots, S\}. \quad (4)$$

$S_i$  is the  $i^{\text{th}}$  segment in the image and  $S$  is the number of segments. The parameter  $p$  can be any kind of threshold. We define  $C_i$  as the *core* of segment  $i$ , such

that  $C_i(p) \subset S_i(p)$ . For our hierarchical approach we now require that

$$\forall i \quad \forall q \geq p \quad \exists j \text{ such that } C_i(p) \subset C_j(q). \quad (5)$$

This requirement means that we can build a tree structure for the cores. If we take the cores as homogeneous regions in the image, they form the inner parts of the final segments (hence the name ‘core’).

Our segmentation algorithm now consists of the following steps:

1. A homogeneity criterium satisfying expression (5) is defined to decide whether a pixel should be part of a core at homogeneity level (threshold)  $p$ .
2. We generate a family of cores by letting this threshold vary ( $p = p_0, p_1, \dots, p_N$ ). For each value of  $p$ , we compute the resulting cores.
3. A tree structure can be built for the cores, due to requirement (5). If  $p$  increases, cores grow larger, and cores which are separated for a low value for  $p$  might merge at a higher level of  $p$  (see figure 2).
4. To create a segmentation, the tree is *pruned*. We define a binary *validity constraint*  $V(C_i, m)$ , which indicates whether a core  $C_i$  satisfies some desired property. It has one free parameter  $m$ . For instance,  $V(C_i, m)$  may check whether core  $C_i$  has a color variance smaller than  $m$ . The tree is pruned such that every core satisfies the validity constraint. Note that we have in this manner replaced the original threshold  $p$  with an alternative threshold  $m$ . Pruning the tree such that each core satisfies  $V(C_i, m)$  is equivalent to setting  $p$  differently in different regions of the image. However, where in general a region-dependent threshold increases the amount of freedom in the parameter choice, here it is a *result* of a fixed requirement. Moreover, a clear requirement on the segmentation (such as, say, the color variance of a segment) can not be expressed directly in terms of an initial threshold  $p$ , but can now be specified through  $V(C_i, m)$ .
5. The segments are grown from the cores by assigning non-core pixels to a core. The growing is based on the color gradient. The distance within the cores is zero by definition, for a non-assigned pixel the distance to the cores is calculated as the sum of the absolute value of the color difference on any path connecting the pixel with the core, using a distance transform algorithm. The pixel is then assigned to the core for which this distance is minimal. This means that if there is a color discontinuity between two cores, the segment border is coincident with this discontinuity.

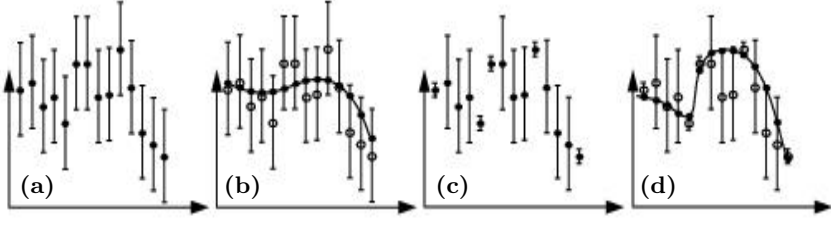
A simple illustration of this concept is shown in figure 2.

## 2.5 Relaxation: Using Depth Accuracy for Post Processing

The accuracy of the depth estimate depends on the amount of information available in the region of interest, and as a result noisy depth values may occur in places of low texture. The accuracy also depends on the camera motion: for instance, in two images with purely horizontal camera displacement, a texture consisting of horizontal stripes gives a large depth uncertainty.

In this section, we present a method to improve the quality for depth maps. The method is designed to correct outliers and other noise, but to preserve





**Fig. 3.** Illustration of the relaxation algorithm for a 1-D signal. **(a)** The initial signal  $\{d_i\}$  (black circles). **(b)** The smooth signal  $\tilde{d}$  is given by the solid line and the black circles; the original signal  $d_i$  is drawn as open circles. Since the error bounds are large, the final  $\tilde{d}_i$  can deviate arbitrarily far from the initial  $d_i$ . **(c)** Now some  $d_j$  have small error bounds. **(d)** In regions with large depth uncertainties, the resulting  $\tilde{d}_i$  values are smooth; however, if the error intervals are small the smoothing is restricted by the error bounds and depth discontinuities are preserved.

real depth discontinuities. Conventional regularization methods typically add a term  $\lambda E_s$  to the match penalty which accounts for smoothness. If  $\lambda = 0$ , a rather instable solution results; for larger  $\lambda$  the depth map is smoother but it may deviate more from the actual values. Also, depth discontinuities may get washed out. Anisotropic operators  $E_s$  do not smooth across discontinuities, so that edges are preserved. However, their computation is expensive. Moreover, anisotropic smoothing suffers from a chicken-and-egg problem: The smoothing operator assumes to have detected a large gradient, and preserves this large gradient. So effectively, it only smooths those areas that already are in some sense ‘smooth’ (i.e., have low gradient values). But if a noise feature happens to have a (locally) coherent nature, it may get mistaken for a gradient by the anisotropic smoother, and hence it is not removed.

If there is no further information available, it is not clear how to do better than anisotropic smoothing. But in the case of segment-based depth estimation, there are two additional information channels: the depth uncertainty (expressions (2) and (3)), and the assumption that depth discontinuities coincide with color discontinuities. Here we discuss a smoothing algorithm for depth maps that takes advantage of these two additional sources of information. It is based on an extension of an iterative relaxation method, which has been studied earlier in computer vision [17,10]. Applications and the study of some special cases have been published in [19]. Let  $N_i$  be the set of indices  $j$  of  $d_i$ ’s neighbors. Then we set

$$\tilde{d}_i \leftarrow \tilde{d}_i + \alpha \frac{|N_i|}{|N_i| + 1} \left( \left( \frac{1}{|N_i|} \sum_{j \in N_i} \tilde{d}_j \right) - \tilde{d}_i \right), \text{ for all } i; \quad (6)$$

$$\tilde{d}_j \leftarrow \tilde{d}_j - \alpha \frac{1}{|N_i| + 1} \left( \left( \frac{1}{|N_i|} \sum_{j \in N_i} \tilde{d}_j \right) - \tilde{d}_i \right), \text{ for } j \in N_i, \quad (7)$$

where all assignments occur simultaneously.

Consider the case (figure 3c) where for some indices  $j$ , the initial signal  $d_j$  has a small error interval. Then, clearly, the value of  $\tilde{d}$  should not be allowed outside this interval. This can be achieved during the iteration by clamping any new value  $\tilde{d}_j$  against the borders of its error interval. Given for each pixel its estimated depth  $d(\mathbf{x})$  and associated uncertainty  $(d_l(\mathbf{x}), d_u(\mathbf{x}))$ , we search for a smooth function  $\tilde{d}(\mathbf{x})$  such that

$$d_l(\mathbf{x}) \leq \tilde{d}(\mathbf{x}) \leq d_u(\mathbf{x}), \quad \forall \mathbf{x}. \quad (8)$$

In this way, depth discontinuities survive smoothing if there is sufficient evidence for a discontinuity in terms of the uncertainties. For clarity, we show the concept of the relaxation algorithm on a 1-D signal in figure 3.

For a regular 2-D grid (blocks), where  $|N_i| = 4$  for all  $i$ , it can be verified that the process converges (if it converges) to a state where the resulting 2-D arrangement of samples lies on a bi-cubic polynomial surface, and hence is optimally smooth, provided that no clamping against error intervals occurs. For an arbitrary topology we cannot give an analytic description for the converged state. Still, since the filter is a direct generalization of the 2-D regular tessellation-case, we may expect that again a smooth surface results.

The second information channel to constrain the relaxation is the color. Consistent with the earlier assumption that depth discontinuities and color discontinuities coincide, we propose to take only neighboring segments into account in equations (6) and (7) with small color differences with the actual segment. This preserves discontinuities even at places where the error bounds are too large to clamp the new  $d$ -values.

This entire procedure is iterated over all segments until the updates are sufficiently small. This typically takes 3-4 iterations.

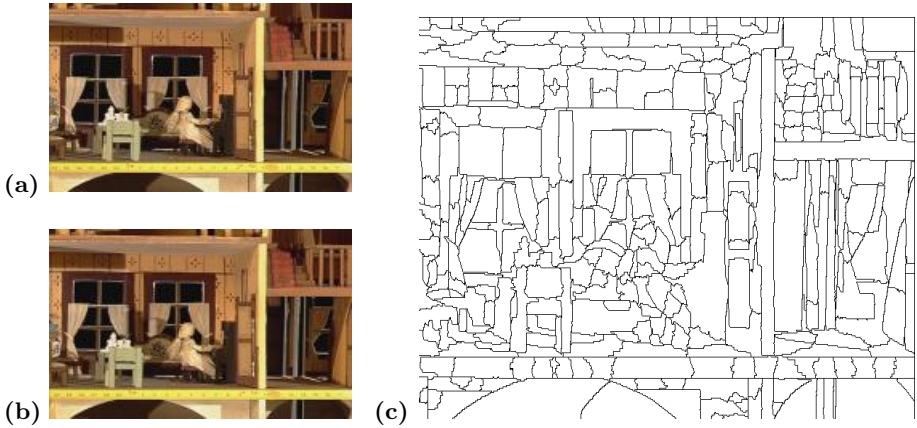
### 3 Results

We have applied our algorithm to numerous different scenes. In this section we show some typical results, and highlight some of the characteristics of the dense structure-from-motion algorithm.

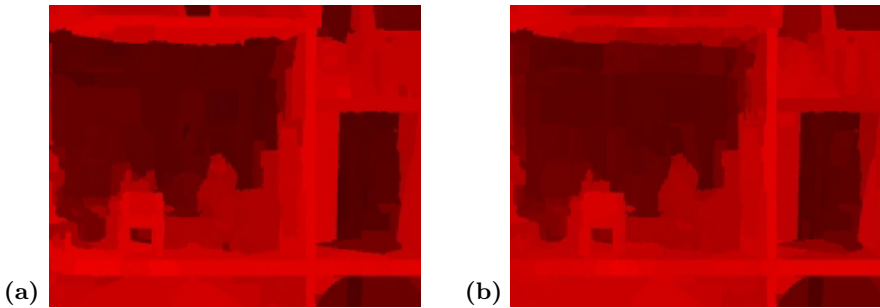
The first example applies DSFM to a scene depicting a doll house. In this case, the camera trajectory was known, and consisted of a horizontal translation. The typical parallax between subsequent images in the sequence is of the order of 5 pixels. In figure 4a-b we show the images for which we do DSFM. The segmentation is shown in figure 4c. All pixels which satisfy

$$\max(|C(x', y') - C(x, y)|, |x - x'| \leq 1, |y - y'| \leq 1) \leq p \quad (9)$$

are taken as core pixels on level  $p$ . Here,  $C$  is the color of a pixel. For noise reduction reasons, in the segmentation step a low-pass filtered version of the image is used. It can easily be seen that cores based on equation (9) satisfy the tree criterion specified in equation (5), since pixels which are core pixels at level  $p$  are also core pixels at all higher levels. As a validity constraint, we require that the color variation in a core should be smaller than 8% of the dynamic color range.



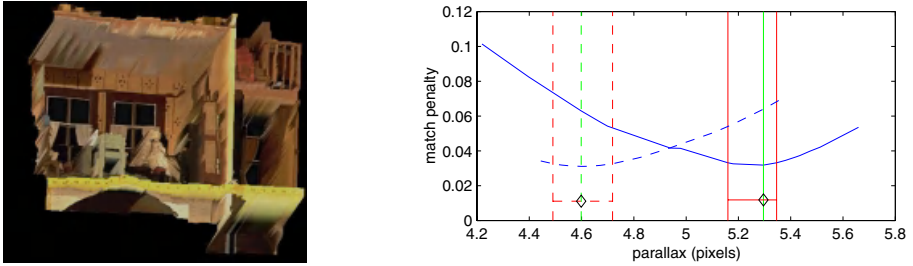
**Fig. 4.** (a), (b) Two subsequent frames from the doll house sequence. The parallax between the frames is on the order of 5 pixels. (c) The segmentation of the frame in (a). There are segment boundaries at all depth discontinuities; some oversegmentation is visible.



**Fig. 5.** (a) Raw depth map for the doll house. Bright color means small depth, dark color means large depth. Some outliers are visible, for instance near the middle of the upper boundary, and below the doll. (b) Depth map after the relaxation procedure. The depth map is smoothed at the location of the discontinuities, but the depth discontinuities between doll and background and between the legs of the chair are not affected. The color scale is the same as in (a).

We have pruned the tree by removing all cores with a higher color variation. It can be clearly seen that there is some form of oversegmentation. However, for depth recovery this is not a problem: an object which is split into multiple segments can still have a single depth. After applying the DSFM algorithm, the raw depth map is shown in figure 5a. The overall quality is quite good, however, some depth outliers are present at the top of the image in the middle, and below the doll. Depth discontinuities around the doll and on the edge of the piano are recovered well. The resulting depth map after application of the relaxation algorithm is shown in figure 5b. The ‘real’ depth discontinuities have not been affected by the smoothing, whereas the outliers have been removed. The depth discontinuities around the tea pot on the table have even been sharpened, be-

cause the uncertainty of those segments allows an assignment of the background depth to those segments. To indicate the accuracy of the matching, we show error curves for segments containing the head of the doll and the curtain adjacent to it in figure 6. Here, the value of the match penalty as a function of segment displacement (parallax, and thus depth) is displayed. There is a clear and pronounced global minimum for both segments, and the depth difference between the two segments is clearly visible. Note that it is very well possible to find the parallax with a subpixel accuracy. This error curve is typical for pairs of images containing sufficient texture and having a good camera calibration. From our experience, parallax differences up to  $1/8^{th}$  of a pixel can be detected reliably.



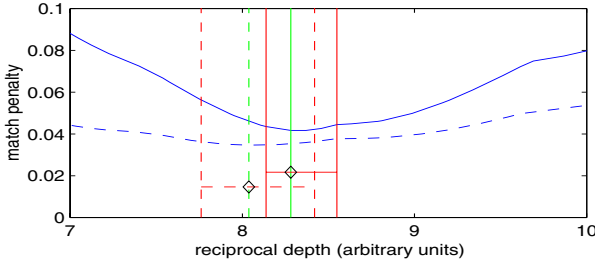
**Fig. 6.** (a) Screen shot from the 3-D reconstruction of the doll house, where the viewpoint has been moved to the lower left. (b) Error curve for the doll's head (solid line) and the right curtain of the right window (dashed line) as a function of the parallax (in this case corresponding to reciprocal depth). The optimal value and error bounds surrounding it are denoted by the vertical bars. A pronounced global minimum can be discerned, and it can be estimated to high sub-pixel accuracy (up to  $1/8^{th}$  of a pixel).



**Fig. 7.** (a), (b) Frames of the Dionysos sequence. (c) The resulting depth map after applying the DSFM algorithm. A pixel-accurate distinction between foreground and background is made; however, also depth differences within the statue are reconstructed (see also figure 8).

The second example is based on a hand-held video sequence of a statue of Dionysos (see figure 7a-b). In this case, the camera position and orientation were not known, but have been estimated with a camera calibration algorithm.

In this case, the camera motion consisted of both a rotation and a translation. The final depth map is shown in figure 7c. We see a pixel-accurate distinction between foreground and background. However, to show that also details in the face can be recovered, we have plotted error curves of the segments containing the nose and the eyes in figure 8. The very small parallax difference (of the order of 0.25 pixels) has been recovered; for comparison, the background is at a depth of 3.4 units, and maximal parallax differences are on the order of 7 pixels. This means that DSFM does not only have a high accuracy, but also a high dynamic range.



**Fig. 8.** Error curve for the nose of Dionysos (solid line) and the right eye (dashed line). The optimum values and its error bounds are denoted by the vertical bars. Even the small depth difference between eyes and nose (corresponding to a difference in segment motion of approximately 0.25 pixels) can be discerned. For comparison, the background has a reciprocal depth of approximately 3.4 units.

## 4 Handling Occlusions in Depth Estimation

In the computation of the match penalty (1), we estimate for all pixels in a segment their location in the next image, based on the proposed depth, and compute the absolute difference between the colors at the positions in the first ( $I_0$ ) and second ( $I_1$ ) image. However, it may occur that the point  $X$  corresponding to a pixel  $\mathbf{x}$  which is visible in the first image, is not visible in the second image, because another part of the scene occludes it (for instance due to camera movement, see figure 9). In that case, the contribution of this pixel to the match penalty has no relation to the depth of the segment. Therefore, we propose a modification of the match penalty, where we only sum over those pixels of a segment which are also visible in the second image:

$$E(d) = \sum_{\mathbf{x} \in S} v(\mathbf{x}) |I_1(\mathbf{x} + \Delta \mathbf{x}(d)) - I_0(\mathbf{x})|, \quad (10)$$

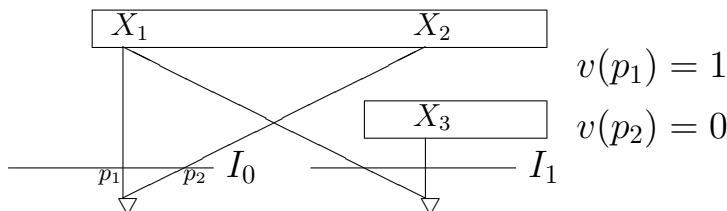
where  $v(\mathbf{x})$  is defined as

$$v(\mathbf{x}) = \begin{cases} 1, & \text{if } X, \text{ corresponding to } \mathbf{x} \text{ in } I_0, \text{ is visible in } I_1 \\ 0, & \text{if } X, \text{ corresponding to } \mathbf{x} \text{ in } I_0, \text{ is not visible in } I_1 \end{cases} \quad (11)$$

The function  $v(\mathbf{x})$  is called the *visibility map*. In order to compute the visibility map, the following steps have to be carried out for each pixel  $\mathbf{x}$ :

1. Set  $v(\mathbf{x}) = 1$  everywhere.
2. Estimation of the depth for the segment containing  $\mathbf{x}$  in  $I_0$  using expression (10).
3. Computation of the point  $\mathbf{X}$  in 3-D space which projects onto  $\mathbf{x}$ .
4. Computation of the resulting location and depth of  $\mathbf{X}$  in  $I_1$ .
5. Set  $v(\mathbf{x}) = 1$  if this is either the only pixel at that location in  $I_1$ , or the pixel with the smallest depth of all pixels which end up at that location. Else set  $v(\mathbf{x}) = 0$ .
6. If the process has not converged yet, go to step 2.

The correct handling of occlusion removes a source of distortion in the computation of the match penalty, resulting in a more accurate depth estimate.



**Fig. 9.** Illustration of the visibility map concept. The location  $X_1$ , corresponding to  $p_1$ , is visible in the first (left) and second (right) image. The location  $X_2$ , however, is only visible in the first image (as pixel  $p_2$ ), but not in the right image, since it is occluded by the shape containing  $X_3$ .

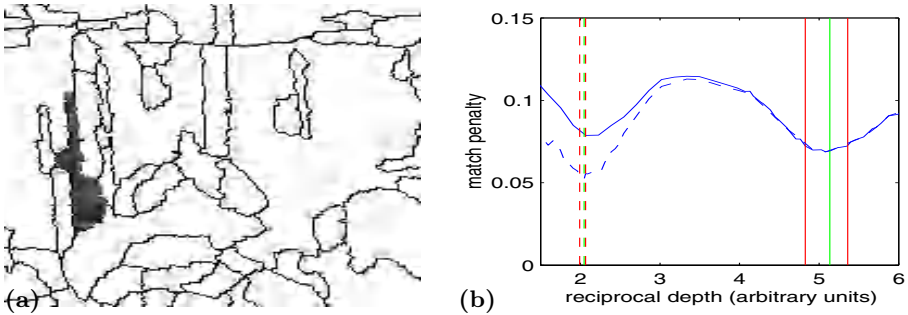
To show the effect of occlusion handling, we have applied DSFM to a stereo sequence depicting a farm scene. Due to the large camera baseline, there are large occlusion areas. We focus especially at the area around the child's head, where parts of the fence are occluded (see figure 10). The segmentation of that area is shown in figure 11a. Figure 11b shows the error curve for the shaded segment (one of the posts of the fence). Without taking occlusion into account, there are two pronounced minima, of which - of course - only one corresponds to the real depth. The other minimum corresponds to a parallax where this post is mapped onto another post, resulting in a wrong depth (see figure 12a). With the visibility map, the adapted error curve still has two minima, but the global minimum is now much lower (see figure 11b). The final depth map is shown in figure 12b.

## 5 Conclusion and Discussion

We have presented a dense structure-from-motion algorithm (DSFM) based on segment matching. Its major advantage is its robustness in providing dense depth maps, since region instead of point correspondences are found between images. After segmentation, the depth and depth accuracy of each segment is estimated by minimizing a match penalty. Post processing removes outliers and does a depth-discontinuity preserving smoothing of the depth field. Occlusions are explicitly taken into account through an adaptation of the match penalty. In con-



**Fig. 10.** Part of the left and right image of a frame of the ‘Farm’ stereo sequence. Parts of the posts of the fence in the left image are occluded by the child’s head in the right image, which may hamper the depth estimation.

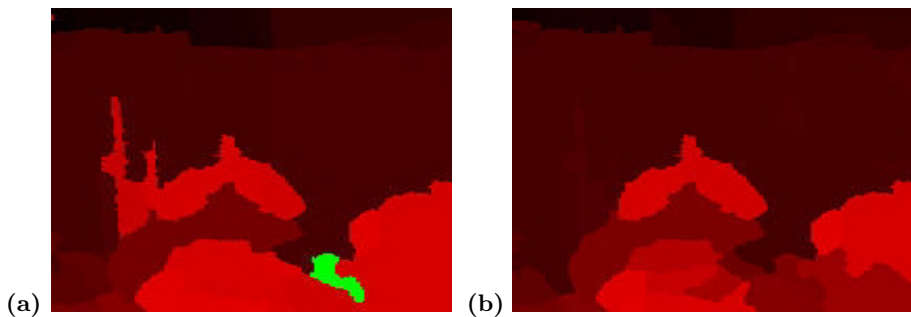


**Fig. 11.** (a) Segmentation of the left image of figure 10. The shaded segment is partly occluded in the right image. (b) Error curves for the shaded segment. The solid curve is the error curve for the basic match penalty (1). The minimum corresponding to an incorrect depth is lower than the correct minimum due to occlusion effects. If the occlusion effect is taken into account with expression (10) (the dashed line), the minimum corresponding to the correct depth is lower and hence taken.

trast to stereovision methods, this method can be used for arbitrary camera motions (not only horizontal translation).

In the examples, we have demonstrated the robustness and accuracy of the algorithm. A depth accuracy corresponding to motion differences of  $1/8^{th}$  of a pixel is obtainable, also in low-textured regions. DSFM is especially suited for the case of small motions and small motion differences. Because regions are matched, the correspondence problem typically encountered in feature-based methods can to a large extent be avoided. Since segments are relatively large and hence contain more information, DSFM is more robust than feature-based approaches. Increasing the segment size does not decrease the resolution, however, since the segment borders are aligned with color, and hence depth, discontinuities.

Although the DSFM algorithm gives a quite satisfactory depth estimation for (static or stereo) video sequences, there is still some room for improvement, e.g. in the parameterization of the depth within a segment. A challenging re-



**Fig. 12.** (a) Depth map without taking occlusion into account. Parts of the fence are too close to us. (b) Depth map when the visibility is taken into account. The fence now has the correct depth.

search subject is still the estimation of depth from dynamic video sequences (the “independent motion” problem).

DSFM has several possible applications, from scene modeling to object extraction and tracking. In the domain of digital television, 3-D reconstruction is a necessary element to convert existing 2-D video for future 3D TV applications. Depth reconstruction algorithms, if implemented in hardware, may be efficient enough to do the computation in real time. The present algorithm is very regular, and due to the candidate-based minimization algorithm also very efficient. A slight change of the algorithm allows for segment-based motion estimation, which has prominent applications in scan-rate conversion and MPEG encoding.

**Acknowledgement.** We would like to thank Marc Pollefeys (KU Leuven) for providing the Dionysos sequence. Financial support from the ITEA BEYOND project is gratefully acknowledged.

## References

1. M. Accame, F.G.B. De Natale, and D. Giusto. Hierarchical block matching for disparity estimation in stereo sequences. In *ICIP95*, pages 374–377, 1995.
2. G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. PAMI*, 7:384–401, 1985.
3. S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. PAMI*, 2:333–340, 1980.
4. H.A. Beyer. Some aspects of the geometric calibration of CCD cameras. In *ISPRS Intercomm. Conf. on Fast Processing of Photogrammetric Data*, Interlaken, 1987.
5. G. de Haan and P. Biezen. Sub-pixel motion estimation with 3D recursive search block matching. *Signal Processing: Image Communication*, 6:229–239, 1994.
6. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
7. MPEG-4 Video group ISO WG11. MPEG-4 overview (Maui Version). Technical Report ISO/IEC/JTC1/SC29/WG11 N3156, ISO, 1999.
8. J.R. Jain and A.K. Jain. Displacement measurement and its application in inter-frame image coding. *IEEE Trans. Comm.*, 29:1799–1808, 1981.



9. Tony Jebara, Ali Azarbayejani, and Alex Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, pages 66–84, May 1999.
10. J.L. Mallet. Discrete smooth interpolation in geometric modelling. *Computer Aided Design*, 24:178–191, 1992.
11. M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Metric 3D surface reconstruction from uncalibrated image sequences. In *Proc. SMILE Workshop (post-ECCV'98)*, LNCS 1506, pages 138–153. Springer-Verlag, 1998.
12. P.A. Redert, E.A. Hendriks, and J. Biemond. Correspondence estimation in image pairs. *IEEE Signal Processing Magazine*, 16:29–46, 1999.
13. R. Rodrigues, K. van Overveld, and P. Wilinski. Depth reconstruction based on irregular patches. In *Proc. EPCG no 9*, Marinha Grande, Portugal, 1999.
14. P. Salembier and F. Marques. Region-based representations of image and video: segmentation for multimedia services. *IEEE Trans. CSVT*, 9:1147–1169, 1999.
15. S. Soatto and P. Perona. Reducing “Structure from Motion”: A general framework for dynamic vision. part 1: Modeling. *IEEE Trans. PAMI*, 20:933–942, 1998.
16. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proc. ICCV*, pages 532–539, Vancouver, Canada, 2001.
17. D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. PAMI*, 10:417–438, 1988.
18. R.Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV camera lenses. *IEEE Journal on Robotics and Automation*, RA-3:323–344, 1987.
19. C.W.A.M. van Overveld. The application of relaxation and optimisation methods in computer aided geometric design. In B. Özgüç and V. Akman, editors, *Proc. of First Bilent Comp. Graphics Conf.*, pages 161–180, Ankara, Turkey, 1993.
20. L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. PAMI*, 13:583–598, 1991.
21. D. Wang. Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Trans. CSVT*, 8:539–546, 1998.
22. A. Yezzi and S. Soatto. Stereoscopic segmentation. In *Proc. ICCV*, pages 59–66, Vancouver, Canada, 2001.

# Maximizing Rigidity: Optimal Matching under Scaled-Orthography

João Maciel<sup>1</sup> and João Costeira<sup>1</sup>

Instituto de Sistemas e Robotica, Instituto Superior Tecnico, Portugal  
`{maciel,jpc}@isr.ist.utl.pt`

**Abstract.** Establishing point correspondences between images is a key step for 3D-shape computation. Nevertheless, shape extraction and point correspondence are treated, usually, as two different computational processes. We propose a new method for solving the correspondence problem between points of a fully uncalibrated scaled-orthographic image sequence. Among all possible point selections and permutations, our method chooses the one that minimizes the fourth singular value of the observation matrix in the factorization method. This way, correspondences are set such that shape and motion computation are optimal. Furthermore, we show this is an optimal criterion under bounded noise conditions.

Also, our formulation takes feature selection and outlier rejection into account, in a compact and integrated way. The resulting combinatorial problem is cast as a concave minimization problem that can be efficiently solved. Experiments show the practical validity of the assumptions and the overall performance of the method.

## 1 Introduction

Extracting 3D-shape information from images is one of the most important capabilities of computer vision systems. In general, computing 3D coordinates from 2D images requires that projections of the same physical world point in two or more images are put to correspondence.

Shape extraction and point correspondence are treated, usually, as two different computational processes. Quite often, the assumptions and models used to match image points are unrelated to those used to estimate their 3D coordinates. On one hand, shape estimation algorithms usually require known correspondences [21], solving for the unknown shape and motion. On the other hand, image feature matching algorithms often disregard the 3D estimation process, requiring knowledge of camera parameters [15] or use other specific assumptions [18]. Furthermore, while matching algorithms tend to rely on local information — *e.g.* brightness [10,18] — shape computation algorithms [23,20,21, 19] rely on rigidity as a global scene attribute. These methods recover the rigid camera motion and object shape that best fit the data.

We present a new method that links shape computation to image feature matching by choosing the point correspondences that maximize a single global criterion — rigidity. In other words, correspondences are set such that they optimize one criterion for which we know how to compute the optimal solution for shape and motion. Also, our formulation takes feature selection and outlier rejection into account, in a compact and integrated way. This is made possible by formulating the matching process as an integer programming problem where a polynomial function — representing rigidity deviation — is minimized over the whole set of possible point correspondences. Combinatorial explosion is avoided by relaxing to continuous domain.

### 1.1 Previous Work

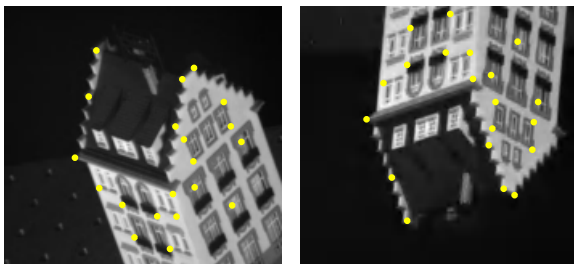
Rigidity has been used before in the correspondence framework [1,20,15,18], though used in conjunction with other assumptions about the scene or camera. The work of [22] is an example of a successful use of global geometrical reasoning to devise a pruning mechanism that is able to perform outlier rejection in sets of previously matched features. Optimality is guaranteed in a statistical sense.

Other approaches use a minimal set of correspondences which help computing the correspondences for the complete set of features [19]. This is related to using prior knowledge about camera geometry in order to impose epipolar constraints [24] or multi-view motion constraints.

Finally the approach of [4] is an example where matching and 3D reconstruction are deeply related. Correspondences, shape and motion are simultaneously optimized by an Expectation Maximization algorithm. Spurious features are not explicitly taken into account.

## 2 Maximizing Rigidity: Problem Statement

Consider the images of a static scene shown in Figure 1<sup>1</sup>. Segment  $p_1$  feature-



**Fig. 1.** Two images from the Hotel sequence, with extracted corners.

<sup>1</sup> Data was provided by the Modeling by Video group in the Robotics Institute, CMU

points on the first image and  $p_2 > p_1$  on the second — the white dots — arrange their image coordinates  $u_p$  and  $v_p$  in two matrices  $\mathbf{X}$  and  $\mathbf{Y}$  :

$$\mathbf{X} = \begin{bmatrix} u_1^1 & v_1^1 \\ \vdots & \vdots \\ u_{p_1}^1 & v_{p_1}^1 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} u_1^2 & v_1^2 \\ \vdots & \vdots \\ u_{p_2}^2 & v_{p_2}^2 \end{bmatrix} \quad (1)$$

Some of these features are projections of the same 3D points. We wish to recover their 3D coordinates assuming no prior knowledge except that the object is rigid and the camera is scaled-orthographic. To do so a selection mechanism should arrange some of the observed features in a matrix of centered measurements  $\mathbf{W}$ , as in [21]<sup>2</sup>. Matched features must lie in the same row of  $\mathbf{W}$ . Note that no local image assumptions are made, no calibration information is known and no constraints in the disparity field are used.

Without noise, matrix  $\mathbf{W}$  is, at most, rank 3, even with scale changes. We propose to solve this problem by searching for the correspondences that best generate a rank-three  $\mathbf{W}$  matrix. This is only a necessary condition for rigidity, so multiple solutions minimize this criterion<sup>3</sup>. Within the optimization framework used — Section 4 — retrieving solutions of similar cost is trivial. The number of solutions decreases dramatically with increasing number of frames, or decreasing number of rejections. In any case, for orthographic cameras we can choose the only solution leading to an orthonormal motion matrix — the rigid solution.

With noisy measurements,  $\mathbf{W}$  is always full-rank, so we must be able to answer the following questions:

1. Is it possible to generalize the rank criterion in the presence of noise?
2. Is there any procedure to search for the best solution of this problem with reasonable time complexity?

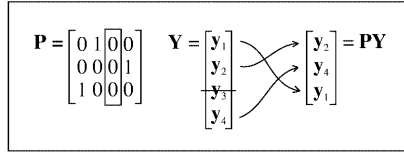
This paper tries to give a positive answer to these questions, by formulating the correspondence problem as an optimization problem with polynomial cost function.

### 3 Optimal Matching

For the sake of simplicity, we start with the two-image case. Our goal is to determine a special permutation matrix  $\mathbf{P}^* \in \mathcal{P}_p^c(p_1, p_2)$ , such that  $\mathbf{X}$  and  $\mathbf{P}^*\mathbf{Y}$  have corresponding features in the same rows.  $\mathbf{P}$  is constrained to  $\mathcal{P}_p^c(p_1, p_2)$ , the set of  $p_1 \times p_2$  *columnwise partial permutation matrices* ( $p_p$ -matrices). A  $p_p$ -matrix is a permutation matrix with added columns of zeros. The optimal  $\mathbf{P}^*$  is a zero-one variable that selects and sorts some rows of  $\mathbf{Y}$ , putting them to correspondence with the rows of  $\mathbf{X}$ . Each entry  $\mathbf{P}_{i,j}$  when set to 1 indicates that features  $\mathbf{X}_i$ . (row  $i$  of  $\mathbf{X}$ ) and  $\mathbf{Y}_j$ . (row  $j$  of  $\mathbf{Y}$ ) are put to correspondence. Figure 2 shows an example. Such a matrix guarantees robustness in the presence

<sup>2</sup> Our  $\mathbf{W}$  corresponds to their  $\mathbf{W}^\top$

<sup>3</sup> Without noise, any object deforming according to a linear transformation in 3D space generates rank-3 projections.



**Fig. 2.** A partial permutation matrix representing a particular selection and permutation of rows of  $\mathbf{Y}$ .

of outliers by allowing some features to be "unmatched". It encodes one way of grouping the measurements in a matrix of centered observations

$$\mathbf{W}_{\mathbf{P}} = \mathbf{W}(\mathbf{P}) = [\mathbf{C}\mathbf{X} \mid \mathbf{C}\mathbf{P}\mathbf{Y}]_{[p_1 \times 4]} \quad (2)$$

Matrix  $\mathbf{C}_{[p_1 \times p_1]} = \mathbf{I} - \frac{1}{p_1} \mathbf{1}_{[p_1 \times p_1]}$  normalizes the columns of the observation matrices to zero mean. The correct  $p_p$ -matrix  $\mathbf{P}^*$  generates  $\mathbf{W}^*$  which is the measurement matrix of Tomasi-Kanade [21]. With noise-free measurements, non-degenerate full 3D objects produce rank-3 observation matrices  $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^*)$  whenever  $\mathbf{P}^*$  is the correct partial permutation. A single mismatched point will generate two columns of  $\mathbf{W}_{\mathbf{P}}$  that are outside the original 3-dimensional column-space — also called the shape-space. This makes  $\mathbf{W}_{\mathbf{P}}$  full-rank even in the absence of noise. In conclusion, the noise-free correspondence problem can be stated as Problem 1

$$\begin{aligned} \text{Problem 1} \quad & \mathbf{P}^* = \arg \min_{\mathbf{P}} \quad \text{rank}(\mathbf{W}_{\mathbf{P}}) \\ & \text{s.t.} \quad \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

### 3.1 Approximate Rank

Consider now the case of noisy measurements. The observation matrix includes two additive noise terms  $\mathbf{E}^X$  and  $\mathbf{E}^Y$

$$\mathbf{W}'_{\mathbf{P}} = [\mathbf{C}(\mathbf{X} + \mathbf{E}^X) \mid \mathbf{C}\mathbf{P}(\mathbf{Y} + \mathbf{E}^Y)] \quad (3)$$

The factorization method [21] provides an efficient way of finding the best rigid interpretation of the observations. It deals with noise by disregarding all but the largest 3 singular values of  $\mathbf{W}'_{\mathbf{P}}$ . The approximation error is measured by  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$ , the fourth singular value of  $\mathbf{W}'_{\mathbf{P}}$ . If we use  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  as the generalization of the criterion of Problem 1, then we should search for the correspondence that minimizes the approximation error made by the factorization method. When noise is present, we formulate the correspondence problem as follows

$$\begin{aligned} \text{Problem 2} \quad & \mathbf{P}^* = \arg \min_{\mathbf{P}} \quad \lambda_4(\mathbf{W}'_{\mathbf{P}}) \\ & \text{s.t.} \quad \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

In fact, for bounded noise, the solution to Problem 2 is again  $\mathbf{P}^*$ , which is the solution to Problem 1 when no noise is present. This is precisely stated in Proposition 1

**Proposition 1** *For every nondegenerate rigid object observed by a scaled orthographic camera it is possible to find a scalar  $\epsilon > 0$  such that if  $|\mathbf{E}_{(i,j)}^X| < \epsilon$  and  $|\mathbf{E}_{(i,j)}^Y| < \epsilon \forall i, j$  then the solution of Problem 2 is exactly  $\mathbf{P}^*$ , which is the solution of Problem 1 in the absence of noise.*

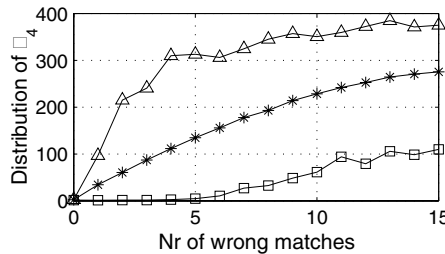
**Proof:** In the absence of noise, the correct  $\mathbf{P}^*$  matrix generates  $\mathbf{W}'^*$ , and  $\text{rank}(\mathbf{W}'^*) = 3 \Leftrightarrow \lambda_4(\mathbf{W}^*) = 0$ . Assuming that Problem 2 has a single nondegenerate solution in the absence of noise, then there is a nonzero difference between the best and second best values of its cost. That is to say

$$\exists \delta > 0 : \lambda_4(\mathbf{W}'^*) + \delta < \lambda_4(\mathbf{W}'_{\mathbf{P}}), \forall \mathbf{P} \neq \mathbf{P}^* \quad (4)$$

Since  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  is a continuous function of the entries of  $\mathbf{W}'_{\mathbf{P}}$  then this is also a continuous function of the entries of  $\mathbf{E}^X$  and  $\mathbf{E}^Y$ . By definition of continuity,  $\exists \epsilon > 0$  such that if  $|\mathbf{E}_{(i,j)}^X| < \epsilon$  and  $|\mathbf{E}_{(i,j)}^Y| < \epsilon \forall i, j$  then Equation 4 still holds. This guarantees that, under these noise constraints,  $\mathbf{P}^*$  is still the optimal solution to Problem 2.

Our proof for Proposition 1 does not present a constructive way to compute  $\epsilon$ , so we did an empirical evaluation about the practical validity of this noise bound.

We segmented a set of points on two images. For each number of wrong matches, a set of randomly generated  $\mathbf{P}$  matrices were used to compute  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$ . Figure 3 shows its statistics. The global minimum is reached for the correct corre-



**Fig. 3.** Minimum, maximum and average of  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  as functions of the number of mismatches.

spondence  $\mathbf{P}^*$ , even with noisy feature locations. This shows that the bound  $\epsilon$  is realistic. It also validates  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  as a practical criterion. Finally note that the average values of  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  increase monotonously with number of mismatches. This means that suboptimal solutions with objective values close to optimal will, on average, have a small number of mismatches. This is most useful to devise a stopping criterion for our search algorithm (section 4).

### 3.2 Explicit Polynomial Cost Function

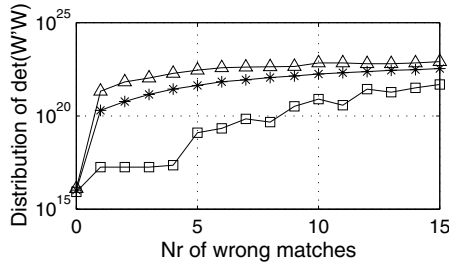
We show here that the 4<sup>th</sup> singular value of  $\mathbf{W}$  (cost in Problem 2) has an equivalent explicit fourth-order polynomial cost function:

$$J(\mathbf{P}) = \omega^\top \mathbf{q}^{[4]} \quad (5)$$

Here,  $\omega$  is a vector independent of  $\mathbf{q} = \text{vec}(\mathbf{P})$ ,  $\text{vec}()$  is the vectorization operator<sup>4</sup> and  $\mathbf{q}^{[4]} = \mathbf{q} \otimes \mathbf{q} \otimes \mathbf{q} \otimes \mathbf{q}$  (symbol  $\otimes$  stands for Kronecker product). Considering the bounded noise assumption once more, we can use the same sort of reasoning used to prove Proposition 1 and show that the original cost function of Problem 2 can be changed to

$$J(\mathbf{P}) = \det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}}) \quad (6)$$

In the absence of noise the result is immediate (they are both zero). The full proof of this result can be found in Appendix 1 (also in [12]). Figure 4 shows statistics of  $\det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}})$  computed for different matches of points that were segmented from real images. Once again, the global minimum is reached for the



**Fig. 4.** Minimum, maximum and average of  $\det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}})$  as functions of the number of mismatches.

correct correspondence  $\mathbf{P}^*$ , so we conclude that our new noise bound is also valid in practice. Furthermore, the global minimum is even steeper — note that the plot is shown in logarithmic scale.

Finally, keeping in mind that  $\mathbf{C}$  is symmetrical and idempotent and that  $\mathbf{C}\mathbf{X}$  is full rank, the determinant cost function of Equation 6 can be transformed into the polynomial form of Equation 5 (see appendix 2):

$$\begin{aligned}
 J(\mathbf{P}) &= \omega^\top \mathbf{q}^{[4]} \\
 \omega &= \left[ \left( \text{vec}(\Delta)^\top \mathbf{Y}^{[4]\top} \right) \otimes \left( \text{vec}(\Xi)^{[2]\top} \mathbf{I}_{p_1^4} \right) \right] \Pi \\
 \Xi &= \mathbf{C} - \mathbf{C}\mathbf{X}(\mathbf{X}^\top \mathbf{C}\mathbf{X})^{-1} \mathbf{X}^\top \mathbf{C}
 \end{aligned}$$

<sup>4</sup> Stacks the columns of a matrix as a single column vector

$$\Delta = \frac{1}{2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^{[2]}$$

$\Pi$  is a fixed  $(p_1 p_2)^4 \times (p_1 p_2)^4$  permutation such that  $\text{vec}(\mathbf{P}^{[4]}) = \Pi \text{vec}(\mathbf{P})^{[4]}$ . The important fact is that  $J(\mathbf{P})$  has a simple biquadratic structure, depending only on data matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .

### 3.3 Outline of the Complete Algorithm

The cost function of Equation 5 is ready to be formatted in a global optimization framework, where the correct permutation is the solution to the following problem:

$$\textbf{Problem 3} \quad \mathbf{P}^* = \arg \min_{\mathbf{P}} J(\mathbf{X}, \mathbf{P}\mathbf{Y}) \quad (7)$$

s.t.

$$\mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \iff \begin{cases} \mathbf{P}_{i,j} \in \{0, 1\}, \quad \forall i = 1 \dots p_1, \quad \forall j = 1 \dots p_2 & .1 \\ \sum_{i=1}^{p_1} \mathbf{P}_{i,j} \leq 1, \quad \forall j = 1 \dots p_2 & .2 \\ \sum_{j=1}^{p_2} \mathbf{P}_{i,j} = 1, \quad \forall i = 1 \dots p_1 & .3 \end{cases} \quad (8)$$

### 3.4 Reformulation with a Compact Convex Domain

Problem 3 is a brute force NP-hard integer minimization problem. In general, there is no efficient way of (optimally) solving such type of problems. Nonetheless there is a related class of optimization problems for which there are efficient, optimal algorithms. Such a class can be defined as Problem 4.

$$\textbf{Problem 4} \quad \mathbf{P}^* = \arg \min_{\mathbf{P}} J_{\epsilon}(\mathbf{X}, \mathbf{P}\mathbf{Y})$$

s.t.  $\mathbf{P} \in \mathcal{DS}_s(p_1, p_2)$

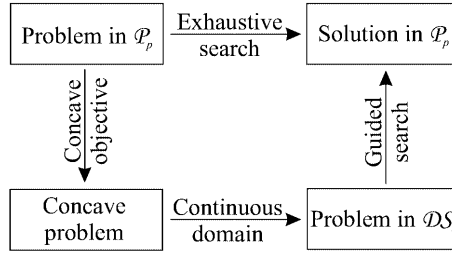
where  $J_{\epsilon}$  is a concave version of  $J$  — to be def. later, equation 10— and  $\mathcal{DS}_s(p_1, p_2)$  is the set of real  $p_1 \times p_2$  *columnwise doubly sub-stochastic matrices*. This set is the convex hull of  $\mathcal{P}_p^c(p_1, p_2)$ , constructed by dropping the zero-one condition (8.1), and replacing it with

$$\mathbf{P}_{i,j} \geq 0, \quad \forall i = 1 \dots p_1, \quad \forall j = 1 \dots p_2 \quad (9)$$

Problems 3 and 4 can be made equivalent — same global optimal — by finding an adequate concave objective function  $J_{\epsilon}$ . Also we must be sure that the vertices of  $\mathcal{DS}_s(p_1, p_2)$  are the elements of  $\mathcal{P}_p(p_1, p_2)$ . Figure 5 summarizes the whole process.

The main idea is to transform the integer optimization problem into a global continuous problem, having exactly the same solution as the original, which minimum can be found efficiently. The full process is outlined as follows:





**Fig. 5.** Efficient solution of the combinatorial problem.

1. Extract interest points and build  $\mathbf{X}$ ,  $\mathbf{Y}$  — Equation 1.
2. Use  $\mathbf{X}$ ,  $\mathbf{Y}$  to build the cost  $J(\mathbf{P})$  — Equation 5.
3. Build a concave equivalent  $J_\epsilon(\mathbf{P})$  — Equation 10,11.
4. Write  $\mathcal{S}_s^c(p_1, p_2)$  in canonical form — see section 3.6.
5. Build Problem 5 and solve it using a concave minimization algorithm — Section 4.

### 3.5 Equivalence of Problems 3 and 4

Theorem 1 states the fundamental reason for the equivalence. [7] contains its proof.

**Theorem 1** *A strictly concave function  $J : \mathcal{C} \rightarrow \mathbb{R}$  attains its global minimum over a compact convex set  $\mathcal{C} \subset \mathbb{R}^n$  at an extreme point of  $\mathcal{C}$ .*

The constraining set of a minimization problem with concave objective function can be changed to its convex-hull, provided that all the points in the original set are extreme points of the new compact set.

The problem now is how to find a concave function  $J_\epsilon : \mathcal{DS}_s(p_1, p_2) \rightarrow \mathbb{R}$  having the same values as  $J$  at every point of  $\mathcal{P}_p(p_1, p_2)$ . Furthermore, we must be sure that the convex-hull of  $\mathcal{P}_p(p_1, p_2)$  is  $\mathcal{DS}_s(p_1, p_2)$ , and that all  $\mathbf{p}_p$ -matrices are vertices of  $\mathcal{DS}_s(p_1, p_2)$ , even in the presence of the rank-fixing constraint.

Consider Problem 3, where  $J(\mathbf{q})$  is a class  $C^2$  scalar function. Each entry of its Hessian is a continuous function  $\mathbf{H}_{ij}(\mathbf{q})$ .  $J$  can be changed to its concave version  $J_\epsilon$  by

$$J_\epsilon(\mathbf{q}) = J(\mathbf{q}) + \sum_{i=1}^n \epsilon_i q_i^2 - \sum_{i=1}^n \epsilon_i q_i \quad (10)$$

Note that the constraints of Problem 3 include  $q_i \in \{0, 1\}$ ,  $\forall i$ . This means that  $J_\epsilon(\mathbf{q}) = J(\mathbf{q})$ ,  $\forall \mathbf{q}$ . On the other hand  $\mathcal{P}_p(p_1, p_2)$  is bounded by a hypercube  $\mathcal{B} = \{\mathbf{q} \in \mathbb{R}^n : 0 \leq q_i \leq 1, \forall i\}$ . All  $\mathbf{H}_{ij}(\mathbf{q})$  are continuous functions so they are

bounded for  $\mathbf{q} \in \mathcal{B}$  — Weierstrass' theorem. This means that we can always choose a set of finite values  $\epsilon_r$ , defined by

$$\epsilon_r \leq -\frac{1}{2} \left( \max_{\mathbf{q}} \sum_{s=1, s \neq r}^n \left| \frac{\partial^2 J(\mathbf{q})}{\partial q_r \partial q_s} \right| - \min_{\mathbf{q}} \frac{\partial^2 J}{\partial q_r^2} \right) \quad (11)$$

which impose a negative strictly dominant diagonal to the Hessian of  $J_\epsilon$ , that is to say,  $|\mathbf{H}_{ii}| > \sum_{j=1, j \neq i}^n |\mathbf{H}_{ij}|$ ,  $\forall i$ . A strictly diagonally dominant matrix having only negative elements on its diagonal is strictly negative definite [6], so these values of  $\epsilon_r$  will guaranty that  $J_\epsilon(\mathbf{q})$  is concave for  $\mathbf{q} \in \mathcal{B}$  and, therefore, also for  $\mathbf{q} \in \mathcal{DS}_s(p_1, p_2)$ .

Finally, note that problem 4 is constrained to the set of doubly sub-stochastic matrices, defined by conditions 8.2, 8.3 and 9. This set has the structure of a compact convex set in  $\mathbb{R}^{p_1 \times p_2}$ . Its extreme points are the elements of  $\mathcal{P}_p^c(p_1, p_2)$  — see [12] for greater details. This fact together with Theorem 1 proves that the continuous Problem 4 is equivalent to the original discrete Problem 3, since we're assuming that  $J_\epsilon$  was conveniently made concave.

### 3.6 Constraints in Canonical Form

Most concave and linear programming algorithms assume that the problems have their constraints in canonical form. We now show how to put the constraints that define  $\mathcal{DS}_s(p_1, p_2)$  in canonical form, that is, how to state Problem 4 as

$$\begin{aligned} \text{Problem 5} \quad & \mathbf{P}^* = \arg \min_{\mathbf{q}} J_\epsilon(\mathbf{X}, \mathbf{Y}, \mathbf{q}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{q} \leq \mathbf{b}, \quad \mathbf{q} \geq \mathbf{0} \end{aligned}$$

where  $\mathbf{A}_{[m \times n]}$  and  $\mathbf{b}_{[m \times 1]}$  define the intersection of  $m$  left half-planes in  $\mathbb{R}^n$ .

The natural layout for our variables is a matrix  $\mathbf{P}$ , so we use  $\mathbf{q} = \text{vec}(\mathbf{P})$ , where  $\text{vec}()$  stacks the columns of its operand into a column vector. Condition 8.2 is equivalent to  $\mathbf{P} \cdot \mathbf{1}_{[p_2 \times 1]} \leq \mathbf{1}_{[p_1 \times 1]}$ . Applying the  $\text{vec}$  operator [11] to both sides of this inequality we obtain  $(\mathbf{1}_{[1 \times p_2]}^\top \otimes \mathbf{I}_{[p_1]}) \mathbf{q} \leq \mathbf{1}_{[p_1 \times 1]}$ , where  $\otimes$  is the Kronecker product, so set

$$\mathbf{A}_1 = \mathbf{1}_{[1 \times p_2]}^\top \otimes \mathbf{I}_{[p_1]} \quad ; \quad \mathbf{b}_1 = \mathbf{1}_{[p_1 \times 1]} \quad (12)$$

By the same token we express condition 8.3 as

$$\mathbf{1}_{[p_1 \times 1]}^\top \cdot \mathbf{P} \leq \mathbf{1}_{[p_1 \times 1]} \wedge \mathbf{1}_{[p_1 \times 1]}^\top \cdot \mathbf{P} \geq \mathbf{1}_{[p_1 \times 1]} \quad (13)$$

$$\mathbf{A}_2 = \mathbf{I}_{[p_2]} \otimes \mathbf{1}_{[1 \times p_1]}^\top \quad ; \quad \mathbf{b}_2 = \mathbf{1}_{[p_2 \times 1]} \quad (14)$$

The intersection of conditions 8.2 and 8.3 results on the constraints of Problem 5 with

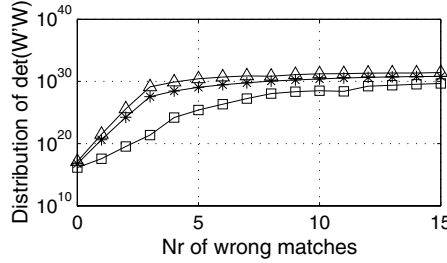
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ -\mathbf{A}_2 \end{bmatrix} \quad ; \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ -\mathbf{b}_2 \end{bmatrix} \quad (15)$$

### 3.7 Image Sequences

With  $F$  frames, the observation matrix is

$$\mathbf{W} = [\mathbf{C}\mathbf{X}_1 \mid \cdots \mid \mathbf{C}\mathbf{P}_{F-1}\mathbf{X}_F]_{[p_1 \times 2F]} \quad (16)$$

The original Problem 1 must be extended to the  $F - 1$  variables  $\mathbf{P}_1$  to  $\mathbf{P}_{F-1}$ . The obvious consequence is an increase of the dimensionality and number of constraints. Furthermore putting the cost function in explicit polynomial form is not trivial. However,  $\det(\mathbf{W}^\top \mathbf{W})$  is still a good criterion, as Figure 6 demonstrates.



**Fig. 6.** Minimum, maximum and average of  $\det(\mathbf{W}'^\top \mathbf{P}' \mathbf{W}' \mathbf{P})$  as functions of the number of forced mismatches, in a 3 image example.

A slight modification in the formalism [12] makes possible to impose rejection mechanism in all images. In other words, it is possible to choose the best  $p_t < p_1$  matches among all possible.

## 4 Minimizing Linearly Constrained Concave Functions

To minimize nonlinear concave cost functions constrained to convex sets we cannot rely on local methods, because many local minima may occur. Instead we apply *global optimization algorithms* that exploit both the concavity of the cost function and the convexity of the constraining set.

Concave programming is the best studied class of problems in global optimization [7,17], so our formulation has the advantage that several efficient and practical algorithms are available for its resolution. Among existing optimal methods, cutting-plane and cone-covering [14] provide the most efficient algorithms, but these are usually hard to implement. Enumerative techniques [16] are the most popular, mainly because their implementation is straightforward. We implemented the method of [3]. As iterations run, the *current best* solution follows an ever improving sequence of extreme points of the constraining polytope. On each iteration, global optimality is tested and a pair of upper and lower bounds are updated. Also, we use a threshold on the difference between bounds as stopping criterion. Since cost grows fast with the number of mismatches — Section 3.2 — this suboptimal strategy returns close to optimal solutions —

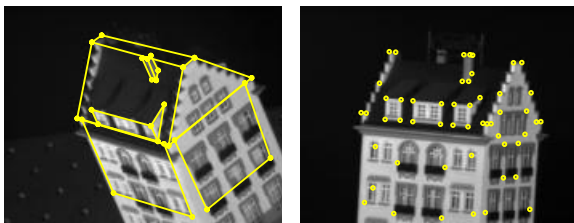
optimal most of the times — and dramatically reduces the number of iterations. Worst case complexity is non-polynomial but, like the *simplex* algorithm, it typically visits only a fraction of the extreme points. Our implementation takes advantage of the sparse structure of the constraints, and deals with redundancy and degeneracy using the techniques of [9].

Recently, special attention has been paid to sub-optimal concave minimization algorithms. [8] describes implementations of Frank and Wolfe [5] and Keller [8] algorithms and claims good performances in large-scale sparse problems. Simulated Annealing [2] is also having growing popularity.

## 5 Results

### 5.1 Experiment 1

Figure 7 shows two images of the Hotel sequence, with large disparity. No prior knowledge was used neither local image assumptions. Points were manually se-



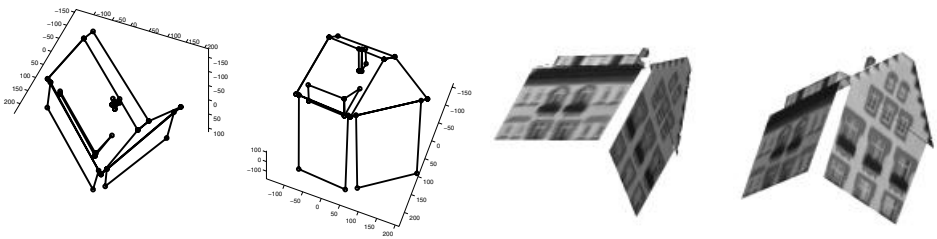
**Fig. 7.** Two images from the Hotel sequence, with manually segmented points.

lected in both images. In the second image, the number of points is double. The wireframe is shown just for a better perception of the object's shape. It is shown in the plot but was never used in the matching process.

The method was applied exactly as described before, using rigidity as the only criterion. Figure 8 shows the reconstruction of the matched points. The solution was found using an implementation of the optimal method of [3]. As expected, all matches are correct, corresponding to the global minimum of  $\det(\mathbf{W}'^\top \mathbf{W}')$ .

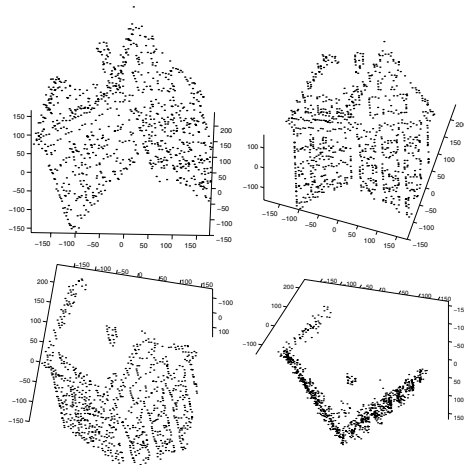
### 5.2 Experiment 2

In this experiment we show how this optimal method can be used to match large points sets in a fully automatic way. At a first stage a corner detector selected a small number of points in each of 8 images from the Hotel sequence. Motion between each pair of images was then computed using the same procedure of experiment 1. An edge detector was then used to extract larger sets of points in each image. With known motion, the epipolar constraint could be used to



**Fig. 8.** Views of a reconstruction of the Hotel.

eliminate most of the matching candidates, so another global criterion — that of [13] — could be optimized with feasible computation. At the end a total of 1000 points were put into correspondence in all 8 images. Less than 10 were wrong matches, and all these were rejected by thresholding the distance between the observations and the rank-3 column space of  $\mathbf{W}$ . The set of remaining points was reconstructed using the factorization method. Figure 9 shows some views of the generated reconstruction.



**Fig. 9.** Some views of an automatically generated 3D cloud of 900 points.

## 6 Conclusion

The described method solves the correspondence problem between points of a fully uncalibrated scaled-orthographic image sequence. Correspondences are set so that the shape and motion computation is optimal, by minimizing the fourth singular value of the observation matrix. We have shown that this is an optimal

criterion under bounded noise assumption. The method is also applicable to other problems where the rigidity assumption can be used, like 3D-to-3D matching, image-to-model matching and multibody factorization.

The most important limitation is the dimensionality of the resulting optimization problems. One practical way of handling this issue is the inclusion of additional *a priori* constraints — see [13] — with minor changes to the underlying problem formulation. Ongoing work is being conducted on experimenting different optimal and suboptimal algorithms, and testing their efficiency. Also, we are currently formulating and testing different ways of building explicit polynomial cost functions for multi-image matching problems.

**Acknowledgments.** João Maciel was supported by FCT through the Science and Technology Program of the 2nd EC Board of Support.

João Costeira was partially supported by FCT Project 3DVideo funded by program POSI-34121-SRI-2000

## Appendix 1: Optimality of Rigidity Cost Functions

If Problem 1 is not degenerate, then both  $\lambda_4(\mathbf{W}'_{\mathbf{P}})$  and  $\det(\mathbf{W}'_{\mathbf{P}}{}^{\top}\mathbf{W}'_{\mathbf{P}})$  are, in some sense, optimal criteria. This is stated in the following proposition:

**Proposition 2** *If there is one single non-degenerate solution to Problem 1, it is possible to find a scalar  $\epsilon > 0$  such that if  $|\mathbf{E}_{(i,j)}^X| < \epsilon$  and  $|\mathbf{E}_{(i,j)}^Y| < \epsilon \forall i, j$  then the solution to Problem 2 is exactly the same  $\mathbf{P}^*$  if  $J(\mathbf{W}'_{\mathbf{P}}) = \det(\mathbf{W}'_{\mathbf{P}}{}^{\top}\mathbf{W}'_{\mathbf{P}})$ . Furthermore, this is the solution to Problem 1 without noise.*

**Proof:** Without noise, a unique  $\mathbf{P}^*$  is also solution to Problem 1 with  $J() = \lambda_4()$  or  $J() = \det()$  because

$$\text{rank}(\mathbf{W}^*) = 3 \Leftrightarrow \lambda_4(\mathbf{W}^*) = 0 \Leftrightarrow \det(\mathbf{W}^{*\top}\mathbf{W}^*) = 0 \quad (17)$$

Non-degeneracy means that there is a nonzero difference between the best and second best cost values if  $J() = \lambda_4()$  or  $J() = \det()$ . This is to say that

$$\exists \delta_1 > 0 : \quad \lambda_4(\mathbf{W}^*) + \delta_1 < \lambda_4(\mathbf{W}_{\mathbf{P}}) \quad , \quad \forall \mathbf{P} \neq \mathbf{P}^* \quad (18)$$

$$\exists \delta_2 > 0 : \det(\mathbf{W}^{*\top}\mathbf{W}^*) + \delta_2 < \det(\mathbf{W}_{\mathbf{P}}{}^{\top}\mathbf{W}_{\mathbf{P}}) \quad , \quad \forall \mathbf{P} \neq \mathbf{P}^* \quad (19)$$

$\lambda_4(\mathbf{W}'_{\mathbf{P}})$  and  $\det(\mathbf{W}'_{\mathbf{P}}{}^{\top}\mathbf{W}'_{\mathbf{P}})$  are continuous functions of the entries of  $\mathbf{W}'_{\mathbf{P}}$  so they are also continuous functions of the entries of  $\mathbf{E}^X$  and  $\mathbf{E}^Y$ . By definition of continuity,  $\exists \epsilon > 0$  such that if  $|\mathbf{E}_{(i,j)}^X| < \epsilon$  and  $|\mathbf{E}_{(i,j)}^Y| < \epsilon \forall i, j$  then equations (18) and (19) still hold for  $\mathbf{W}'$ . This guarantees that, under these noise constraints,  $\mathbf{P}^*$  is still the optimal solution to Problem 1 with any of the two cost function.

## Appendix 2: Writing the Determinant as a Polynomial Function

We will show here how to express the determinant cost function (6) as a bi-quadratic polynomial cost function

$$J_{rig}(\mathbf{P}) = (\mathbf{q}^\top \mathbf{B}_1 \mathbf{q}) (\mathbf{q}^\top \mathbf{B}_2 \mathbf{q}) - (\mathbf{q}^\top \mathbf{B}_3 \mathbf{q})^2 \quad (20)$$

where  $\mathbf{B}_i$  are matrices independent of  $\mathbf{q} = \text{vec}(\mathbf{P})$ .

Start by using the fact that for any two matrices  $\mathbf{M}_{[l \times m]}$  and  $\mathbf{N}_{[l \times n]}$ , if  $\mathbf{L}_{[l \times (m+n)]} = [\mathbf{M} \mid \mathbf{N}]$  and  $\mathbf{N}$  is full-rank — see [6,11] — then

$$\det(\mathbf{L}^\top \mathbf{L}) = \det(\mathbf{M}^\top \mathbf{M}) \det \left\{ \mathbf{M}^\top \left[ \mathbf{I}_{[m]} - \mathbf{N}(\mathbf{N}^\top \mathbf{N})^{-1} \mathbf{N}^\top \right] \mathbf{M} \right\} \quad (21)$$

With  $\mathbf{L} = \mathbf{W}'_{\mathbf{P}} = [\mathbf{C}\mathbf{X}' \mid \mathbf{C}\mathbf{P}\mathbf{Y}']$  and since  $\mathbf{C}$  is symmetrical and idempotent, then

$$\det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}}) = \det(\mathbf{X}'^\top \mathbf{C}\mathbf{X}') \det(\mathbf{Y}'^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P}\mathbf{Y}') \quad (22)$$

with  $\mathbf{\Pi}^\perp = \mathbf{C} - \mathbf{C}\mathbf{X}'(\mathbf{X}'^\top \mathbf{C}\mathbf{X}')^{-1} \mathbf{X}'^\top \mathbf{C}$ . Since the first determinant in equation (22) is positive and independent of  $\mathbf{P}$ , we can simplify the cost function, stating

$$\arg \min_{\mathbf{P}} \det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}}) = \arg \min_{\mathbf{P}} \det(\mathbf{Y}'^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P}\mathbf{Y}') \quad (23)$$

Now define  $\mathbf{Y}' = [\mathbf{u}'_2 \mid \mathbf{v}'_2]$ , where  $\mathbf{u}'_2$  and  $\mathbf{v}'_2$  are respectively the row and column coordinates of points on the second image. This leads to

$$\det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}}) = \det \left[ \frac{\mathbf{u}'_2{}^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P} \mathbf{u}'_2}{\mathbf{v}'_2{}^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P} \mathbf{u}'_2} \middle| \frac{\mathbf{u}'_2{}^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P} \mathbf{v}'_2}{\mathbf{v}'_2{}^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P} \mathbf{v}'_2} \right] \quad (24)$$

For any two matrices  $\mathbf{L}_{[l \times m]}$  and  $\mathbf{M}_{[m \times n]}$

$$\text{vec}(\mathbf{L}\mathbf{M}) = (\mathbf{M}^\top \otimes \mathbf{I}_{[m]}) \text{vec}(\mathbf{L}) \quad (25)$$

The observations  $\mathbf{u}'_2$  and  $\mathbf{v}'_2$  are vectors so, using equation (25), we obtain

$$\begin{aligned} \mathbf{u}'_2{}^\top \mathbf{P}^\top \mathbf{\Pi}^\perp \mathbf{P} \mathbf{v}'_2 &= \text{vec}(\mathbf{P} \mathbf{u}'_2)^\top \mathbf{\Pi}^\perp \text{vec}(\mathbf{P} \mathbf{v}'_2) \\ &= \mathbf{q}^\top (\mathbf{u}'_2 \otimes \mathbf{I}_{[p_2]}) \mathbf{\Pi}^\perp (\mathbf{v}'_2{}^\top \otimes \mathbf{I}_{[p_2]}) \mathbf{q} \end{aligned} \quad (26)$$

There are similar expressions for the other combinations of  $\mathbf{u}'$  and  $\mathbf{v}'$ , so

$$\arg \min_{\mathbf{P}} \det(\mathbf{W}'_{\mathbf{P}}{}^\top \mathbf{W}'_{\mathbf{P}}) = \arg \min_{\mathbf{P}} \left[ (\mathbf{q}^\top \mathbf{B}_1 \mathbf{q}) (\mathbf{q}^\top \mathbf{B}_2 \mathbf{q}) - (\mathbf{q}^\top \mathbf{B}_3 \mathbf{q})^2 \right] \quad (27)$$

with  $\mathbf{q} = \text{vec}(\mathbf{P})$  and

$$\mathbf{B}_1 = (\mathbf{u}'_2 \otimes \mathbf{I}_{[p_2]}) \mathbf{\Pi}^\perp (\mathbf{u}'_2{}^\top \otimes \mathbf{I}_{[p_2]}) \quad (28)$$

$$\mathbf{B}_2 = (\mathbf{v}'_2 \otimes \mathbf{I}_{[p_2]}) \mathbf{\Pi}^\perp (\mathbf{v}'_2{}^\top \otimes \mathbf{I}_{[p_2]}) \quad (29)$$

$$\mathbf{B}_3 = (\mathbf{u}'_2 \otimes \mathbf{I}_{[p_2]}) \mathbf{\Pi}^\perp (\mathbf{v}'_2{}^\top \otimes \mathbf{I}_{[p_2]}) \quad (30)$$

## References

1. R. Berthilsson, K. Astrom, and A. Heyden. Projective reconstruction of 3d-curves from its 2d-images using error models and bundle adjustments. In *Scandinavian Conference on Image Analysis*, 1997.
2. R. Burkard, E. Çela, and Klinz B. On the biquadratic assignment problem. pages 117–146, May 1993.
3. A. Cabot and R. Francis. Solving certain nonconvex quadratic minimization problems by ranking the extreme points. *Operations Research*, 18(1):82–86, Feb 1970.
4. F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proc. CVPR*. IEEE Press, June 2000.
5. R. Fletcher. *Practical methods of optimization*. Wiley, second edition, 1987.
6. R. Horn and C. Johnson. *Matrix Analysis*. Cambridge U. Press, 1985.
7. R. Horst and P. Pardalos, editors. *Handbook of Global Optimization*. Kluwer, 1995.
8. J. Júdice and A. Faustino. Solution of the concave linear complementary problem. In *Recent Advances in Global Optimization*, pages 76–101, 1992.
9. M. Karwan, V. Lotfi, Telgen J., and S. Zions. *Redundancy in Mathematical Programming*. Springer-Verlag, 1983.
10. B. Lucas and T. Kanade. An iterative image registration technique with app. to stereo vision. In *Proc. of the 7th International Joint Conference on AI*, 1981.
11. H. Lutkepohl. *Handbook of Matrices*. Wiley, 1996.
12. J. Maciel. *Global Matching: optimal solution to correspondence problems*. PhD thesis, Instituto Superior Técnico, 2002.
13. J. Maciel and J. Costeira. Robust point correspondence by concave minimization. In *Proc. of the 11th BMVC*, pages 626–635, Bristol, UK, 2000.
14. C. Meyer. A simple finite cone covering algorithm for concave minimization. Technical Report 193, Karl-Franzens-Universitat Graz, Apr. 2000.
15. Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. PAMI*, 7(2):139–154, March 1985.
16. P. Pardalos. Enumerative techniques for solving some nonconvex global optimization problems. Technical Report CS-86-33, Dept. Computer Science, The Pennsylvania State University, Nov. 1986.
17. P. Pardalos and J. Rosen. Methods for global concave minimization: A bibliographic survey. *SIAM Review*, 28(3):367–379, Sep 1986.
18. S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. ICCV*, 1997.
19. A. Shashua. Correspondence and affine shape from two orthographic views. Technical Report AIM 1327, MIT, AI Lab, 1991.
20. G. Sudhir, S. Banerjee, and A. Zisserman. Finding point correspondences in motion sequences preserving affine structure. *CVIU*, 68(2):237–246, November 1997.
21. C. Tomasi and T. Kanade. Shape from motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, November 1992.
22. P. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, Dept. Engineering Science, U. Oxford, 1995.
23. S. Ullman. Maximizing rigidity: the incremental recovery of 3-d structure from rigid and rubbery motion. Technical Report AIM 721, MIT, AI Lab, 1983.
24. Z. Zhang. Determining the epipolar geometry and its uncertainty - a review. *IJCV*, 27(2):161–195, 1998.



# Dramatic Improvements to Feature Based Stereo

V.N. Smelyansky, R.D. Morris, F.O. Kuehnel, D.A. Maluf, and P. Cheeseman

NASA Ames Research Center, MS 269-2, Moffett Field, CA 94035, USA  
[vadim,rdm,kuehnel,maluf,cheesem]@email.arc.nasa.gov

**Abstract.** The camera registration extracted from feature based stereo is usually considered sufficient to accurately localize the 3D points. However, for natural scenes the feature localization is not as precise as in man-made environments. This results in small camera registration errors. We show that even very small registration errors result in large errors in dense surface reconstruction.

We describe a method for registering *entire* images to the inaccurate surface model. This gives small, but crucially important improvements to the camera parameters. The new registration gives dramatically better dense surface reconstruction.

## 1 Introduction

The goal of surface recovery is to take a set of images and estimate the positions and orientations of the cameras that produced the images, and a representation of the surface that was imaged. This is an example of an *inverse problem*. The forward (or direct) problem is: given a surface and the position and orientation of a camera, what is the expected image? This is the area of computer graphics known as rendering [1]. The inverse problem is: given a set of images, estimate the position and orientation of the cameras, and the shape and reflectance properties of the surface. That is, estimate a *generative model* [2,3].

The conventional feature based approach to 3D surface reconstruction takes a sparse set of corresponding feature points from which the positions and orientations of the cameras are estimated. The quality of the camera calibration crucially depends on well localized features. Feature tracking in a sequence of images with small frame to frame disparity has been demonstrated successfully. The two main concerns are the robustness and the accuracy of such an approach. Robustness is usually improved by tracking across a sequence with small inter-frame displacements, but for many applications this cannot be assured. A further concern is that the overall accuracy of the reconstructed 3D model from a sparse point cloud is rather doubtful and prior knowledge is not easily incorporated in the conventional reconstruction scheme.

We show that a robust and accurate reconstruction scheme that can incorporate any prior knowledge can be implemented by applying Bayesian inference of the underlying model space. We postulate *models* for the surface and for the imaging process, and Bayes theorem tell us how to estimate the parameters of these models from the image data. We show that this approach allows us to make

*small but crucially important* improvements to the camera parameters estimated from point matching. These improvements result in a *dramatic improvement* in the accuracy of the 3D surface model.

In this paper we restrict our reconstruction to simple surface models (no occluding parts), therefore we use a simple triangulated mesh model for the geometry of the surface, storing heights,  $z$ , at each vertex of the mesh. We also associate a parameterized reflectance model with the surface. For simplicity here we consider the Lambertian model, and store a single albedo value,  $\rho$  at each vertex. (For multispectral data we store an array of albedo values, one for each spectral band.)

We use the standard pinhole camera model for the image formation process [9], and assume that the internal camera parameters are known. (See, for example, [10] for a simple method of internal camera calibration.) The theoretical development of our approach can be generalized to other imaging geometries and surface reflectance models.

The closest work to that described here is in [4,3]. That work also used a triangulated mesh as the surface representation. The cost function they used is based on minimising the variance of the grey levels of the vertices' projection into the images, rather than the direct image error that is used here. The approach in [4,3] is thus restricted to triangulated meshes that are coarse when projected into the images. The approach described here places no restrictions on the density of the mesh, which may be super-resolved [5]. The system in [4,3] is also restricted to cases where the lighting was from the same direction in all images. Here we require only that the lighting direction is known.

Thus we wish to infer the heights,  $\mathbf{z}$ , the albedos,  $\rho$  and the camera parameters,  $\Theta$ , from the images. Bayes theorem gives

$$p(\mathbf{z}, \rho, \Theta | \{I\}) \propto p(\{I\} | \mathbf{z}, \rho, \Theta) p(\mathbf{z}, \rho, \Theta) \quad (1)$$

We assume that the priors are independent, so that

$$p(\mathbf{z}, \rho, \Theta) = p(\mathbf{z}) p(\rho) p(\Theta)$$

and currently we use a simple smoothness prior for  $\mathbf{z}$  and  $\rho$  based on penalizing curvature, and a uniform prior on  $\Theta$ . These initial prior assumptions are made for the sake of simplicity, and are not fundamental to the approach (see below). The likelihood is assumed to result from Gaussian errors between the image  $\hat{I}(\mathbf{z}, \rho, \Theta)$  synthesized from the surface model and the observed images  $\{I\}$ , giving

$$p(\{I\} | \mathbf{z}, \rho, \Theta) \propto \exp \left( \sum_{f,p} \left( I_{f,p} - \hat{I}_{f,p}(\mathbf{z}, \rho, \Theta_f) \right)^2 / (2\sigma_e^2) \right) \quad (2)$$

where the sum is over all pixels,  $p$  in all images,  $I_f$ . The surface parameters,  $\mathbf{z}$ ,  $\rho$ , are clearly shared between all images. Each image has its own set of camera parameters,  $\Theta_f$ .

The function  $\hat{I}(\mathbf{z}, \rho, \Theta)$  is the process of *rendering* the surface described by  $\{\mathbf{z}, \rho\}$  with the camera location and orientation given by  $\Theta$ . This is clearly non-linear, and makes optimization of the posterior distribution in equation 1 difficult. To make progress in finding the maximum a-posteriori (MAP) estimate, we linearize the image formation process about the current estimate,

$$\hat{I}(\mathbf{z}, \rho, \Theta) = \hat{I}(\mathbf{u}_0) + \mathbf{D}\mathbf{x} \quad (3)$$

where  $\mathbf{u} = \{\mathbf{z}, \rho, \Theta\}$ ,  $\mathbf{x} = \mathbf{u} - \mathbf{u}_0$  and

$$\mathbf{D} = \left\{ \frac{\partial \hat{I}}{\partial \mathbf{z}}, \frac{\partial \hat{I}}{\partial \rho}, \frac{\partial \hat{I}}{\partial \Theta} \right\}$$

If we use a Gaussian smoothness prior with covariance matrix  $\Sigma$  as described above then the linearization converts finding the MAP estimate to the minimization of a quadratic form

$$L = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b} \mathbf{x} \quad (4)$$

$$\mathbf{A} = \Sigma^{-1} + \frac{1}{\sigma_e^2} \mathbf{D} \mathbf{D}^T \quad (5)$$

$$\mathbf{b} = \frac{I - \hat{I}(\mathbf{z}, \rho, \Theta)}{\sigma_e^2} \mathbf{D} \quad (6)$$

which is equivalent to the solution of the system of equations

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (7)$$

Note that for this linearization to proceed, the only restriction on the smoothness prior is that it can be expressed as a covariance matrix. This makes no assumption of spatial uniformity; indeed the prior can easily be made spatially adaptive, to allow for the formation of discontinuities in the heights and albedos.

Consider the structure of this system of equations. The matrix of derivatives  $\mathbf{D}$  is of dimensions

$$\begin{aligned} &(\text{no. of pixels}) \times (\text{no. of heights} + \text{no. of albedos} + \\ &\quad \text{no. of camera parameters}) \end{aligned} \quad (8)$$

or, for the results presented later

$$(256 \times 256) \times (301 \times 301 + 301 \times 301 + 6)$$

The portion of this matrix that is due to the differentials with respect to  $\mathbf{z}$  and  $\rho$  is very sparse, as typically each mesh vertex is used by a few of the triangles that make up the surface, and these triangles project into only a few pixels. The portion due to the differentials with respect to the camera parameters is, however, dense, as changing any one of the camera parameters typically affects

the intensities of all the pixels in the image. As a result of this,  $\mathbf{D}\mathbf{D}^T$  and hence  $\mathbf{A}$  are very large (around  $(180,000 \times 180,000)$  and *dense* (around  $3 \times 10^{10}$  elements). It is clearly impractical to perform joint estimation in this manner. Instead we estimate alternately the camera parameters and the surface parameters, that is

$$\begin{aligned} &\text{given } \Theta, \text{ estimate } \{\mathbf{z}, \rho\} \\ &\text{given } \{\mathbf{z}, \rho\}, \text{ estimate } \Theta \end{aligned} \quad (9)$$

In this way we compute either with a very large, but very sparse matrix when estimating  $\mathbf{z}$  and  $\rho$ , or with a very small, dense matrix when estimating  $\Theta$ . The estimates are made by using conjugate gradient to solve equation 7 in an iterative manner. At convergence, we update the current estimate,  $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{x}$ , re-render to compute new values of  $\hat{I}(\mathbf{z}, \rho, \Theta)$  and  $\mathbf{D}$ , and repeat the solution of equation 7 until a stable solution is reached. This optimization process can be applied in a multiresolution framework, to both accelerate and improve convergence.

This requires an initialization for either  $\Theta$  or  $\{\mathbf{z}, \rho\}$ . We use initial values for  $\Theta$  from point matching a *very small* number of points, or from nominal camera position and orientations, if they are known (eg from rover or aircraft dead-reckoning). In the experiments described later, point matching was used.

The remainder of the paper is organized as follows: In section 2 and 3 we describe the basic rendering algorithm for a renderer which efficiently computes the images *and* calculates the derivative values used for the conjugate gradient search outlined in section 1. Results and Conclusions are given in sections 4 and 5

## 2 The Fractional Derivative Renderer

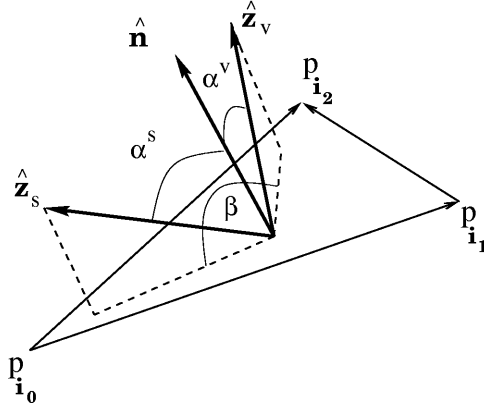
As we have seen, to solve the inverse problem we must be able to simulate the forward problem, to compute  $\hat{I}(\mathbf{z}, \rho, \Theta)$ , (“rendering”). Current rendering technology uses “image space” computation, where the fundamental unit is the pixel. Each pixel is assumed to be illuminated by light from one, and only one, triangular facet. This assumption makes for very fast rendering, but results in aliasing artefacts. It also makes the rendering process non-differentiable.

To enable a renderer to also compute derivatives it is necessary that all computations are done in “object space”. This implies that the light from a surface triangle, as it is projected into a pixel, contributes to the brightness of that pixel with a weight proportional to the fraction of the area of the triangle which projects into that pixel. The total brightness of the pixel is thus the sum of the contributions from all the triangles whose projections overlaps with the pixel

$$\hat{I}_p = \sum_{\Delta} f_{\Delta}^p \Phi_{\Delta}, \quad (10)$$

where  $f_{\Delta}^p$  is the fraction of the flux from triangle  $\Delta$  that falls into pixel  $p$ , given by

$$f_{\Delta}^p = \frac{\bar{A}_{\text{polygon}}}{\bar{A}_{\Delta}}, \quad (11)$$



**Fig. 1.** Geometry of the triangular facet, illumination direction and viewing direction.  $\hat{\mathbf{z}}_s$  is the vector to the illumination source;  $\hat{\mathbf{z}}_v$  is the viewing direction.

where  $\bar{A}$  denotes projected area, and  $\Phi_\Delta$  is the total flux from the triangle, and  $\bar{a}_{\text{polygon}}$  is the area on the image plane of the intersection of the projection of the triangle and the pixel. In the case of Lambertian reflection, this is given by

$$\begin{aligned}\Phi_\Delta &= \rho E(\alpha^s) \cos \alpha^v (\cos \theta)^\kappa \Delta\Omega, \\ E(\alpha^s) &= \mathcal{A} (\mathcal{I}^s \cos \alpha^s + \mathcal{I}^a). \\ \Delta\Omega &= S/d^2.\end{aligned}\tag{12}$$

Here  $\rho$  is an average albedo of the triangular facet. Orientation angles  $\alpha^s$  and  $\alpha^v$  are defined in figure 1.  $E(\alpha^s)$  is the total radiation flux incident on the triangular facet with area  $\mathcal{A}$ . This flux is modeled as a sum of two terms. The first term corresponds to direct radiation with intensity  $\mathcal{I}^s$  from the light source at infinity (commonly the sun). The second term corresponds to ambient light with intensity  $\mathcal{I}^a$ . The parameter  $\theta$  in equation (12) is the angle between the camera axis and the viewing direction (the vector from the surface to the camera);  $\kappa$  is the lens falloff factor.  $\Delta\Omega$  in (12) is the solid angle subtended by the camera which is determined by the area of the lens  $S$  and the distance  $d$  from the centroid of the triangular facet to the camera. If shadows are present on the surface the situation is somewhat more complex. In this paper we assume that there are no shadows or occlusions present in the images. However the presence of shadows and occlusions, whilst making more complex the computation of the image [7,8] and its derivatives, should lead to a more precise and robust surface estimate, as *long-range correlations* are incorporated into the estimation.

The overall complexity of the rendering procedure and derivative calculation procedure scales as

$$\mathcal{C} = \# \text{ images} \times \# \text{ triangles} \times \frac{\text{pixel area}}{\text{triangle area}}.\tag{13}$$

This can be seen from the algorithmic outline of the rendering step:

```

loop over images
  loop over surface triangles
    loop over affected pixels
      calculate fractions and derivatives
      calculate light contribution and
        derivatives
      pixelintensity += light * fractions
    end
  end
end

```

Where the fractions are those in equation 11. The corresponding derivatives are efficiently calculated as shown in the next section.

### 3 Efficient Derivative Computation

To compute the MAP estimates of  $\{\mathbf{z}, \rho\}$  and  $\Theta$  we must compute both the image  $\hat{I}(\mathbf{z}, \rho, \Theta)$  and the derivative matrices  $\mathbf{D}_{\mathbf{z}}$ ,  $\mathbf{D}_{\rho}$  and  $\mathbf{D}_{\Theta}$ .

The derivatives with respect to the albedo values can easily be derived from equations 10 and 12. Note that because  $0 < \rho < 1$ , in practice, we work with transformed albedo values, where  $\rho \rightarrow \log(\rho/(1 - \rho))$ .

Denoting by  $u$  the component of  $\mathbf{z}$  or  $\Theta$  that we are currently considering, the pixel intensity derivatives with respect to  $u$  have two components

$$\frac{\partial \hat{I}_p}{\partial u} = \sum_{\Delta} \left( f_{\Delta}^p \frac{\partial \Phi_{\Delta}}{\partial u} + \Phi_{\Delta} \frac{\partial f_{\Delta}^p}{\partial u} \right) \quad (14)$$

The first component is due to changes in angle – as the height of a vertex changes, the normal to the facet changes, and so the derivative has a component due to the change in angle between the normal and the sun direction; as the camera changes position, the angle between the normal and the ray to the camera changes.

Consider first  $\partial \Phi_{\Delta} / \partial \Theta_i$ . We neglect the derivatives with respect to the fall-off angle,  $\theta$ , as their contribution will be small, and so it is clear from equation 12 that the derivative with respect to any of the camera orientation angles is zero.

The derivative with respect to the camera position parameters is given by

$$\begin{aligned} \frac{\partial \Phi_{\Delta}}{\partial \Theta_i} &\propto \frac{\partial}{\partial \Theta_i} \cos \alpha^v \\ &= \frac{\hat{\mathbf{n}}}{v} (\hat{z}_i - \hat{z}_v (\hat{z}_v \cdot \hat{z}_i)) \end{aligned} \quad (15)$$

where  $\mathbf{v}$  is the vector from the triangle to the camera,  $v = |\mathbf{v}|$ ,  $\Theta_i$  are the three components of the camera position,  $\hat{z}_i$  are unit vectors in the three coordinate directions and  $\hat{z}_v = \mathbf{v}/v$  (see figure 1).  $\hat{\mathbf{n}}$  is the normal to the triangular facet.

Consider now the derivative with respect to the height of one of the mesh vertices,  $z_i$ . The flux derivative,  $\partial\Phi/\partial z_i$ , can be computed directly from the coordinates of the triangle vertices and the camera position using equation 12. For the surface triangle with vertices  $(\mathbf{P}_{i_0}, \mathbf{P}_{i_1}, \mathbf{P}_{i_2})$  the flux derivative with respect to the  $z$  component of the vertex  $\mathbf{P}_{i_0}$  equals

$$\frac{\partial\Phi}{\partial z_{i_0}} = \frac{1}{2}\rho(\mathbf{P}_{i_2} - \mathbf{P}_{i_1}) \times \hat{\mathbf{z}} \cdot \mathbf{g} \frac{S}{d^2}, \quad (16)$$

where

$$\mathbf{g} = \mathcal{I}_s(\hat{\mathbf{z}}_v \cos \alpha_s + \hat{\mathbf{z}}_s \cos \alpha_v - \hat{\mathbf{n}} \cos \alpha_s \cos \alpha_v) + \mathcal{I}_a \hat{\mathbf{z}}_v$$

and  $\hat{\mathbf{z}}$  is a unit normal in the vertical direction.

For a triangle that projects entirely within a pixel, this completes the derivative computation – the second term in equation 14 is the derivative of the *fractional area* of the triangle that projects into the pixel.

### 3.1 Fractional Area Derivatives

When the height of a vertex,  $z$ , changes, its projection on the image plane,  $\bar{\mathbf{P}}$ , also moves, by  $\delta\bar{\mathbf{P}}$ . This gives rise to a change  $\delta\bar{A}_\Delta$  in the area of the projection of the triangle, and also the change  $\bar{A}_{\text{polygon}}$  in the polygon area. It follows from equation 11 that

$$\frac{\partial f_\Delta^p}{\partial z_{i_0}} = \frac{1}{\bar{A}_\Delta} \left( \frac{\partial \bar{A}_{\text{polygon}}}{\partial \bar{\mathbf{P}}_{i_0}} - f_\Delta^p \frac{\partial \bar{A}_\Delta}{\partial \bar{\mathbf{P}}_{i_0}} \right) \frac{\partial \bar{\mathbf{P}}_{i_0}}{\partial z_{i_0}}. \quad (17)$$

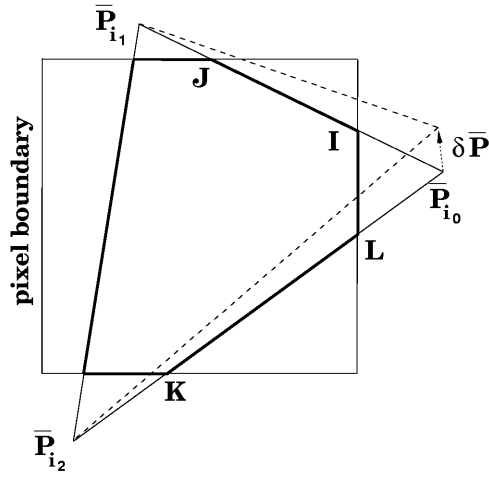
where the point displacement derivative  $\partial\bar{\mathbf{P}}_{i_0}/\partial z_{i_0}$  can be found in [13].

However, when the camera parameters change, the positions of the projections of *all* the mesh vertices into the image plane will move. Then the the derivative of the fractional area is simply a sum of all three position changes and is given by

$$\frac{\partial f_\Delta^p}{\partial \Theta_i} = \frac{1}{\bar{A}_\Delta} \sum_{\mathbf{j}=\mathbf{i}_0, \mathbf{i}_1, \mathbf{i}_2} \left( \frac{\partial \bar{A}_{\text{polygon}}}{\partial \bar{\mathbf{P}}_{\mathbf{j}}} - f_\Delta^p \frac{\partial \bar{A}_\Delta}{\partial \bar{\mathbf{P}}_{\mathbf{j}}} \right) \frac{\partial \bar{\mathbf{P}}_{\mathbf{j}}}{\partial \Theta_i}. \quad (18)$$

The point displacement derivatives are again in [13].

Thus, the task of computing the derivative of the area fraction given in equation 18 is reduced to the computation of  $\partial\bar{A}_\Delta/\partial\bar{\mathbf{P}}_{\mathbf{j}}$  and  $\partial\bar{A}_{\text{polygon}}/\partial\bar{\mathbf{P}}_{\mathbf{j}}$ . Note that the intersection of a triangle and a pixel for a rectangular pixel boundary can, in general, be a polygon with 3 to 7 edges with various possible forms. However the algorithm for computing the polygon area derivatives that we have developed is general, and does not depend on a particular polygon configuration. The main idea of the algorithm can be described as follows. Consider, as an example, the polygon shown in figure 2 which is a part of the projected surface triangle with indices  $\mathbf{i}_0, \mathbf{i}_1, \mathbf{i}_2$ . We are interested in the derivative of the polygon



**Fig. 2.** The intersection of the projection of a triangular surface element ( $\mathbf{i}_0, \mathbf{i}_1, \mathbf{i}_2$ ) onto the pixel plane with the pixel boundaries. Bold lines corresponds to the edges of the polygon resulting from the intersection. Dashed lines correspond to the new positions of the triangle edges when point  $\mathbf{P}_{i_0}$  is displaced by  $\delta \bar{\mathbf{P}}$

area with respect to the point  $\bar{\mathbf{P}}_{i_0}$  that connects two edges of the projected triangle,  $(\mathbf{P}_{i_2}, \mathbf{P}_{i_0})$  and  $(\mathbf{P}_{i_0}, \mathbf{P}_{i_1})$ . These triangular edges contain segments **(I, J)** and **(K, L)** that are sides of the corresponding polygon. It can be seen from figure 2 that when the point  $\bar{\mathbf{P}}_{i_0}$  is displaced by  $\delta \bar{\mathbf{P}}_{i_0}$  the change in the polygon area is given by the sum of two terms

$$\delta \bar{A}_{\text{polygon}} = \delta A_{\mathbf{I}, \mathbf{J}} + \delta A_{\mathbf{K}, \mathbf{L}}$$

These terms are equal to the areas spanned by the two corresponding segments taken with appropriate signs. Therefore the polygon area derivative with respect to the triangle vertex  $\bar{\mathbf{P}}_{i_0}$  is represented as a sum of the two “segment area” derivatives for the two segments adjacent to a given vertex. Using straightforward geometrical arguments one can calculate the areas  $\delta A_{\mathbf{I}, \mathbf{J}}$  and  $\delta A_{\mathbf{K}, \mathbf{L}}$  to first order in the displacement  $\delta \bar{\mathbf{P}}_{i_0}$ . Then the polygon area derivative can be written in the following form:

$$\frac{\partial \bar{A}_{\text{polygon}}}{\partial \bar{\mathbf{P}}_{i_0}} = \frac{1}{2} \hat{\sigma} \cdot \mathbf{W}, \quad \hat{\sigma} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (19)$$

The unit antisymmetric matrix  $\hat{\sigma}$  performs a  $-\pi/2$  rotation in the image plane and vector  $\mathbf{W}$  equals

$$\begin{aligned} \mathbf{W} = & [(1 - R_{\mathbf{I}}^2) - R_{\mathbf{J}}^2] (\bar{\mathbf{P}}_{i_0} - \bar{\mathbf{P}}_{i_2}) \\ & + [(1 - R_{\mathbf{K}}^2) - R_{\mathbf{L}}^2] (\bar{\mathbf{P}}_{i_1} - \bar{\mathbf{P}}_{i_0}). \end{aligned} \quad (20)$$



The ratio factors  $R$  determine the positions of the intersection points  $\mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}$  on the edges of the triangle (see figure 2).

Equations 19 and 20 are the central result of the area fraction derivative computation. It is given for the general case of triangle-pixel intersection where two edges of triangle adjacent to the vertex  $\mathbf{P}_{i_0}$  each have two intersection points. Note that pairs of intersection points,  $\mathbf{I}, \mathbf{J}$  and  $\mathbf{K}, \mathbf{L}$  are defined in a unique way if one considers the triangle edges in counterclockwise order. Therefore equations 19- 20 can be applied to all possible intersection cases. For example, assume that all three triangle vertices are projected inside the pixel. In this case intersection point  $\mathbf{K}$  has merged with  $\mathbf{P}_{i_2}$ , points  $\mathbf{L}$  and  $\mathbf{I}$  have merged with  $\mathbf{P}_{i_0}$  and  $\mathbf{J}$  with  $\mathbf{P}_{i_1}$ . Then in equation 20 we should put

$$R_{\mathbf{K}} = R_{\mathbf{L}} = R_{\mathbf{I}} = R_{\mathbf{J}} = 0. \quad (21)$$

In this case polygon area derivative in equation 19 is reduced to the derivative of the full area of the projected triangle

$$\frac{\partial A'_{\Delta}}{\partial \mathbf{P}_{i_0}} = \frac{1}{2} \hat{\sigma} \cdot (\mathbf{P}_{i_1} - \mathbf{P}_{i_2}). \quad (22)$$

The general rule for computing the ratio factors  $R_{\mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}}$  can be formulated as follows:

- If point  $\mathbf{P}_{i_0}$  lies inside of the pixel one should set in equation 20 ratio factors  $R_{\mathbf{L}} = 0$  and  $R_{\mathbf{I}} = 0$ .
- If point  $\mathbf{P}_{i_2}$  lies inside of the pixel then one sets  $R_{\mathbf{K}} = 0$ .
- If  $\mathbf{P}_{i_1}$  lies inside then  $R_{\mathbf{J}} = 0$ .

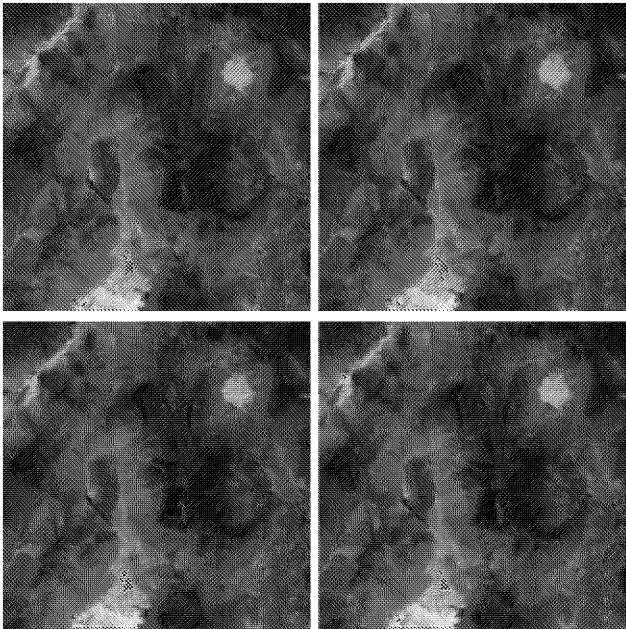
This describes all possible intersection cases and provides a full description for the area fraction derivative (18).

Further details of the derivative computation, together with full details of the point displacement derivatives, can be found in [13].

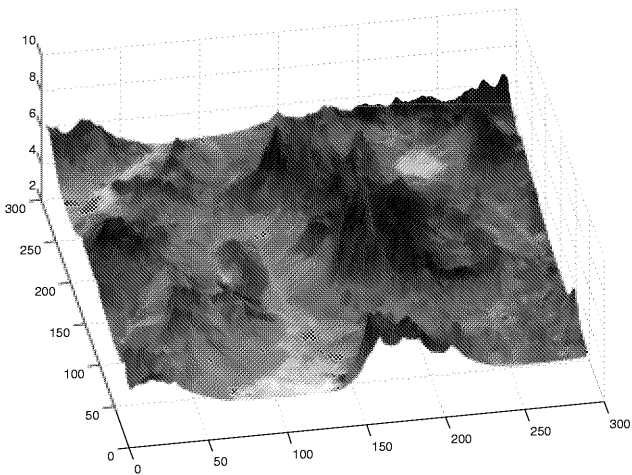
## 4 Results

We present here the results of applying our methodology. We will demonstrate our contention that the *small improvements* made by our registration method to the camera parameter estimates results in *large improvements* to the quality of the inferred surface.

Figure 3 shows four synthetic images of a region of Duckwater, Nevada. They were generated by rendering a synthetic surface. The surface was constructed by using the USGS Digital Elevation Model for the heights, and using the scaled intensities of a LANDSAT-TM image as surrogate albedos. The size of the surface is  $301 \times 301$  points. The distance between grid points was taken to be one unit, and the heights scaled appropriately. Figure 4 shows a perspective view of the surface with expanded vertical scale. Table 1 gives the camera parameters that were used to generate the images.



**Fig. 3.** Four synthetic images of Duckwater, Nevada



**Fig. 4.** True surface

An initial estimate of the camera parameters was made by using point matching [12]. We have found that the Harris corner detector [11] typically used to select features does not find many reliable features in the types of natural scenery we are concerned with here. Table 1 gives the parameters estimated by matching

five points across the four images. Note that these camera parameter estimates appear accurate, with the major error being in the orientation angle (view-up vector).

Using these estimated camera parameters, a dense surface estimate can be made. For space reasons we do not show the surface estimate, instead, in figure 7 we show the error surface, and a cross section in figure 6. The main points to note are that

1. the small inaccuracies in the camera parameter estimation have resulted in an erroneous slope in the surface estimate.
2. the overall height of the surface is shifted upwards; but note that the overall shift is a small percentage (less than 0.5%) of the distance from the surface to the cameras. The overall height is only weakly determined.
3. the albedo estimates are in general quite good (the RMSE for the albedo estimate is 0.022).

Using the gradient-based, whole image, approach to camera calibration to a surface, that we have described above, we then registered the images to the surface estimate. Using the new camera parameters, we re-estimated the surface. This was iterated three times. On a 1.2GHz Athlon PC, rendering and computing the derivative matrix takes less than 2 seconds per image. Convergence of the Conjugate Gradient for updating the surface estimate is achieved in around 200 seconds, and for updating the camera parameters in less than a second. Table 1 gives the final camera parameters, and figure 5 shows the final surface estimate. Again, note that the improvements to the registration parameters are small, but figures 8 and 6 show that these small improvements are crucial. Figure 8 is the error surface and figure 6 is a section through the error surfaces. We note the following:

1. the main improvement in the camera parameter estimation is in the orientation angle, defined by the view-up vector
2. the erroneous slope has been corrected
3. the error in the global height remains
4. the estimate shows most inaccuracies close to rapid changes in albedo, for example the white (salt lake) area to the top right of the surface, where albedo and slope effects have not been completely decoupled.

From these numerical experiments, it is clear that the quality of the surface inference is very sensitive to even small changes in the camera parameters. The *convergence radius* of a successful surface reconstruction with respect to the camera parameters is quite small, and therefore the improvements our registration method give, whilst appearing to be small, have a large effect on the accuracy of the surface estimate.

## 5 Conclusions

In this paper we have described a system that takes a set of images and uses them to infer both the *camera parameters* and a *dense surface model*. It does

this by iterative linearization of a model of the image formation process, and minimization of the error between the whole of the observed and rendered images with respect to the camera and surface parameters. We have demonstrated the convergence of this system on a set of images rendered from a model of a region of Nevada. We have demonstrated the need for *extremely accurate* camera registrations in order to accurately infer a dense surface model, and have shown that our registration method achieves this.

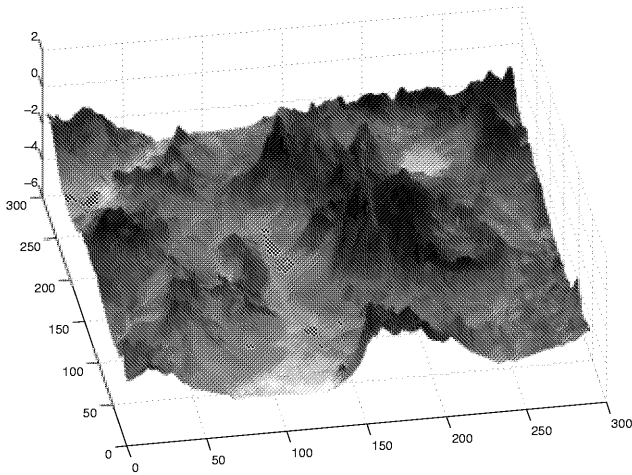
Though the computational cost of our system is high compared to a conventional 3D reconstruction algorithm, it is still of linear complexity, and the system we have described has many advantages. The accurate, dense surface reconstruction which also has albedo information can be used for a number of scientific applications, for example spectroscopy for remote mineral type determination. The scale of the surface model that is estimated is decoupled from the pixel scale of the images via the rendering process. This means that the surface model scale can be chosen by the user, either on the basis of the use to which the surface model will be put, or a scale may be chosen which is best justified by the image data. This is important – if we have many low resolution images of a region, the scale of the surface model may be super-resolved (where a triangular surface element projects onto an area smaller than a pixel on the image plane). If the coverage of the surface by the images is non-uniform, we can specify a spatially-varying mesh for the surface, denser in regions where we have more images.

The information about the surface captured by the system is not just the MAP surface estimate, but also the accuracy of the estimate, represented by the inverse covariance matrix ( $\mathbf{A}$  in equation 5). Knowing the inverse covariance matrix allows for recursive updates – as new images become available the information they contain can be integrated into the model. In Bayesian terminology, the posterior distribution from one set of images (defined by the MAP estimate and the inverse covariance matrix) becomes the prior for estimation with new images.

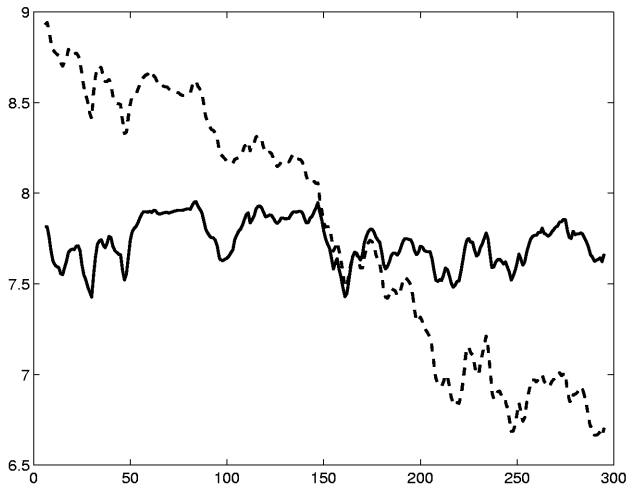
Finally, we are not restricted to only image data. If data from other sensing modalities is available (for example, laser altimetry data) then we can add a term to the likelihood (equation 2) for this data, take derivatives of a model of how this new sensor makes measurements with respect to the surface model parameters, and our surface model estimate will seamlessly integrate the multi-modal information.

## References

1. J. Foley, A. van Dam, S. Finer, and J. Hughes. *Computer Graphics, principles and practice*. Addison-Wesley, 2nd ed. edition, 1990.
2. A.L. Yuille, D. Snow, R. Epstein and P.N. Belhumeur. Determining Generative Models of Objects Under Varying Illumination: Shape and Albedo from Multiple Images Using SVD and Integrability. *International Journal of Computer Vision*, 35(3), 203-222, 1999.

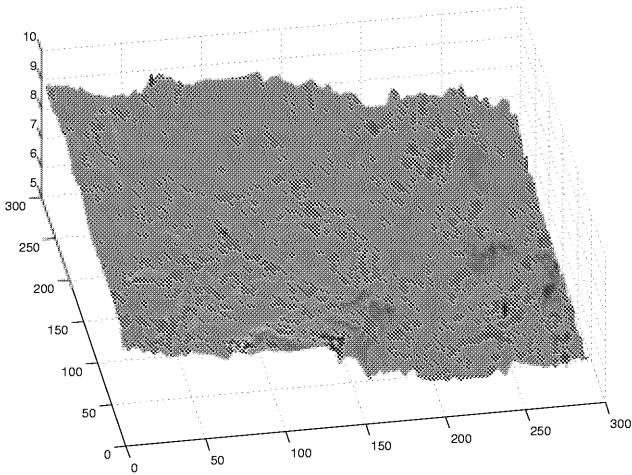


**Fig. 5.** Inferred surface

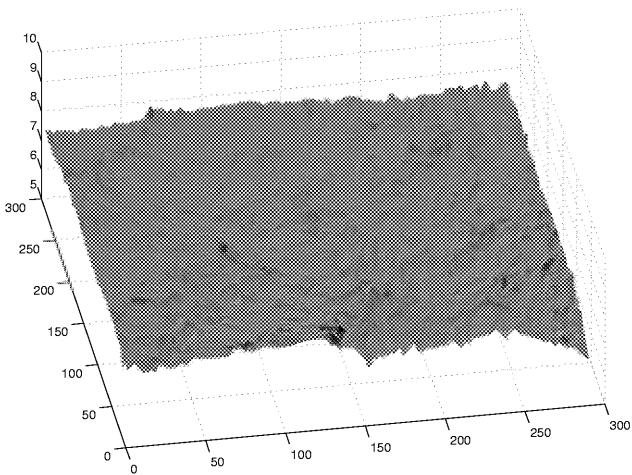


**Fig. 6.** Section through the error surfaces. Dotted line - section through the error surface from the pointmatching surface estimate; solid line - section through the error surface of the final estimate

3. P. Fua and Y.G. Leclerc Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, September 1995.
4. P. Fua and Y.G. Leclerc Registration Without Correspondences. *International Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 121-128, Jun 1994.



**Fig. 7.** Error surface for the surface estimate using camera parameters from point matching. The shape of the surface gives the error in the heights, and the colouring of the surface is propotional to the error in the albedo.



**Fig. 8.** Error surface for the surface estimate using iteratively refined camera parameters

**Table 1.** Camera parameters used to generate the images in figure 3 (“True”), estimated using point matching (“Pointmatch”) (The first image was taken as a known reference.) and final camera parameters estimated using gradient based image error estimation on the estimated surface (3rd iteration)(“Gradient-based”).

		True	Pointmatch	Gradient-based
image 1	camera	(−100, 150, 1700)	(−100, 150, 1700)	(−100.0, 150.0, 1700.0)
	look at	(150, 150, 0)	(150, 150, 0)	(150.0, 150.0, 0)
	view up	(0, 1, 0)	(0, 1, 0)	(−0.00084, 1, $9.0 \times 10^{-5}$ )
image 2	camera	(300, 150, 1700)	(301, 150, 1700)	(301.0, 150.0, 1700.0)
	look at	(150, 150, 0)	(152, 150, 0)	(151.9, 150.1, 0)
	view up	(0, 1, 0)	(0.0031, 1, −0.00031)	(0.00066, 1, $-2.38 \times 10^{-5}$ )
image 3	camera	(150, −100, 1700)	(151, −101, 1700)	(151.0, −101.0, 1700.0)
	look at	(150, 150, 0)	(151, 149, 0)	(151.1, 149.0, 0)
	view up	(0, 1, 0)	(−0.0032, 0.989, 0.146)	(0.00039, 0.989, 0.146)
image 4	camera	(150, 345, 1700)	(153, 348, 1700)	(153.0, 348.0, 1700.0)
	look at	(150, 150, 0)	(151, 151, 0)	(151.0, 151.0, 0)
	view up	(0, 1, 0)	(0.0021, 0.933, −0.116)	(0.00055, 0.9934, −0.115)

5. V.N. Smelyanskiy, P. Cheeseman, D.A. Maluf and R.D. Morris Bayesian Super-Resolved Surface Reconstruction from Images. Proceedings of International Conference on Computer Vision and Pattern Recognition, June 2000
6. J. Bernardo and A. Smith. *Bayesian Theory*. Wiley, Chichester, New York, 1994.
7. K. Weiler and P. Atherton. *Hidden Surface Removal Using Polygon Area Sorting* Proceedings of SIGGRAPH, pp 214-222, 1977.
8. E. Catmull *A Hidden-Surface Algorithm with Anti-Aliasing* Proceedings of SIGGRAPH, pp 6-11, 1978.
9. O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.
10. Z. Zhang. A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71, Microsoft Research, Redmond, Washington.
11. C. Harris. A Combined Corner and Edge Detector. Proceedings of the Alvey Vision Conference, pp 189-192, 1987.
12. Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. AI Journal, vol. 78, pp 87-119, 1994.
13. V.N. Smelyanskiy, R.D. Morris, D.A. Maluf and P. Cheeseman, (Almost) Featureless Stereo – Calibration and Dense 3D Reconstruction Using Whole Image Operations. Technical Report TR01-26, 2001, RIACS. [www.riacs.edu](http://www.riacs.edu)

# Motion Curves for Parametric Shape and Motion Estimation

Pierre-Louis Bazin and Jean-Marc Vézien

INRIA Rocquencourt,  
Domaine de Voluceau, Rocquencourt B.P.105,  
78153 Le Chesnay Cedex, France

**Abstract.** This paper presents a novel approach to camera motion parametrization for the structure and motion problem. In a model-based framework, the hypothesis of (relatively) continuous and smooth sensor motion enables to reformulate the motion recovery problem as a global curve estimation problem on the camera path. Curves of incremental complexity are fitted using model selection to take into account incoming image data. No first estimate guess is needed. The use of modeling curves lead to a meaningful description of the camera trajectories, with a drastic reduction in the number of degrees of freedom. In order to characterize the behaviour and performances of the approach, experiments with various long video sequences, both synthetic and real, are undertaken. Several candidate curve models for motion estimation are presented and compared, and the results validate the work in terms of reconstruction accuracy, noise robustness and model compacity.

**Keywords:** Structure from motion, camera modeling, model selection, motion curves, model-based estimation.

## 1 Introduction

The structure and motion estimation problem is one of the central concerns of computer vision. It can be formulated in several ways, depending on the requirements of the underlying applications. Real-time constraint, stereo imaging, availability of a complete image sequence or of a human operator are important elements that can make some solutions better than others. In our primary application, i.e. video post-production, fast and precise algorithms dedicated to camera motion recovery and 3D reconstruction are key tools for tasks such as special effects generation and augmented reality. Here, the visual quality of the result and the ease of use are important constraints but an operator can feed the algorithm with high level a priori knowledge of the scene described in the images.

Camera motion is very constrained in professional video imaging: the motion is generally very fluid, even for highly composite motions. Some sequences can be very complex, including various compositions of translations and rotations, while others are pure translations along a line or panoramic-like rotations. Curiously, most of the current algorithms for shape and motion recovery do not take advantage of such constraints. Some approaches based on filtering have also suggested the use of motion models to predict the camera position at one frame from previous ones [6] [4] [7]. Model selection between motion models have been proposed in [9] [10] [11]. In these approaches, the



different models to choose from describe the motion between two images. Regular video sequences in post-production may span several seconds, with 25 or 30 frames per second, and the number of motion parameters to be estimated would remain very high, even with the simplest motion models. A better solution in such cases would be to model the complete motion of the camera, in the entire sequence, regardless of the number of images.

A natural way of modeling such motion is to represent the camera trajectory as a 3D curve. Translation and rotation can be separated and modeled independently. The remaining problem is to fix the curve complexity: a parametric curve with many parameters will properly fit very complex motions, but will overfit simpler motions. On the other side, simple curves will perform well only for simple motions, so we need to estimate the real complexity of the unknown camera motion and the corresponding motion parameters simultaneously. An alternative modeling strategy would be to classify the most usual types of motion, such as uniform camera traveling or panoramic motion, then model each of them separately and test every hypothetic motion model. This approach has two disadvantages: it requires the estimation of all different models, and remains highly irrelevant for unusual, erratic or noisy motion.

Both aspects of modeling the trajectories and estimating the motion complexity are addressed in the present work. With a parametric estimation viewpoint, we can describe the shape and motion problem using models, linear or not, that make a direct link between observed data in the images, the underlying scene structure and the camera motion. The structure and motion problem then becomes a parametric estimation problem, in which the free parameters of the models have to be estimated to optimally fit the observed data. In this framework, we have designed a sequential estimation procedure and coupled it with a model selection step to compute the optimal complexity for the estimated motion. Different kinds of motion curves can be integrated in the procedure, and we present several general models based on polynomials.

The motions we consider are supposed to be smooth camera motions, and we show that the approach allows in that case an impressive reduction of the number of free parameters in the system, as well as accurate motion recovery. Under the term *smooth* motion, we do not consider  $C^\infty$  functions but only continuous motions with reasonable acceleration. The technique can handle some trembling and fast changes in the camera motion.

In section 2, we briefly outline the estimation procedure allowing the computation of the trajectory models. A Bayesian selection procedure is proposed in section 3 to set the complexity of the curves. Section 4 reviews the motion curves we have tested. Finally, section 5 presents estimation results on both simulated and real sequences and section 6 discuss them.

## 2 Overview of the Method

The problem of shape and motion recovery we address can be summarized as follows: we observe  $K$  geometric primitives of dimension  $d$  in each of  $T$  images, through primitive tracking. Different parametric models are proposed to recover structure and motion, sharing the same shape model but with more or less complex motion models. With a

Bayesian estimator coupled with an adapted model selection criterion, we proceed to compute the optimal model and associated parameters that best interpret the data.

## 2.1 Scene Description

Our applicative context has led us to consider as primitive features points, segments and rectangles. Moreover, we allow the user to define additional relations of parallelism, orthogonality, collinearity and coplanarity (see Fig.1).



primitives:  
 11 segments,  
 10 rectangles.  
 relations:  
 6,7,11,20 : coplanar,  
 2,3,4,5 : coplanar,  
 17,18,19 : orthogonal,  
 9,13 : orthogonal,  
 2,20 : orthogonal,  
 13,15,19 : parallel,  
 11,12,17 : parallel.

**Fig. 1.** A scene, as described by the user. The primitives are drawn on a key frame and relations are specified beside.

**Geometric reduction.** These constraints, along with inner constraints of specific primitives like rectangles or corners, make the raw description of the scene in terms of points or separate shapes over-determined. To cope with it, a specific geometric reduction algorithm is used to derive the minimal set of unconstrained parameters from the initial primitives and their relationships (see [1] for a detailed explanation). This reduction procedure is complementary to the motion modeling scheme we develop in the present work.

## 2.2 The Projection Model

In video imaging, a pinhole camera model gives a generally good approximation of the real camera. It performs the following perspective projection for a 3D point  $P$ :

$$\begin{aligned} u &= f(x + u_0) \\ v &= f(y + v_0)/r \end{aligned} \quad \text{with} \quad \begin{aligned} x &= \frac{(\mathbf{R} \cdot (\mathbf{P} - \mathbf{T}))_x}{(\mathbf{R} \cdot (\mathbf{P} - \mathbf{T}))_z} \\ y &= \frac{(\mathbf{R} \cdot (\mathbf{P} - \mathbf{T}))_y}{(\mathbf{R} \cdot (\mathbf{P} - \mathbf{T}))_z} \end{aligned}$$

We will assume that intrinsic parameters are known and constant, with the exception of the focal length. This assumption is reasonable, as the parameters can be either obtained

in camera specs, or computed through off-line calibration. The center of projection  $(u_0, v_0)$  and the aspect ratio  $r$  are fixed by the user and a rough approximation of the focal length value  $f$  is supposed known. The remaining parameters are the translation vector  $\mathbf{T}$  and the rotation matrix  $\mathbf{R}$ , that represent the change of coordinates from a reference coordinate system to the camera coordinate system.

Independently from the projection model, the camera moves at each frame  $t$ , so the translation vector  $\mathbf{T}_t$  and the three pose angles  $\theta_t$  that determine the rotation  $\mathbf{R}_t = \mathbf{R}(\theta_t)$  are changing through time.  $(\mathbf{T}_t, \theta_t)$  reflect the camera motion from one frame to the next: they are directly used as motion parameters in most structure and motion algorithms. In the present approach,  $\mathbf{T}_t$  and  $\theta_t$  are modeled with the 3D curves proposed in Sect.4.

### 2.3 Primitive Tracking

The primitives specified in the first image frame are tracked in the images of the sequence, with a correlation and matching procedure (see Fig.2). Some of them are lost due to



Fig. 2. Tracking for the House Sequence: frames 1, 8, 15, 21.

occlusion, loss of track or when they get out of the camera field of view. The relationships between primitives are not enforced in that step, but the inner shape of rectangles is preserved. The 2D tracks obtained are not complete, and are corrupted by noise. This step will be avoided (i.e. simulated) in the synthetic experiments. Details on the actual algorithm dedicated to the tracking step can be found in [2].

### 2.4 Bayesian Estimation

In order to compute the parameters of the scene and the camera motion from noisy 2D tracks, the problem is formulated in a Bayesian estimation framework [5].

**Maximum a posteriori.** We use the *Maximum a Posteriori* estimator (MAP):

$$\begin{aligned}\hat{\Theta}_{MAP} &\triangleq \arg \max_{\Theta} p(\Theta | \{X_{k,t}\} MI) \\ &= \arg \min_{\Theta} [-\log p(\{X_{k,t}\} | \Theta MI) - \log p(\Theta | MI)]\end{aligned}$$

to obtain the set of shape and motion parameters  $\Theta$  that best explains the observed data  $\{X_{k,t}\}$ , where  $X_{k,t}$  are the coordinates of the primitive  $k$  in the image frame  $t$ . The prior

information  $I$  represents the context of the problem and the model  $M$  represents the final parametric model for both shape and motion, that depends on the set of parameters  $\Theta$  to be estimated.

We set  $p(\{X_{k,t}\}|\Theta MI)$  to be a Gaussian distribution centered on the true primitive coordinates with covariance matrices  $\{A_{k,t}\}$ , and the prior  $p(\Theta|MI)$  is supposed to be a uniform distribution  $p(\Theta|MI) = \prod_i \frac{1}{r_i}$  on a bounded domain. The final function to minimize is a non-linear weighted least-square function:

$$J(\Theta) = \sum_{k,t} (X_{k,t} - \hat{X}_{k,t}(\Theta))^T A_{k,t}^{-1} (X_{k,t} - \hat{X}_{k,t}(\Theta))$$

where  $\hat{X}_{k,t}(\Theta)$  represents the non-linear models for the primitive  $k$  projected in the image  $t$ . To find the parameters  $\hat{\Theta}$  that minimize  $J(\Theta)$ , we use the Levenberg-Marquardt minimization algorithm, which has fast convergence with such non-linear least-square functions. Like all local minimization algorithms, it needs a good first estimate of the parameters, which can be problematic to provide.

**Frame-by-frame minimization.** To overcome the initial estimate problem, we proceed on a frame-by-frame basis. The following relation holds:

$$p(\Theta|\{X_{k,t}\}_{t=0\dots T_0+1}MI) \sim \frac{p(\{X_{k,t}\}_{t=0\dots T_0+1}|\Theta MI)}{p(\{X_{k,t}\}_{t=0\dots T_0}|\Theta MI)} p(\Theta|\{X_{k,t}\}_{t=0\dots T_0}MI)$$

Thus, if we take the negative log of the formula, the function to minimize for data in the frames 0 to  $T_0 + 1$  is the sum of the function already minimized for frames 0 to  $T_0$  and the least square error on the new data at frame  $T_0 + 1$ , so the MAP estimate  $\hat{\Theta}_{T_0}$  is an efficient first guess for  $\hat{\Theta}_{T_0+1}$ .

An initial guess for the parameters must be computed for  $t = 0$ , when there is no camera motion: the scene can be derived from an arbitrary flat reconstruction of the projected points. The camera motion is then recovered gradually, and the procedure for optimally adjusting the motion complexity will be naturally integrated in this sequential framework.

### 3 Selection of the Complexity

The MAP estimator gives, at every frame, the optimal parameter set for a given model. To properly model camera motion, motion curves of different complexity must be tested and the best one should be selected. Thanks to the frame-by-frame estimation, we can start using the simplest curve model, then test more complex models while adding data from new frames.

#### 3.1 Model Selection Criteria

To select the best model, we need a selection criterion that ranks the concurrent models. In statistical modeling, different kinds of criteria are used, relying on different information

theories. The global idea is to choose the simplest model  $M$  that represents the relevant data but not the noise. The criterion gives a quantitative balanced value of both the model complexity and the error on the data  $D$ .

Ad hoc criteria can be employed, usually a simple function of the number  $p$  of parameters  $\Theta$  in the model  $M$ , but such methods must be tuned and tested extensively. From considerations on the likelihood, the *Akaike Information Criterion*  $AIC = L + 2p$  has been derived, where  $L = -2 \log p(D|\Theta MI)$  is the inverse log-likelihood. *Minimum Description Length* (MDL) is another well-known criterion, based on algorithmic theory of information. The MDL criterion computes the code length necessary to describe the model and the code length necessary to describe the remaining stochastic complexity of the data [12]. The *Bayesian Information Criterion*  $BIC = L + p \log N$ , with  $N$  the number of data measurements, is used for Bayesian inference and maximizes the evidence  $p(D|MI)$ .

All these criteria are very similar, and the distinctions between them on a practical point of view is often subtle [3]. In statistical modeling, these criteria are asymptotic and may differ for geometric problems [8]. A transposition in the geometric context makes different approximations related to the models in use, that can lead to slightly different criteria [9] [11]. For parameters estimation, we used a Bayesian formalism. To maintain coherence, we choose to use the evidence maximization for model selection as well.

### 3.2 Bayesian Evidence Criterion

The geometric BIC criterion is not directly adapted here and we must compute a new approximating criterion for evidence. The outline of the computation is similar to BIC derivation from evidence in [11], but asymptotic approximations are avoided here. The evidence is  $p(D|MI) = \int_{\Theta} p(D|\Theta MI)p(\Theta|MI)d\Theta$ . Stating  $\Phi(\Theta) = -\log(p(D|\Theta MI)p(\Theta|MI))$ ,  $p(D|MI) = \int_{\Theta} \exp -\Phi(\Theta)d\Theta$ . As  $\hat{\Theta}$  is the minimum of  $\Phi(\Theta)$  and  $\nabla \nabla \Phi(\hat{\Theta}) = \hat{\Lambda}_{\hat{\Theta}}^{-1}$ , a second order Taylor approximation of the function  $\Phi(\Theta)$  at  $\hat{\Theta}$  gives:

$$\begin{aligned} p(D|MI) &\approx \int_{\Theta} \exp -\Phi(\hat{\Theta}) - \frac{1}{2}(\Theta - \hat{\Theta})^T \hat{\Lambda}_{\hat{\Theta}}(\Theta - \hat{\Theta})d\Theta \\ &= (2\pi)^{p/2} |\hat{\Lambda}_{\hat{\Theta}}|^{1/2} \exp -\Phi(\hat{\Theta}) \end{aligned}$$

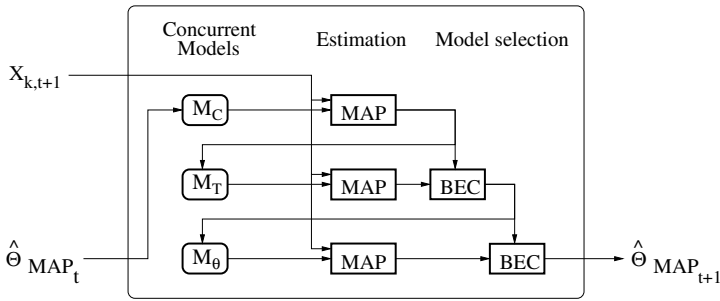
The expression of  $\Phi(\Theta)$  specific to our problem is given by  $\Phi(\Theta) = \frac{1}{2}J(\Theta) + \frac{N_d T}{2} \log(2\pi) + \frac{1}{2} \sum_{k,t} \log |A_{k,t}| + \sum_i \log r_i$ , where  $N_d T$  represents the total number of data coordinates over the  $T$  frames of the sequence, and finally:

$$\begin{aligned} BEC = -2 \log p(D|MI) &\approx J(\hat{\Theta}) + (N_d T - p) \log(2\pi) + \sum_{k,t} \log |A_{k,t}| \\ &\quad + \log |\hat{\Lambda}_{\hat{\Theta}}^{-1}| + 2 \sum_i \log r_i \end{aligned}$$

The quantities in the criterion are all set a priori or evaluated in the MAP estimation of parameters, thus no further approximation is needed.

### 3.3 Complete Procedure

To integrate the model selection step in the sequential estimation procedure, we have to estimate all possible models, then compare them with the criterion. If the number of possible models is important, the computational effort becomes rapidly untractable. To avoid such problem, the models must be *nested*, in the sense that models of lower complexity must be contained in models of higher complexity. Therefore, more complex models are initialized from simpler models. At each frame, only the currently selected model is memorized and more complex models can be computed from it. The motion complexity is supposed to stay unchanged or increase from one frame to the next, independently for translation and rotation parameters. Three different estimations are then necessary to select the optimal model (see Fig.3).



**Fig. 3.** The current model  $M_C$  and more complex models for translation  $M_T$  and rotation  $M_\theta$  are estimated and compared at each frame.

## 4 Motion Curves

The sequential estimation and selection algorithm proposed here impose constraints on the motion model we can consider: the motion curves must be nested, and more or less complex curves should share parameters close together. Three families of curves are proposed: a polynomial of high degree representing the complete motion, linear splines that model the motion more locally, and a hybrid model made of piecewise polynomials of various possible degrees.

### 4.1 Chebyshev Polynomials

In the first model, the parameters  $\mathbf{T}(t), \boldsymbol{\theta}(t)$  are functions of a single polynomial of degree  $K, L$ :

$$\begin{aligned}\mathbf{T}(t) &= \sum_{n=0}^N \mathbf{a}_n P_n(t) \\ \boldsymbol{\theta}(t) &= \sum_{m=0}^M \mathbf{b}_m P_m(t)\end{aligned}$$

where  $P_n(t)$  is the Chebyshev polynomial of degree  $n$ . Therefore, the motion parameters become  $\{\mathbf{a}_n, \mathbf{b}_m\}$ , reducing the degrees of freedom from  $6T$  to  $3N + 3M + 6$  parameters.

Chebyshev polynomials are preferred for several reasons. First, they are known to be close to the minimax approximation polynomial for general functions. Next, Chebyshev coefficients  $a_n$  smoothly decrease to zero when the degree  $n$  increases. The polynomials values range from -1 to +1, so the coefficients are homogeneous to the approximated values. Finally, the polynomial curve changes gradually when high order coefficients are set to zero.

The complexity of the motion is represented by the degree of polynomials, and the concurrent motion models to choose from in the model selection phase are simply polynomials of higher or lower degree. The models are nested and the polynomial coefficients common to simple and complex models are close together, the coefficients of simple models directly giving a good first estimate for models of higher complexity.

## 4.2 Linear Splines

Camera motions can be viewed as a sequence of simple moves, and it is reasonable to hope that changes on the motion estimation on one of the last frames should not induce changes on the motion corresponding to the first frames. For such motion, piecewise polynomials can be preferred to a global high degree polynomial. As the simplest piecewise models, we use the linear spline:

$$\begin{aligned} \mathbf{T}(t) &= \mathbf{a}_n(1 - \tau_n) + \mathbf{a}_{n+1}\tau_n & \text{if } t \in [t_n, t_{n+1}] \\ \boldsymbol{\theta}(t) &= \mathbf{b}_m(1 - \tau_m) + \mathbf{b}_{m+1}\tau_m & \text{if } t \in [t_m, t_{m+1}] \end{aligned}$$

where  $\tau_n = \frac{t - t_n}{t_{n+1} - t_n}$ . The definition is identical for  $\tau_m$ . The motion parameters are  $\{\mathbf{a}_n, \mathbf{b}_m\}$  along with the two series of time intervals  $\{[t_n, t_{n+1}]\}$ ,  $\{[t_m, t_{m+1}]\}$ . The selection problem is not to find the proper degrees, that are fixed, but to determine how many pieces to make and where to place them on the time interval of the sequence.

In the model selection procedure, concurrent models will be models with more or less pieces. More complex curves are obtained by adding a new interval when adding a new frame, whereas the intervals for simpler models remain unchanged. This model family is nested, like Chebyshev polynomials: models of higher complexity just have additional polynomial pieces. Intervals with the same pieces have the same coefficients, regardless of the rest of the curve and polynomials of lower complexity are a good first estimate for polynomials of higher complexity.

## 4.3 Piecewise Chebyshev Polynomials

Finally, a hybrid model between Chebyshev and piecewise curves has been tested. This model is a disjoint succession of Chebyshev polynomials:

$$\begin{aligned} \mathbf{T}(t) &= \sum_{n=0}^{N_i} \mathbf{a}_{n,i} P_n(\tau_i) & \text{if } t \in [t_i, t_{i+1}], \\ \boldsymbol{\theta}(t) &= \sum_{m=0}^{M_j} \mathbf{b}_{m,j} P_m(\tau_j) & \text{if } t \in [t_j, t_{j+1}], \end{aligned}$$

This model has the smoothness of Chebyshev polynomials on regular parts, and can accommodate for composite motions. It is nested, and simple curves are good initializations for more complex ones. For this family of models, there are two possibilities to increase the complexity: we can add a degree to the last polynomial which describes the current motion, or we can create a new polynomial to fit the motion independently of the past frames. Therefore, this model necessitates to perform two additional estimation and selection steps.

#### 4.4 Other Curve Models

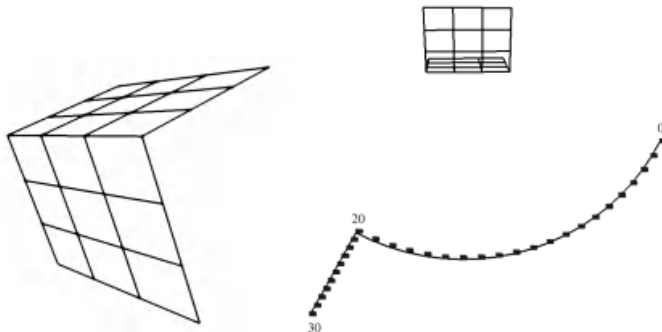
Other curve models could come to mind to describe the motion, like Bernstein polynomials or cubic splines. However, these are generally suited for interpolation purposes, and the modeling procedure proposed here is more related to approximation problems. Some experiments with such interpolation curves led to poor results, mainly due to instabilities of the curve coefficients when the complexity is modified.

### 5 Experiments

The technique proposed in this paper has been tested on both synthetic and real video sequences, in order to characterize its performances. Cramer-Rao lower bounds for accuracy have been computed on the simulated experiments. Finally, classical estimation without motion model have also been performed.

#### 5.1 Simulated Data

To perform the synthetic experiments, a 3D scene was created. The scene consists in two sides of a Rubik's cube, and the camera orbits the cube for 20 frames, then goes backward along a straight line for 10 frames (see Fig.4). The scene description is composed of 2



**Fig. 4.** Simulation settings: left: the scene observed at  $t = 0$ ; right: the camera path.



rectangles, 8 segments as well as geometric constraints. The tracking part is completely simulated by exact projection of the true 3D points and addition of Gaussian noise of increasing magnitude.

Experiments have been performed with a Gaussian noise of zero mean and standard deviation of 0, 1, 2, 3, 5, 7 and 10 pixels. For each configuration and each noise level, three instances of synthetic data were generated for estimation and the results were averaged. Errors in 3D shape, translation and rotation are computed with respect to the true values, after an appropriate transformation that makes the scenes suitable for comparison. This transformation places the origin of space at the center of the 3D scene, rotates the coordinate system to match the scene eigenvectors and normalizes the standard deviation of scene points from the origin. The final error curves have been obtained by averaging the results with the three noisy sets. The complexity of the optimal curves are defined as the total number of 3D parameter vectors needed to describe them. Theoretical Cramer-Rao lower bounds (CRLB) have also been computed from the original scene.

The error curves in Fig.5 show that the results are similar with the different models and without model. The optimization procedure has always properly converged, as the 2D residual errors are very close to their theoretical bounds. Reconstruction errors are also close to the Cramer-Rao bounds, especially for the motion.

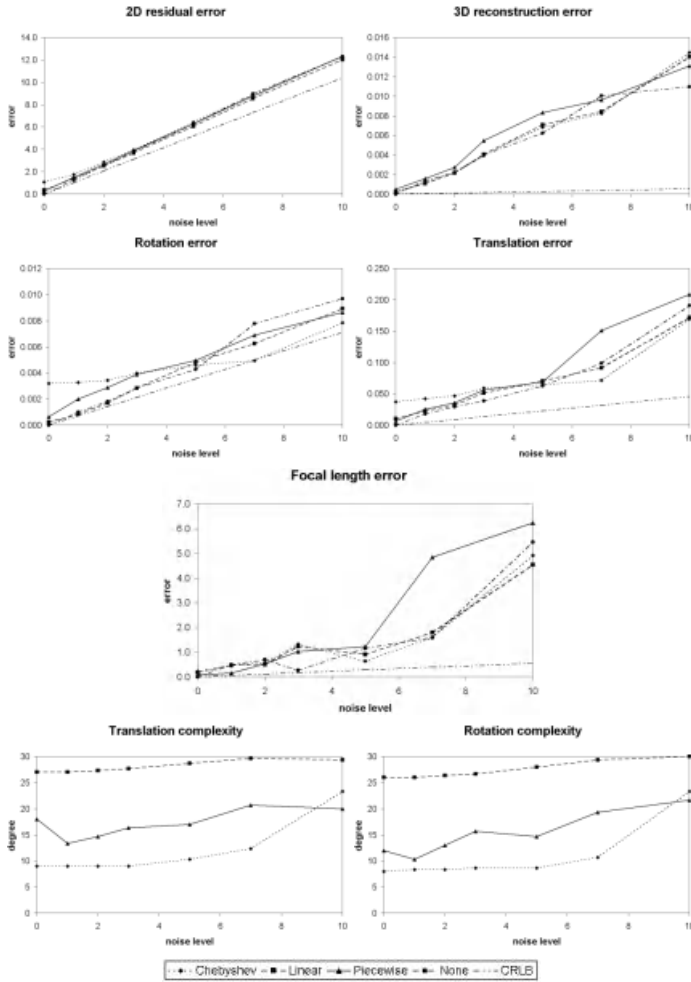
The big difference here lie in the complexity of the motion models needed to achieve such accuracy. The complexity of Chebyshev polynomials is the lowest, and remains extremely stable as noise increase. The linear spline model is not very efficient, as its complexity remains very high, but is also stable. Piecewise polynomials are in-between, with larger complexity fluctuations: in practice, they often hesitate between a long Chebyshev curve of high degree and numerous curves of degree 1 or 2. This stability is remarkable, as it shows that the estimated complexity is really related to the complexity of the motion itself and not to the noise. The corresponding motion curves are not an approximation, as they manage to reach the same accuracy levels as model-free estimation.

## 5.2 Real Sequences

The capabilities on real video sequences of the algorithm has also been tested. The following videos, borrowed from various laboratories, have been used:

- the "Sport" sequence (from Inria Syntim): the camera motion is a slow translation in straight line,
- the "House" sequence (from Inria Movi): the camera undergoes a fast circular motion around a model house,
- the "StainX" sequence (from Alias|Wavefront): the camera is moving towards the scene in a translation/rotation motion,
- the "Begijnhof" sequence (from K. U. Leuven Visics): the camera is hand-held and moving backward almost linearly, but not smoothly.

For every sequence, 10 to 20 features (points, lines and rectangles) have been tracked, some of them only on a limited part of the sequence (see the features on the first images on Tab.1). The structure of the scene and the camera motion have been computed with the three motion models, and with the model-free algorithm. Reconstruction results are







**Fig. 5.** Error and complexity results in the synthetic experiments, for increasing noise levels.

more or less subjective for real sequences, and error curves cannot be drawn. We can still compare the visual 3D reconstruction obtained with the different motion curves (see Fig.7), but the only reliable metric is the 2D residual error. Its mean and standard deviation values are given on Tab.1, along with the complexities for the three models. The visual correctness of the reconstruction can be assessed by mean of augmented reality (addition of synthetic elements), as presented on Fig.6.

In all cases, the 3D structure reconstructions are visually correct and seem identical. The camera motion also correspond to the qualitative descriptions given above. The residual errors are low, and both mean and standard deviation values are very close

**Table 1.** Residuals and complexity on the real sequences.

				
	Sport	House	StainX	Begijnhof
Number of frames:	90	20	70	42
2D residual error				
no model:	3.438	3.394	4.357	2.783
Chebyshev:	3.470	3.412	4.579	2.778
linear:	3.490	3.547	4.392	2.732
piecewise:	3.464	3.546	4.371	2.643
2D standard deviation				
no model:	2.972	3.367	2.461	2.159
Chebyshev:	3.095	3.429	2.570	2.202
linear:	3.376	3.469	2.527	2.198
piecewise:	3.056	3.478	2.498	2.219
translation complexity				
Chebyshev:	6	5	9	6
linear:	4	13	4	11
piecewise:	7	16	10	27
rotation complexity				
Chebyshev:	5	4	9	24
linear:	3	12	9	21
piecewise:	7	17	12	27

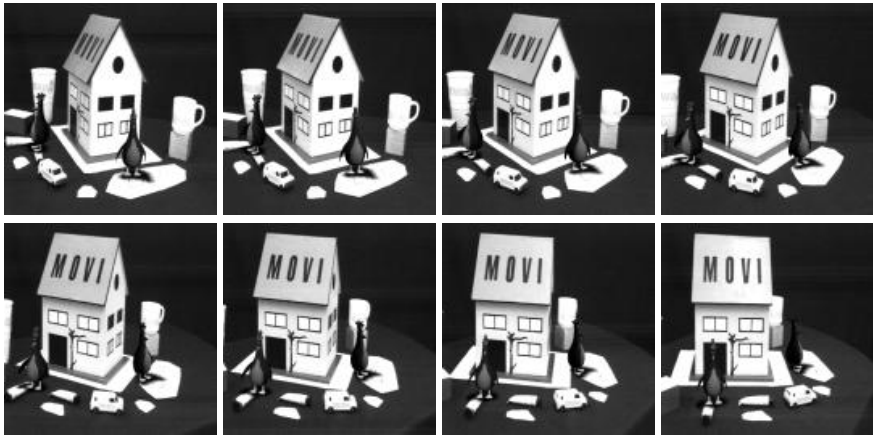
for any given sequence: in terms of the reprojection error, the results are all identically optimal.

The first important difference is the reduction of the motion complexity when we use motion curves. Even with the longest sequences, the complexity remains very low, and the accuracy is not jeopardized by the reduction. Moreover, the motion curves provide a better robustness: in Fig.7, the reconstructed path for the camera without model is strongly perturbed by noise, whereas the visual motion in the video sequence is perfectly smooth. These perturbations are removed with any of the motion models.

As in the synthetic experiments, Chebyshev polynomials have proven to give a very compact motion model without loss of accuracy. Their approximation properties allow very stable computations, and changes in the trajectory are correctly handled.

Linear splines perform better here. On almost linear motions, they go below Chebyshev polynomials (cf. the Sport sequence), but become limited for curved motions like the orbit motion of the simulations or the House sequence.

Piecewise Chebyshev polynomials still suffer from their flexibility: they perform well, but always with an increased complexity. It is hard to discriminate between the two possible ways to grow in complexity here, and the final models will involve long pieces of Chebyshev curves and small line segments or even points on the less regular parts.



**Fig. 6.** Augmented reality with virtual penguins on the House sequence: images 1, 5, 9, 13 (top), 17, 21, 25, 30 (bottom).

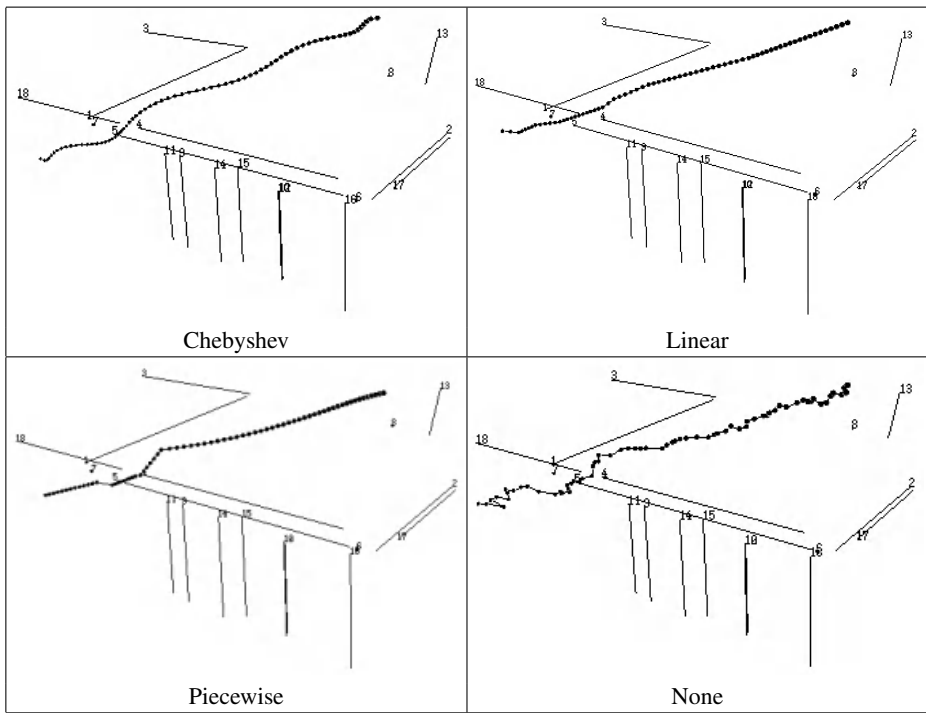
Finally, the smoothness assumption shared by the three models is not a strong constraint. In the Begijnhof sequence, the camera motion is not smooth, as the camera oscillates when the cameraman moves. The related motion models capture this phenomenon with an increased complexity of the orientation curve, and the results are comparable in 2D and in 3D with an unconstrained structure and motion recovery.

## 6 Conclusions

We presented here an estimation framework for shape and motion recovery from video sequences. The technique takes into account the continuity and smoothness of the camera motion, through motion curves. The curves are optimally fitted to more or less complex motions thanks to a sequential estimation and selection procedure.

As the motion curves drastically reduce the number of parameters to estimate, the computations are greatly simplified and the estimation is more robust to tracking noise. The accuracy of reconstruction is not decreased by the approximation introduced by the curves. The two major roles of the introduced curves are to reduce the complexity of the problem through a limited and relevant set of parameters and to factor out the tracking noise from the recovery by imposing some regularity.

The polynomial functions proposed as curve models can describe any kind of motion. Chebyshev polynomials are near optimal in the sense of function approximation, and always perform well in practice for this problem. Their only weakness is for describing sharp changes in the motion, generally related to a change in the nature of the motion itself. We plan to investigate more subtle models to get over this limitation. Studying descriptions of different kinds of camera motions could also help to refine even more the models, particularly for erratic and trembling motions like hand-held camera motions.



**Fig. 7.** Reconstructed scene from the StainX sequence, with the proposed motion models. The spheres indicate the camera position at each frame.

## References

1. P. L. Bazin. A parametric scene reduction algorithm from geometric relations. In *Proc. Vision Geometry IX*, SPIE's 45th annual meeting, San Diego, 2000.
2. P. L. Bazin and J. M. Vézien. Tracking geometric primitives in video streams. In *Proc. Irish Machine Vision and Image Processing Conference*, Belfast, 2000.
3. Kishore Bubna and Charles V. Stewart. Model selection and surface merging in reconstruction algorithms. In *Proc. International Conference on Computer Vision*, Bombay, 1998.
4. Alessandro Chiuso, Paolo Favaro, Hailin Jin, and Stefano Soatto. 3-d motion and structure from 2-d motion causally integrated over time: Implementation. In *Proc. European Conference on Computer Vision*, Dublin, 2000.
5. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall, 1995.
6. Xiaoping Hu and Narendra Ahuja. Motion and structure estimation using long sequence motion models. *Image and Vision Computing*, 11(9):549–569, november 1993.
7. Frederik Kahl and Anders Heyden. Auto-calibration and euclidean reconstruction from continuous motion. In *Proc. International Conference on Computer Vision*, Vancouver, 2001.
8. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, Amsterdam, 1996.

9. Kenichi Kanatani and Chikara Matsunaga. Geometric mdl and its media applications. In *Proc. Workshop on Information-Based Induction Sciences*, Izu, 2000.
10. S. J. Maybank and P. Sturm. Minimum description length and the inference of scene structure from images. In *Proc. IEE Colloquium on Applications of Statistics to Pattern Recognition*, 1999.
11. P. H. S. Torr. *Data segmentation and model selection for computer vision*, chapter Model selection for structure and motion recovery from multiple images. Springer, 2000.
12. P. Vitányi and M. Li. Ideal mdl and its relation to bayesianism. In *Proc. Information, Statistics and Induction in Science Conference*, Melbourne, 1996.

# Bayesian Self-Calibration of a Moving Camera <sup>★</sup>

Gang Qian and Rama Chellappa

Center for Automation Research and  
Department of Electrical and Computer Engineering  
University of Maryland  
College Park, MD 20742, U.S.A.  
{gqian, rama}@cfar.umd.edu

**Abstract.** In this paper, a Bayesian self-calibration approach is proposed using sequential importance sampling (SIS). Given a set of feature correspondences tracked through an image sequence, the joint posterior distributions of both camera extrinsic and intrinsic parameters as well as the scene structure are approximated by a set of samples and their corresponding weights. The critical motion sequences are explicitly considered in the design of the algorithm. The probability of the existence of the critical motion sequence is inferred from the sample and weight set obtained from the SIS procedure. No initial guess for the calibration parameters is required. The proposed approach has been extensively tested on both synthetic and real image sequences and satisfactory performance has been observed.

## 1 Introduction

Automatic retrieval of the intrinsic parameters of a (relatively) moving camera from an observed image sequence has been of great interest to researchers in computer vision since the early 1990s. Subsequent to the pioneering work on camera self-calibration reported by Maybank and Faugeras [1,2], numerous algorithms have been proposed to calibrate cameras with constant (see [3] for a review) or varying intrinsic parameters [4,5,6,7]. Although significant efforts have been made to solve the self-calibration problem, several challenges still remain: sensitivity to observation noise, initialization of the algorithms, and processing of critical motion sequences (CMS) [8,9,10,11]. The first two challenges are common difficulties arising in nonlinear problems such as the camera self-calibration problem. In some situations, these two factors interact with each other and make the problem more complex. For example, large observation noise might create more local minima (or maxima) and can easily trap iterative optimization methods such as Levenberg-Marquardt and steepest-descent algorithms in local minima, preventing them from converging to the global optimal solution. Hence a good initial guess of the calibration parameters is needed.

---

<sup>★</sup> Partially supported by the U.S. Army Research Laboratory (ARL) Collaborative Technology Alliance contract DAAD19-01-2-0008.

In addition to the difficulties due to noise sensitivity and initialization, the existence of CMSs makes camera self-calibration even more difficult in practice. CMSs are sequences of camera motions resulting in inherent ambiguities in camera self-calibration and therefore ambiguities in uncalibrated Euclidean reconstruction [8]. Any practical self-calibration method must take into account the processing of CMSs since CMSs frequently occur in applications. Previous research [9] has shown that ambiguous Euclidean reconstructions from a CMS are conjugated and hypothesis verification can be used to detect and determine the type of CMS. Nevertheless, in the presence of noise, some camera motion sequences which are not CMSs can also result in ambiguous Euclidean reconstruction. Moreover, it has been recently reported in [12] that camera motion sequences “close” to CMSs in the sense of producing ambiguous Euclidean reconstructions can be far away from any type of CMSs in the motion sequence space in the sense of  $L_2$  norm. If hypothesis verification is applied to this kind of sequence, it will be classified as one type of CMSs, and the true solution of the motion sequence, which is actually outside of CMSs, will be lost. Therefore, hypothesis verification is not sufficient in these circumstances.

In this paper, we focus on the main problem of self-calibration: estimation of the field of view (FOV) with all the other intrinsic parameters known. The unknown FOV can be either constant or varying throughout the image sequence. We develop a self-calibration algorithm, which is capable of processing CMS and yielding reasonable calibration estimates without any specific requirements of initialization. The new approach is developed based on the sequential importance sampling (SIS) technique. The SIS procedure is recently introduced by [13] to estimate the state parameters of a non-linear/non-Gaussian dynamic system. In SIS, the joint posterior distribution of the state parameters given the observations is approximated by a set of samples and their related weights. The SIS procedure has been used for solving the structure from motion (SfM) problem. An SIS-based SfM algorithm has been developed in [14] and it was shown to be robust to feature tracking errors and to be able to handle motion/structure ambiguities. However, in that case, all the intrinsic parameters of the camera are assumed to be given. In the paper, we still use the SIS method to attack the camera self-calibration problem because of its capability of solving problems involving non-linear systems.

## 2 Theoretical Background

### 2.1 Self-Calibration of a Moving Camera

In many practical situations, the calibration of the camera used to capture the sequences is not available, i.e. the intrinsic parameters of the camera such as the field of view (or the focal length relative to the film size), the position of principal point, skew factor and lens distortion are not known beforehand. To reconstruct accurate 3D Euclidean structure and motion, these intrinsic parameters have to be found.

Assume that a perspective projection camera model is considered and the lens distortion can be ignored or is already known, the following calibration



matrix is of interest:

$$\mathbf{A} = \begin{bmatrix} fk_u & fk_u \cot \theta & u_0 \\ 0 & \frac{fk_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $f$  is the focal length of the camera in world coordinate units.  $k_u$  and  $k_v$  are the lengths in pixels of the unit length of the world coordinate system in the vertical and horizontal directions, respectively.  $u_0$  and  $v_0$  are the pixel coordinates of the principal point in the image plane.  $\theta$  is the angle between the vertical and horizontal axes in the image plane. Usually it is very close to  $\pi/2$ . Note that by writing the calibration matrix in the above form, we have moved the image plane to the front of the lens and have aligned the coordinate axes in the image plane with those in the world coordinate system.

A 3D point  $W$  has projection  $m$  in the image plane. Following the notation of Faugeras in [15], let  $\mathbf{w} = [X, Y, Z]^T$  be the world coordinates of  $W$  and  $\mathbf{m} = [u, v]^t$  be the pixel coordinates of its projection  $m$ . Let the homogeneous coordinates of any vector  $\mathbf{v} = [v_1, v_2, \dots, v_n]^T$  be  $\tilde{\mathbf{v}} = [v_1, v_2, \dots, v_n, 1]^T$ . Hence, in the homogeneous coordinate system, we have  $\tilde{\mathbf{w}} = [X, Y, Z, 1]^T$  and  $\tilde{\mathbf{m}} = [u, v, 1]^T$ .  $\tilde{\mathbf{m}}$  and  $\tilde{\mathbf{w}}$  are related by a  $3 \times 4$  projection matrix  $\tilde{\mathbf{P}}$

$$\lambda \tilde{\mathbf{m}} = \tilde{\mathbf{P}} \tilde{\mathbf{w}} \quad (2)$$

where  $\lambda$  is called projective depth and does not play any role in the location of  $m$  in the image plane. Hence (2) is often rewritten as

$$\tilde{\mathbf{m}} \simeq \tilde{\mathbf{P}} \tilde{\mathbf{w}} \quad (3)$$

by ignoring  $\lambda$ , where the symbol  $\simeq$  means that the two quantities are equal up to a scale factor. The projection matrix  $\tilde{\mathbf{P}}$  can be decomposed as

$$\tilde{\mathbf{P}} = \mathbf{A}[\mathbf{R} | -\mathbf{R}\mathbf{t}] \quad (4)$$

where  $\mathbf{A}$  is the calibration matrix and  $(\mathbf{R}, \mathbf{t})$  is the displacement of the camera, containing both rotation and translation.

The problem of self-calibration is to estimate the calibration matrix  $\mathbf{A}$  purely from an observed image sequence without any knowledge or control of the motion of the camera. In this paper, we will focus on the estimation of an unknown constant or varying FOV when all the other intrinsic parameters are given. We also assume that the camera moves continuously and takes many camera positions.

## 2.2 Critical Motion Sequences

In the research on solving the camera self-calibration problem, it has been observed that not all camera motion sequences lead to unique camera intrinsic parameters and 3D Euclidean scene reconstruction. Camera motion sequences that produce ambiguous calibrations are called *critical motion sequences*.

Identification of CMSs with various assumptions on the calibration has been systematically investigated in the literature. In [7], Sturm listed all CMSs when

the calibration parameters are constant. The CMSs for known calibration parameters, except for a varying FOV, can be found in [11]. Kahl also discussed CMSs when some intrinsic parameters can vary [10]. Other references on identification of CMSs can be found in [16,17,18]. Recall that we assume that the camera moves continuously and takes many camera positions. According to [10, 11], there are three types of CMSs for varying FOV that contain many camera positions:

- arbitrary translation with arbitrary rotation only about the optical axis
- translation along an ellipse or a hyperbola with the optical axis tangent to the ellipse or hyperbola
- translation along the optical axis with arbitrary rotation about the camera centers (at most two)

When the FOV is constant, only the first type of motion in the above CMS list is critical when many camera positions are present [8,11]. When the camera motion is not continuous, there are more critical motion sequences existing for the self-calibration of a camera with only an unknown FOV. Analyzing CMSs related to discontinuously moving cameras is beyond the scope this paper.

We mainly deal with the self-calibration ambiguities caused by the first kind of CMSs, since this kind of motion sequences are frequently encountered in practice. Because camera motion includes translation and rotation, the assumption of continuous camera motion implies that the rotation of the camera is also continuous if the camera rotates. Since the third type of CMSs contain at most two rotations about the camera centers, it rarely happens to a continuously rotating camera. Although the second type of CMSs is not explicitly considered in this paper, we have shown by experiments that it is possible to remove the self-calibration ambiguities caused by the type of CMSs if we assume that the 3D scene is rigid and non-planar.

To handle the first type of CMSs, we need to find out the transformations between true and false Euclidean reconstructions. Since the false Euclidean reconstruction is actually a projective reconstruction, it is different from the true Euclidean reconstruction by a projective transformation,  $\mathbf{T}_\Phi$  [19].

Assume that a false Euclidean reconstruction has been found. Let  $\Delta f^{(0)}$  be the ratio of the focal lengths of true and false Euclidean reconstructions in the initial time instant, i.e.  $\Delta f^{(0)} = \frac{f_e^{(0)}}{f_\Phi^{(0)}}$  where  $f_e^{(0)}$  and  $f_\Phi^{(0)}$  are the true and false focal lengths at the initial time instant, respectively. Let  $\mathbf{t}_\Phi$  and  $\mathbf{t}_e$  be the translation vectors associated with the false and true reconstructions, respectively. Let  $(\alpha_e, \beta_e)$  and  $(\alpha_\Phi, \beta_\Phi)$  be the translation direction angles associated with  $\mathbf{t}_e$  and  $\mathbf{t}_\Phi$ . The unit vector in the translation direction is given by  $(\sin(\alpha) \cos(\beta), \sin(\alpha) \sin(\beta), \cos(\alpha))^T$ . At any time instant, the true and false projection matrices are related by

$$\tilde{\mathbf{P}}_e = \tilde{\mathbf{P}}_\Phi \mathbf{T}_\Phi = \mathbf{A}_\Phi [\mathbf{R} | -\mathbf{R}\mathbf{t}_\Phi] \begin{bmatrix} \mathbf{T}_{11} & \mathbf{0}_3 \\ \mathbf{0}_3^T & \kappa \end{bmatrix} = \mathbf{A}_\Phi [\mathbf{R}\mathbf{T}_{11} | -\kappa\mathbf{R}\mathbf{t}_\Phi] \quad (5)$$

where

$$\mathbf{T}_{11} = \begin{bmatrix} \Delta f^{(0)} & 0 & 0 \\ 0 & \Delta f^{(0)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

and  $\theta$  is the rotation angle about the optical axis. Hence  $\mathbf{R}\mathbf{T}_{11} = \mathbf{T}_{11}\mathbf{R}$  and (5) can be written as

$$\tilde{\mathbf{P}}_e = A_\Phi[\mathbf{T}_{11}\mathbf{R} | -\kappa\mathbf{R}\mathbf{t}_\Phi] = A_\Phi\mathbf{T}_{11}[\mathbf{R} | -\mathbf{R}(\kappa\mathbf{R}^T\mathbf{T}_{11}^{-1}\mathbf{R}\mathbf{t}_\Phi)] \quad (7)$$

Hence,

$$\mathbf{t}_e = \kappa\mathbf{R}^T\mathbf{T}_{11}^{-1}\mathbf{R}\mathbf{t}_\Phi = \kappa\mathbf{T}_{11}^{-1}\mathbf{t}_\Phi \quad (8)$$

After some straightforward algebra, we have

$$\alpha_e = \arccos \frac{\Delta f \cos \alpha_\Phi}{\sqrt{(\Delta f \cos \alpha_\Phi)^2 + \sin^2 \alpha_\Phi}}, \beta_e = \beta_\Phi \quad (9)$$

Furthermore, the resulting rotation angles are all the same for different ambiguous reconstructions since the rotation matrix does not change when the false focal length is replaced by the true one. If the focal length is free to vary, the relationship between the true and false focal lengths at time  $i$  is given by  $f_e^{(i)} = \Delta f^{(0)} f_\Phi^{(i)}$ .

Regarding the transformations among the 3D structures, we have the following relationship.

$$\tilde{\mathbf{w}}_e = \mathbf{T}_\Phi^{-1}\tilde{\mathbf{w}}_\Phi = \begin{bmatrix} \mathbf{T}_{11}^{-1} & \mathbf{0}_3 \\ \mathbf{0}_3^T & \kappa^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_\Phi \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11}^{-1}\mathbf{w}_\Phi \\ \kappa^{-1} \end{bmatrix} \text{ and } \mathbf{w}_e = \kappa\mathbf{T}_{11}^{-1}\mathbf{w}_\Phi \quad (10)$$

The transformations of motion and scene structure reconstructions reveal the relationships among different reconstructions. It will be employed in the design of a novel self-calibration algorithm in the next section.

### 2.3 Sequential Importance Sampling

The SIS method has been recently proposed for approximating the posterior distribution of the state parameters of a dynamic system [13]. Usually, the state space model of a dynamic system is described by observation and state equations. Denote the measurement by  $\mathbf{y}_t$  and the state parameter by  $\mathbf{x}_t$ . The observation equation essentially provides  $f_t(\mathbf{y}_t|\mathbf{x}_t)$ , the conditional distribution of the observation given the state. Similarly, the state equation gives  $q_t(\mathbf{x}_{t+1}|\mathbf{x}_t)$ , the Markov transition distribution from time  $t$  to time  $t+1$ . Let  $\mathcal{X}_t = \{\mathbf{x}_i\}_{i=1}^t$  and  $\mathcal{Y}_t = \{\mathbf{y}_i\}_{i=1}^t$ . Samples drawn from  $\pi_t(\mathcal{X}_t) = P(\mathcal{X}_t|\mathcal{Y}_t)$ , the posterior distribution of the states given all the available observations up to  $t$ , are needed to compute the ensemble statistics such as mean or modes. However, to directly draw samples from a complex, high-dimensional distribution is very difficult in practice. An alternative way to the approximation of the posterior distribution is by a set of samples called *properly weighted samples* and their corresponding weights [13].

Suppose  $\{\mathcal{X}_t^{(j)}\}_{j=1}^m$  is a set of random samples properly weighted by the set of weights  $\{w_t^{(j)}\}_{j=1}^m$  with respect to  $\pi_t$  and let  $g_{t+1}$  be a trial distribution. Then the recursive SIS procedure to obtain the samples and weights properly weighting  $\pi_{t+1}$  is as follows.

*SIS steps:* for  $j = 1, \dots, m$ ,

(A) Draw  $X_{t+1} = \mathbf{x}_{t+1}^{(j)}$  from  $g_{t+1}(\mathbf{x}_{t+1}|\mathcal{X}_t^{(j)})$ . Attach  $\mathbf{x}_{t+1}^{(j)}$  to form  $\mathcal{X}_{t+1}^{(j)} = (\mathcal{X}_t^{(j)}, \mathbf{x}_{t+1}^{(j)})$ .

(B) Compute the “incremental weight”  $u_{t+1}$  by

$$u_{t+1}^{(j)} = \frac{\pi_{t+1}(\mathcal{X}_{t+1}^{(j)})}{\pi_t(\mathcal{X}_t^{(j)})g_{t+1}(\mathbf{x}_{t+1}|\mathcal{X}_t^{(j)})}$$

and let  $w_{t+1}^{(j)} = u_{t+1}^{(j)}w_t^{(j)}$ .

It can be shown [13] that  $\{\mathcal{X}_{t+1}^{(j)}, w_{t+1}^{(j)}\}_{j=1}^m$  is properly weighted with respect to  $\pi_{t+1}$ . Hence, the above SIS steps can be applied recursively to get the properly weighted set for any future time instant when the corresponding observations are available. The choice of the trial distribution  $g_{t+1}$  is very crucial in the SIS procedure since it directly affects the efficiency of the proposed SIS method. In our approach, we used  $g_{t+1}(\mathbf{x}_{t+1}|\mathcal{X}_t) = q_{t+1}(\mathbf{x}_{t+1}|\mathbf{x}_t)$ . It can be shown that in this case  $u_{t+1} \propto f(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})$ , which is the conditional probability density function of the observations at  $t+1$  given the state sample  $\mathbf{x}_{t+1}$  and it is also known as the likelihood function of  $\mathbf{x}_{t+1}$  since the observations are fixed.

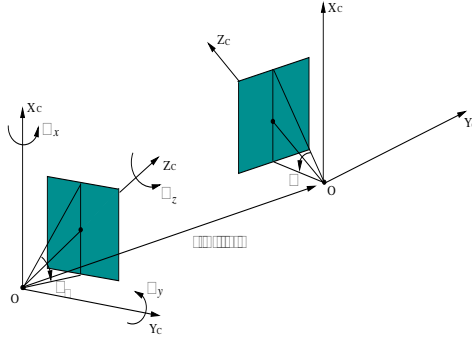
### 3 Bayesian Self-Calibration Using Sequential Importance Sampling

In this section, we design a camera self-calibration algorithm assuming that the camera has an unknown constant or varying focal length or equivalently FOV with all the other parameters given. Our goal is to find an algorithm that does not have any specific requirement for initialization and is able to detect and handle the CMSs. The performance of the algorithm should degrade gracefully as the noise level in the observations increases. SIS is used as the main computational framework because of its capability for solving problems involving non-linear systems.

#### 3.1 Parameterization of the Camera Motion

Before discussing the parameterization of sensor motion, we introduce two 3D Euclidean coordinate systems used in our research. One coordinate system is attached to the camera and uses the center of projection of the camera as its origin. It is denoted by  $C$ . The  $Z$  axis of  $C$  is along the optical axis of the camera, with the positive half-axis in the camera looking direction. The  $X$ - $Y$  plane of  $C$  is perpendicular to the  $Z$  axis with the  $X$  and  $Y$  axes parallel to the borders of the image plane. Also, the  $X$ - $Y$ - $Z$  axes of  $C$  satisfy the right-hand rule. The

other coordinate system is a world inertial frame, denoted by  $I$ .  $I$  is fixed on the ground. The coordinate axes of  $I$  are configured in such a way that initially,  $I$  and  $C$  coincide. When the camera moves,  $C$  travels with the camera and  $I$  stays at the initial position.



**Fig. 1.** Imaging model of a moving camera with an unknown FOV

Since the focal length (or FOV) is unknown,  $\gamma$  is used to represent the unknown focal length (or FOV). It is noted that the ranges of focal length and FOV are  $[0, \infty]$  and  $[0, \pi]$ , respectively. Since a sampling based procedure is to be used, a naturally bounded variable is preferred. Hence, instead of focal length, the vertical FOV of the camera is to be estimated in the algorithm. Let  $\gamma$  denote the unknown FOV. Based on the above discussion, the state vector describing both extrinsic (motion) and intrinsic (FOV) parameters could be defined as

$$\mathbf{x} = (\psi_x, \psi_y, \psi_z, \alpha, \beta, \gamma). \quad (11)$$

Here  $(\psi_x, \psi_y, \psi_z)$  are the rotation angles of the camera about the coordinate axes of the inertial frame  $I$  and  $(\alpha, \beta)$  are the elevation and azimuth angles of the camera translation direction, measured in the world system  $I$ .  $\gamma$  is the FOV of the camera. For simplicity, we still call  $\mathbf{x}$ , the motion parameter: remember that the FOV is now included in  $\mathbf{x}$ . If the FOV is free to change, one more component is added to the motion parameters.

$$\mathbf{x} = (\psi_x, \psi_y, \psi_z, \alpha, \beta, \gamma_0, \gamma) \quad (12)$$

where  $\gamma_0$  represents the FOV of the camera at the initial time instant and  $\gamma$  denotes the FOV at other time instants.

*State space model.* Given the above motion parameterization, a state space model can be used to describe the behavior of a moving camera.

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{n}_x \text{ and } \mathbf{y}_t = Proj(\mathbf{x}_t, \mathcal{S}_t) + \mathbf{n}_y \quad (13)$$

where  $\mathbf{x}_t$  is the state vector and  $\mathbf{y}_t$  is the observation at time  $t$ .  $Proj(\cdot)$  denotes the perspective projection, a function of camera motion  $\mathbf{x}_t$  and scene structure

$\mathcal{S}_t$ .  $n_x$  denotes the dynamic noise in the system, describing the time-varying property of the state vector. If no prior knowledge about motion is available, a random walk will be a suitable alternative for modeling the camera motion.

### 3.2 Processing Critical Motion Sequences

If a motion sequence is critical, the self-calibration algorithm should be able to detect the presence of CMSs. In our approach, CMS detection can be modeled as a hypothesis testing problem and the posterior probability of the non-criticalness of the motion sequence can be estimated. In the hypothesis testing problem, a binary variable  $I_C$  is introduced to indicate the presence of a CMS:

$$I_C = \begin{cases} 1, & \text{the motion sequence is critical} \\ 0, & \text{the motion sequence is not critical} \end{cases} \quad (14)$$

This hypothesis testing problem can be naturally embedded in the SIS procedure. In the current case of interest, only one class of CMS exists: motion sequences that do not contain any rotation about an axis parallel to the image plane. Therefore, if the motion sequence is critical, motion samples with rotation only about the optical axis are enough to interpret the trajectories of the feature points. On the other hand, if the motion sequence is not critical, motion samples with rotation about axes parallel to the image plane have to be used to interpret the feature trajectories. Hence, two sets of samples are involved in the SIS procedure. The set of samples with only rotation about the optical axis is denoted by  $\mathcal{X}_C$  since it can explain the feature trajectories in the image plane introduced by the CMS. The other set of samples is denoted by  $\mathcal{X}_G$  because it will be used to explain the feature trajectories caused by general motion sequences other than CMSs. In the initialization stage of SIS, samples are generated in these two sets. Because no knowledge of the criticalness of the motion sequence is available at the beginning, equal numbers of samples are used in the two sets. The weights of the samples can be computed directly using the formula derived in [14].

During the motion of the camera, the criticalness of the motion sequence can change. A critical motion sequence up to time  $t$  can become non-critical at time  $t + 1$  if rotation about axes parallel to the image plane is present at time  $t + 1$ . However, a non-critical motion sequence can never become critical. If the indicator  $I_C$  is viewed as the state of a dynamic system, this dynamic system can be characterized by a Markov chain. If the probability that a critical motion sequence becomes non-critical at time  $t$  is  $P_{C \rightarrow G}(t)$ , the state transition probabilities of the Markov chain are:

$$\begin{cases} P(I_C(t+1) = 0 | I_C(t) = 1) = P_{C \rightarrow G}(t) \\ P(I_C(t+1) = 1 | I_C(t) = 1) = 1 - P_{C \rightarrow G}(t) \\ P(I_C(t+1) = 0 | I_C(t) = 0) = 1 \\ P(I_C(t+1) = 1 | I_C(t) = 0) = 0 \end{cases} \quad (15)$$

To take this fact into account in SIS, when drawing new samples for time  $t + 1$  from samples for time  $t$ , the samples in  $\mathcal{X}_C$  need to be transferred to  $\mathcal{X}_G$  with probability  $P_{C \rightarrow G}(t)$ . This can be done by adding rotation components about

axes parallel to the image plane, which can be drawn from a trial distribution.  $P_{C \rightarrow G}(t)$  is unknown and no knowledge about it is available. Intuitively, 0.5 could be a good value for  $P_{C \rightarrow G}(t)$  for all  $t$  since it gives the maximum uncertainty to the occurrence of the transformation of the motion sequence from critical to non-critical.

In the SIS procedure, the sample-weight set describes the posterior distribution of the motion parameters:

$$\begin{aligned} P(\mathcal{X}_t | \mathcal{Y}_t) &= \sum_{I_C} P(\mathcal{X}_t, I_C | \mathcal{Y}_t) \\ &= P(\mathcal{X}_t | I_C = 1, \mathcal{Y}_t) P(I_C = 1 | \mathcal{Y}_t) + P(\mathcal{X}_t | I_C = 0, \mathcal{Y}_t) P(I_C = 0 | \mathcal{Y}_t) \end{aligned}$$

The samples in  $\mathcal{X}_G$  are properly weighted by their corresponding weights with respect to  $P(\mathcal{X}_t | I_C = 0, \mathcal{Y}_t)$ , the posterior distribution of the motion parameters conditional on the motion sequence is not critical. The posterior probability of the presence of the critical motion sequence,  $\pi_t(I_C = 1) = P(I_C = 1 | \mathcal{Y}_t)$ , can be obtained using the following theorem.

**Theorem 1.** *Assume that  $\{\mathcal{X}_C, \mathcal{X}_G\}$  is properly weighted by  $\{\mathcal{W}_C, \mathcal{W}_G\}$  with respect to  $P(\mathcal{X}_t | \mathcal{Y}_t)$ .  $\mathcal{X}_C$  is the sample set related to the hypothesis that a critical motion sequence is present and  $\mathcal{W}_C$  is the associated weight set. Then  $\pi_t(I_C = 1)$ , the posterior probability of criticalness of the given motion sequence, is given by*

$$\pi_t(I_C = 1) = \lim_{m \rightarrow \infty} \frac{\sum_{w_c \in \mathcal{W}_C} w_c}{\sum_{w_c \in \mathcal{W}_C} w_c + \sum_{w_g \in \mathcal{W}_G} w_g} \quad (16)$$

where  $m$  is the number of samples.

The proof of the theorem is very straightforward and it is omitted due to the page limitation.

### 3.3 The Algorithm

Based on the discussion in the last section, a Bayesian camera self-calibration algorithm using SIS can be designed. Before the algorithm is presented, one more issue needs to be addressed.

Samples in  $\mathcal{X}_C$  are to be transferred to  $\mathcal{X}_G$  with a certain probability. Let the sample-weight pairs before this transfer be  $\mathcal{S}_C = (\mathcal{X}_C, \mathcal{W}_C)$  and  $\mathcal{S}_G = (\mathcal{X}_G, \mathcal{W}_G)$ . The transfer is done by reducing the weights of the samples in  $\mathcal{S}_C$  to half of the original values such that the number of samples belonging to  $\mathcal{S}_C$  after resampling has been decreased by half. Then, the samples of  $\mathcal{S}_C$  are put into  $\mathcal{S}_G$  with the remaining half weights. Therefore, the new sample-weight sets after the transfer step become

$$\tilde{\mathcal{S}}_C = (\mathcal{X}_C, \frac{\mathcal{W}_C}{2}), \tilde{\mathcal{S}}_G = (\{\mathcal{X}_C, \mathcal{X}_G\}, \{\frac{\mathcal{W}_C}{2}, \mathcal{W}_G\}) \quad (17)$$

After the transfer of samples, resampling is done to the samples in  $\tilde{\mathcal{S}}_C$  to prepare samples for the next time instant. For samples in  $\tilde{\mathcal{S}}_G$ , two procedures will be

utilized. Resampling is applied to the samples from  $\mathcal{S}_G$ . A crucial procedure, called *uniforming*, is applied to the samples transferred from  $\mathcal{S}_C$  to  $\tilde{\mathcal{S}}_G$ . As we mentioned in the last section, due to the fact that only a finite number of samples are used to describe the posterior distribution of the parameters, the empirical distribution of the FOV is not uniform. Uniforming is used to explore the fact that the posterior distribution of the FOV should be uniform in  $[0, \pi]$  if the motion sequence is critical no matter how the FOV is distributed according to the empirical samples and weights. Let  $\mathcal{X}_{C \rightarrow G} = \{\mathbf{x}^{(j)}\}_{j=1}^k$  be the samples transferred with weights  $\mathcal{W}_{C \rightarrow G} = \{\frac{w^{(j)}}{2}\}_{j=1}^k$  in  $\tilde{\mathcal{S}}_G$ . These sample-weight pairs are denoted by  $\mathcal{S}_{C \rightarrow G} = (\mathcal{X}_{C \rightarrow G}, \mathcal{W}_{C \rightarrow G}) = (\mathcal{X}_C, \frac{\mathcal{W}_C}{2})$ . Assume that  $m$  samples are used in the SIS procedure. Uniforming applied to  $\mathcal{S}_{C \rightarrow G}$  can be done in the following way:

Uniforming. For  $j = 1, \dots, k$ ,

(A) Uniformly draw  $\Gamma = \{\gamma^{(i)}\}_{i=1}^{n_j}$ , samples of FOV from  $[0, \pi]$  where

$$n_j = \frac{mw^{(j)}}{2(\sum_{w_c \in \mathcal{W}_C} w_c + \sum_{w_g \in \mathcal{W}_G} w_g)}$$

For  $i = 1, \dots, n_j$

(B) Compute the associated focal length in units of the height of the film:  $f^{(i)} = (2 \tan \frac{\gamma^{(i)}}{2})^{-1}$ . Motion sample  $\mathbf{x}^{(j)}$  is used as a seed to produce more samples. Let the focal length associated with the FOV in  $\mathbf{x}^{(j)}$  be  $f^{(0)}$ . By using the transformation among ambiguous motion estimates derived in section 2, the camera motion parameters related to the current focal length  $f^{(i)}$  can be found directly using (9). A new motion sample can be formed as

$$\mathbf{x}_j^{(i)} = (0, 0, \Psi_z, \alpha_j^{(i)}, \beta, \gamma^{(i)}) \quad (18)$$

where  $\Psi_z$  and  $\beta$  are the corresponding components in  $\mathbf{x}^{(j)}$ . Hence,  $\mathcal{X}_{\mathcal{U}}^{(j)} = \{\mathbf{x}_j^{(i)}\}_{i=1}^{n_j}$  are the new samples obtained from seed  $\mathbf{x}^{(j)}$ .

If the focal length is free to vary, the associated sample value of  $\gamma_0$  needs to be changed properly. The new samples can be written as

$$\mathbf{x}_j^{(i)} = (0, 0, \Psi_z, \alpha_j^{(i)}, \beta, \gamma_0^{(i)}, \gamma^{(i)}) \quad (19)$$

where

$$\gamma_0^{(i)} = 2 \arctan \frac{f_0^{(0)} f^{(0)}}{f^{(i)}}, \quad f_0^{(0)} = \left( 2 \tan \frac{\gamma_0^{(0)}}{2} \right)^{-1} \quad (20)$$

and  $\gamma_0^{(0)}$  is the value of FOV in the seed sample  $\mathbf{x}^{(j)}$ .

(C)  $\{\mathcal{X}_{\mathcal{U}}^{(j)}\}_{j=1}^k$  contains the samples by uniforming the sample-weight pair  $\mathcal{S}_{C \rightarrow G}$ .

Based on the above discussion, the SIS procedure for Bayesian self-calibration proceeds as follows.



## Bayesian Camera Self-Calibration Using SIS

1. **Initialization.** Draw samples of the motion parameters  $\{\mathbf{x}_0^{(j)}\}_{j=1}^m$  from the initial distribution  $\pi_0$ .  $\pi_0$  describes the distribution of the motion parameters  $\mathbf{x}_0$  before the camera moves. The absence of camera motion does not imply that  $\mathbf{x}_0 = 0$ . Although the rotation angle vector  $\psi$  and the translational vector are zero, the translational angles can be uniformly distributed. Hence, in  $\{\mathbf{x}_0^{(j)}\}$ , the components of the rotation angles are all set to zero and the samples of  $\alpha$ ,  $\beta$  and  $\gamma$  (and  $\gamma_0$  if the focal length is free to change) are drawn from the uniform distribution in  $[0, \pi]$ ,  $[0, 2\pi]$  and  $[0, \pi]$ , respectively. Since all the samples are drawn from the exact posterior distributions, equal weights are assigned to these samples. Since at the moment, rotation angles are all zeros, all the current samples belong to  $\mathcal{X}_C$  and  $\mathcal{X}_G$  contains no samples. For  $t = 1, \dots, \tau$ :
2. **Sample transfer.** Two pairs of sample-weight sets are available:  $(\mathcal{X}_C, \mathcal{W}_C)$  and  $(\mathcal{X}_G, \mathcal{W}_G)$ . Transfer all samples in  $\mathcal{X}_C$  to  $\mathcal{X}_G$ , and assign half weight to each sample. Denote the sample-weight pair transferred from  $\mathcal{S}_C$  to  $\mathcal{S}_G$  by  $\mathcal{S}_{C \rightarrow G} = (\mathcal{X}_C, \frac{\mathcal{W}_C}{2})$ . The new sample-weight pairs after sample transfer are  $\tilde{\mathcal{S}}_C = (\mathcal{X}_C, \frac{2\mathcal{W}_C}{2})$  and  $\tilde{\mathcal{S}}_G = \{\mathcal{S}_{C \rightarrow G}, \mathcal{S}_G\} = (\{\mathcal{X}_C, \mathcal{X}_G\}, \{\frac{\mathcal{W}_C}{2}, \mathcal{W}_G\})$ .
3. **Resampling and uniforming.** Resample the samples in  $\tilde{\mathcal{S}}_C$  according to their associated weights.  $\hat{\mathcal{X}}_C$  is used to represent the set containing the resulting samples. For samples in  $\tilde{\mathcal{S}}_G$ , uniforming and resampling are applied to samples belonging to different sets. Uniforming is performed on the samples in  $\mathcal{S}_{C \rightarrow G}$ . The samples originally in  $\mathcal{X}_G$  are then resampled. The sample set produced by these two procedures is denoted by  $\hat{\mathcal{X}}_G$ . Since resampling and uniforming have been executed, all the samples in  $\hat{\mathcal{X}}_C$  and  $\hat{\mathcal{X}}_G$  have equal weights. Let  $\{\hat{\mathbf{x}}_{t-1}^{(j)}\}_{j=1}^m$  denote the current samples.
4. **Sample generation.**  
For  $j = 1, \dots, m$ :  
Draw  $\mathbf{x}_t^{(j)}$  from the distributions of  $\hat{\mathbf{x}}_{t-1}^j + \mathbf{n}_x$ . The following distributions can be used for the dynamic noises in the translation direction angles.  $n_\kappa \sim U(-\delta_\kappa, \delta_\kappa)$ ,  $\kappa \in \{\alpha, \beta\}$  where  $\delta_\alpha$  and  $\delta_\beta$  can be chosen as positive numbers. The distributions of the dynamic noises in the rotation angles depend on  $\hat{\mathbf{x}}_{t-1}^j$ . If  $\hat{\mathbf{x}}_{t-1}^j$  is in  $\hat{\mathcal{X}}_C$ , disturbances are only added to the  $Z$  component of the rotation angles with  $n_{\psi_z} \sim \mathcal{N}(0, \sigma_z)$ . Otherwise, dynamic disturbances can be added to all the three components of the rotation angles and the associated distributions can be  $n_{\psi_\iota} \sim \mathcal{N}(0, \sigma_\iota)$ ,  $\iota \in \{x, y, z\}$  where  $\delta_x$ ,  $\delta_y$  and  $\delta_z$  can also be chosen as some small positive numbers.
5. **Weight computation.** Compute the weights of the samples,  $\{w_t^{(j)}\}$  using the weight computation formulas derived in [14] (See equations (5), (6) and (7) in [14] for details). Notice that in this case, the computation of the positions of terminal points of the epipolar line  $l$  involves not only the extrinsic parameters of the camera motion, but also  $\gamma$ , the field of view. The resulting samples and their corresponding weights  $(\mathcal{X}_t^{(j)}, w_t^{(j)})$  are properly weighted with respect to  $\pi_t(\mathcal{X}_t)$ . If more image frames are available, go back to step 2.

*Inference of Depth Distribution.* By using the SIS procedure proposed above, the posterior distribution of the camera extrinsic and intrinsic parameters can be approximately described by the resulting samples and their corresponding weights. The inference of the posterior distribution of the depths,  $\pi_t(\mathbf{z}_t)$ , can be accomplished as follows. In [14], two algorithms are presented to find the posterior distribution of feature depths when all the camera calibration parameters are known. The discussion about the inference of  $\pi_t(\mathbf{z}_t)$  based on the results obtained on  $\pi_t(\mathcal{X}_t)$  is still valid here since nothing has changed except that the unknown FOV  $\gamma$  is included in the motion vector  $\mathbf{x}$ . Hence, the posterior distribution of the depths can be directly inferred using the samples and weights properly weighted with respect to the posterior motion (both extrinsic and intrinsic) distribution. Both algorithms developed in [14] can be used to find samples and weights properly weighted with respect to the posterior distribution of feature depths in this case except that the known constant value focal length used in [14] needs to be replaced by the values of FOV in motion samples.

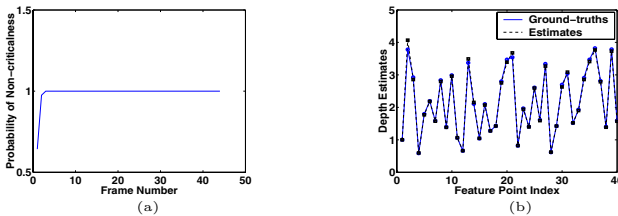
## 4 Experimental Results and Performance Analysis

By using the proposed algorithm for Bayesian self-calibration, constant or varying FOV can be recovered and furthermore, the motion of the camera and the scene structure can be reconstructed.

### 4.1 Constant Field of View

Two experimental results using synthetic image sequences are presented first. The synthetic feature trajectories are corrupted by additive white Gaussian noise (AWGN). In the first experiment, the standard deviation (STD) of the AWGN is 0.5 pixel. We consider this case as a nominal case.

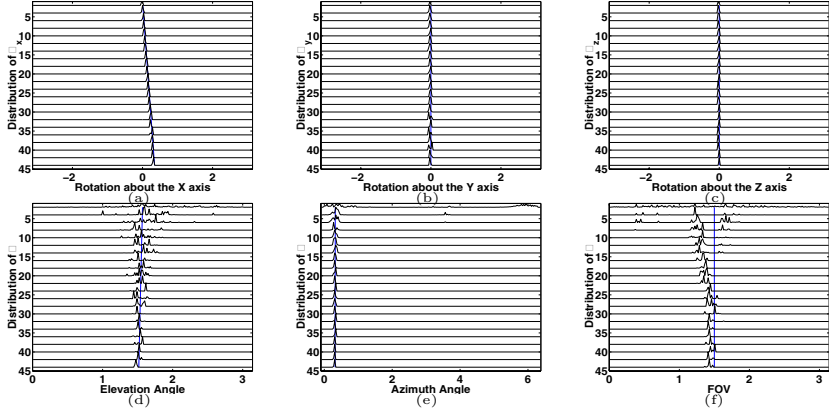
*A Nominal Case Study.* In this case, rotation about the  $X$  axis is present, hence the motion sequence is not critical.



**Fig. 2.** (a) shows the probability of non-criticalness of the motion sequence in the nominal case and (b) shows the MMSE estimates of the feature depths.

Figure 2 (a) shows the probability of the non-criticalness of the motion sequence. The horizontal axis of Figure 2 (a) is the time axis and the corresponding value on the vertical axis indicates the probability of the non-criticalness of the

motion sequence up to that time. It can be seen that this probability starts with a relatively low value at the beginning of the sequence. The reason is as follows. At the beginning of the sequence, the rotation about the  $X$  axis is small. Due to the observation noise in feature correspondences, the sequence looks like a critical motion sequence. Along with the increase in the rotation angle about the  $X$  axis, the probability of non-criticalness of the sequence approaches 1 eventually. This indicates that this motion sequence is not critical.

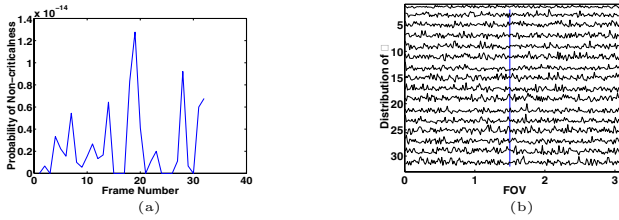


**Fig. 3.** Camera motion and calibration distributions in the nominal case

The motion distributions are shown in Figure 3. The ground-truths (including FOV) are indicated by the bold solid lines in Figure 3. Figures (a,b,c) show the distributions of the rotational angles  $\psi_x, \psi_y$  and  $\psi_z$ , respectively. The following two figures give the distributions of the translational angles  $\alpha$  and  $\beta$ . In each figure, the distribution of the corresponding motion parameter at each time instant is shown from the top of the figure to the bottom.  $(\psi_x, \psi_y, \psi_z)$  are in the range  $[-\pi, \pi]$ .  $\alpha$  is in  $[0, \pi]$  and  $\beta$  in  $[0, 2\pi]$ . All the other motion distribution results in this paper can be interpreted in the same way. We can see that the resulting posterior distributions of motion parameters have peaks very close to the ground-truths. Figure 3(f) shows the ground-truths and the posterior distributions of the field of view. Figure 2 (b) shows the ground-truths and the minimum mean square error (MMSE) estimates of the depths of the feature points. Since they are very close, it is difficult to distinguish one from the other.

*A Critical Motion Sequence.* Critical motion sequences were also generated to test the proposed algorithm. One example is included here. In this example, the virtual camera only translates along the horizontal axis without any rotation.

Figure 4 (a) shows the probability of non-criticalness of this motion sequence. It can be seen that the probability of non-criticalness of the camera stays at zero throughout the sequence, indicating that this motion sequence is critical. The ground-truth and posterior distribution of the FOV are shown in Figure 4 (b) and it can be seen that the FOV is nearly uniformly distributed.



**Fig. 4.** (a) shows the posterior probability of non-criticalness of the motion sequence. Since in the experiment the motion is pure translation, it can be seen that this probability is very close to zero, indicating that the motion sequence is critical. and (b) is the distribution of the FOV and it can be seen that the FOV is nearly uniformly distributed.

## 4.2 Freely Varying Field of View

Now let us look at examples when the FOV of the camera can freely vary.

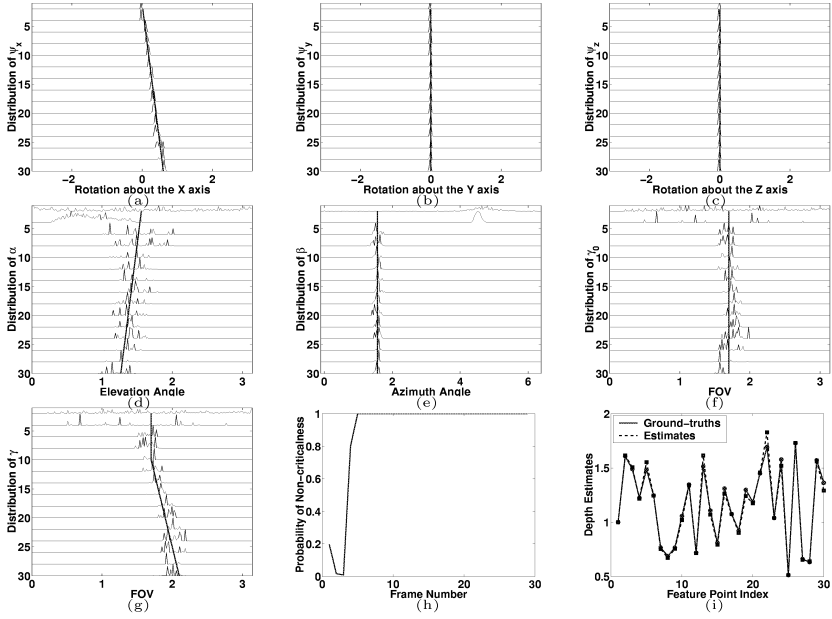
*Circular Camera Motion.* We let the virtual camera move along a circle. At the same time, the field of view of the camera is enlarged.

Figure 5 (a) - (g) shows the ground-truths and posterior distributions of the motion parameters. Figure 5 (h) shows the probability of non-criticalness of the motion sequence. It can be seen that this probability increases to 1 when more image frames are used and the rotation angles about the X axis increases. The MMSE estimates of depths are shown in Figure 5 (i).

*Elliptical Camera Motion.* In this example, we tested the proposed algorithm using an image sequence produced by a virtual camera moving along an ellipse with the optical axis of the camera tangent to the ellipse. Recall that this type of motion sequence was found critical [10,11], when the FOV can varying. The feature points are spread randomly in the 3D space and they are not on a plane. The feature correspondences are corrupted by AWGN with one-pixel of STD. By using the proposed approach, the posterior motion distribution of the camera can be approximated by a set of samples and their weights.

Figure 6 shows the refined estimates after applying the Levenberg-Marquardt non-linear optimization, using the result from the SIS algorithm as an initial guess. In Figure 6, the horizontal axis of each plot is the time axis. The solid lines show the ground-truths of the motion and calibration parameters of the camera and the dashed lines are the estimates of the parameters. It can be seen that the final results are very close to the ground-truths. Hence, it has been experimentally shown that it is possible to remove the calibration ambiguity introduced by the motion along an ellipse, which is the second type of CMS mentioned in Section 2.2.

*3D Face Modeling Using Uncalibrated Camera.* In this example, an image sequence containing 17 frames was captured using SunCamera II, which is an adjustable CCD color camera. The vertical FOV of SunCamera II is 33 degrees,



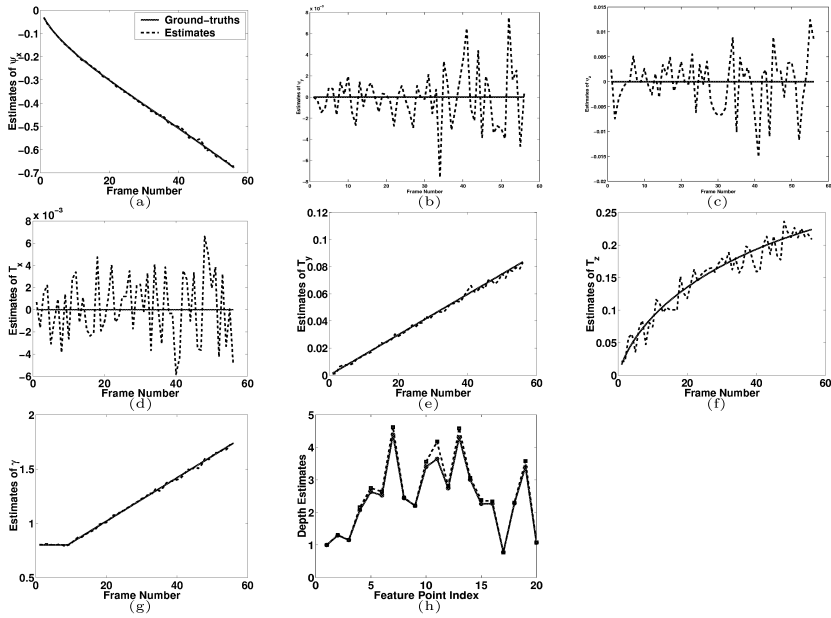
**Fig. 5.** The ground-truths and the posterior distributions of the camera motion and calibration parameters in the case of circular motion with varying focal length. (a)-(c) are the plots for camera rotation angles. (d) and (e) are the plots for camera translation direction angles. (f) is for the FOV of the camera at the initial time instance. (g) is for the varying FOV at different time instances. (h) shows the probability of non-criticalness of the motion sequence and (i) is the MMSE estimate of the feature depths.

which is equal to 0.576 radian. By using the proposed Bayesian self-calibration algorithm, the FOV of the camera can be accurately estimated and a 3D face model can be reconstructed. The MMSE estimate of the vertical fov is 0.5804 radian, which is very close to the ground-truth.

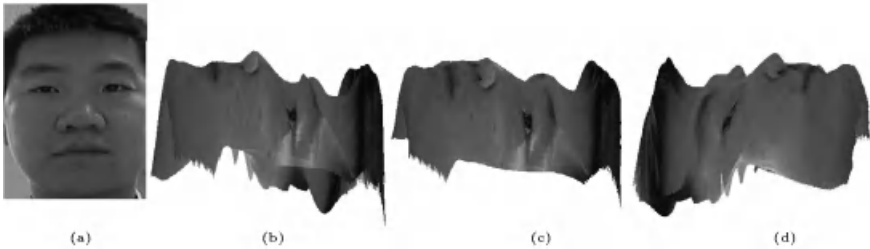
In Figure 7, (a) shows the texture map of this face sequence and (b)-(d) are the reconstructed 3D face model viewed from different angles.

## 5 Conclusions

In this paper, we have presented an algorithm for camera self-calibration using SIS. Our efforts have concentrated on the main problem of self-calibration: estimation of the FOV with all the other intrinsic parameters known, where the unknown FOV can be either constant or varying throughout the image sequence. The proposed algorithm is capable of processing the CMSs and quasi-CMSs and it does not have any specific requirements for initialization. The proposed method has also been tested extensively and satisfactory experimental results are obtained. A future research direction could be the extension of the algorithm



**Fig. 6.** The ground-truths and the estimates of the camera motion, calibration and structure parameters after non-linear optimization in the case of elliptical motion. (a)-(c) are the plots for camera rotation angles. (d)-(f) are the plots for camera translations. (g) shows ground-truth and estimate of the field of view of the camera. (h) is depth estimate.



**Fig. 7.** The texture map and reconstructed 3D model from an uncalibrated face sequence

to self-calibration with more unknown camera intrinsic parameters such as the position of the principle point and the aspect ratio.

## References

1. Faugeras, O., Luong, Q., Maybank, S.: Camera self-calibration: Theory and experiments. In: European Conference on Computer Vision, Santa Margherita Ligure, Italy. (1992) 321-334

2. Maybank, S., Faugeras, O.: A theory of self-calibration of a moving camera. *International Journal of Computer Vision* **8** (1992) 123–151
3. Fusiello, A.: Uncalibrated Euclidean reconstruction: a review. *Image and Vision Computing* **18** (2000) 555–563
4. Enciso, R., Vieville, T.: Self-calibration from four views with possibly varying intrinsic parameters. *Image and Vision Computing* **15** (1997) 293–305
5. Pollefeys, M., Koch, R., van Gool, L.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision* **32** (1999) 7–25
6. de Agapito, L., Hayman, E., Reid, I.: Self-calibration of a rotating camera with varying intrinsic parameters. In: *British Machine Vision Conference*, Southampton, UK. (1998)
7. Kahl, F., Heyden, A.: Euclidean reconstruction and auto-calibration from continuous motion. In: *International Conference on Computer Vision*, Vancouver, Canada. (2001) II: 572–577
8. Sturm, P.: Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In: *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR. (1997) 1100–1105
9. Sturm, P.: Critical motion sequences and conjugacy of ambiguous Euclidean reconstructions. In: *Scandinavian Conference on Image Analysis*, Lappeenranta, Finland. (1997)
10. Kahl, F., Triggs, B., Astrom, K.: Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision* **13** (2000) 131–146
11. Sturm, P.: Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. In: *British Machine Vision Conference*, Nottingham, UK. (1999) 63–72
12. Pollefeys, M., Van Gool, L.: Do ambiguous reconstructions always give ambiguous images? In: *International Conference on Computer Vision*, Vancouver, Canada. (2001) II: 187–192
13. Liu, J.S., Chen, R.: Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.* **93** (1998) 1032–1044
14. Qian, G., Chellappa, R.: Structure from motion using sequential Monte Carlo methods. In: *International Conference on Computer Vision*, Vancouver, Canada. (2001) II: 614–621
15. Faugeras, O.: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press (1993)
16. Kahl, F., Triggs, B.: Critical motions in Euclidean structure from motion. In: *IEEE Computer Vision and Pattern Recognition*, Fort Collins, CO,. (1999) II:366–372
17. Kahl, F.: Critical motions and ambiguous Euclidean reconstructions in auto-calibration. In: *International Conference on Computer Vision*, Corfu, Greece. (1999) 469–475
18. Kahl, F., Astrom, K.: Ambiguous configurations for the 1D structure and motion problem. In: *International Conference on Computer Vision*, Vancouver, Canada. (2001) I: 184–189
19. Hartley, R.: Projective reconstruction and invariants from multiple images. *PAMI* **16** (1994) 1036–1041

# Balanced Recovery of 3D Structure and Camera Motion from Uncalibrated Image Sequences

Bogdan Georgescu<sup>(1)</sup> and Peter Meer<sup>(1,2)</sup>

<sup>(1)</sup> Computer Science Department,

<sup>(2)</sup> Electrical and Computer Engineering Department  
Rutgers University, Piscataway, NJ, 08854-8058, USA  
georgesc, meer@caip.rutgers.edu

**Abstract.** Metric reconstruction of a scene viewed by an uncalibrated camera undergoing an unknown motion is a fundamental task in computer vision. To obtain accurate results all the methods rely on bundle adjustment, a nonlinear optimization technique which minimizes the reprojection error over the structural and camera parameters. Bundle adjustment is optimal for normally distributed measurement noise, however, its performance depends on the starting point. The initial solution is usually obtained by solving a linearized constraint through a total least squares procedure, which yields a biased estimate. We present a more balanced approach where in main computational modules of an uncalibrated reconstruction system, the initial solution is obtained from a statistically justified estimator which assures its unbiasedness. Since the quality of the new initial solution is already comparable with that of the result of bundle adjustment, the burden on the latter is drastically reduced while its reliability is significantly increased. The performance of our system was assessed for both synthetic data and standard image sequences.

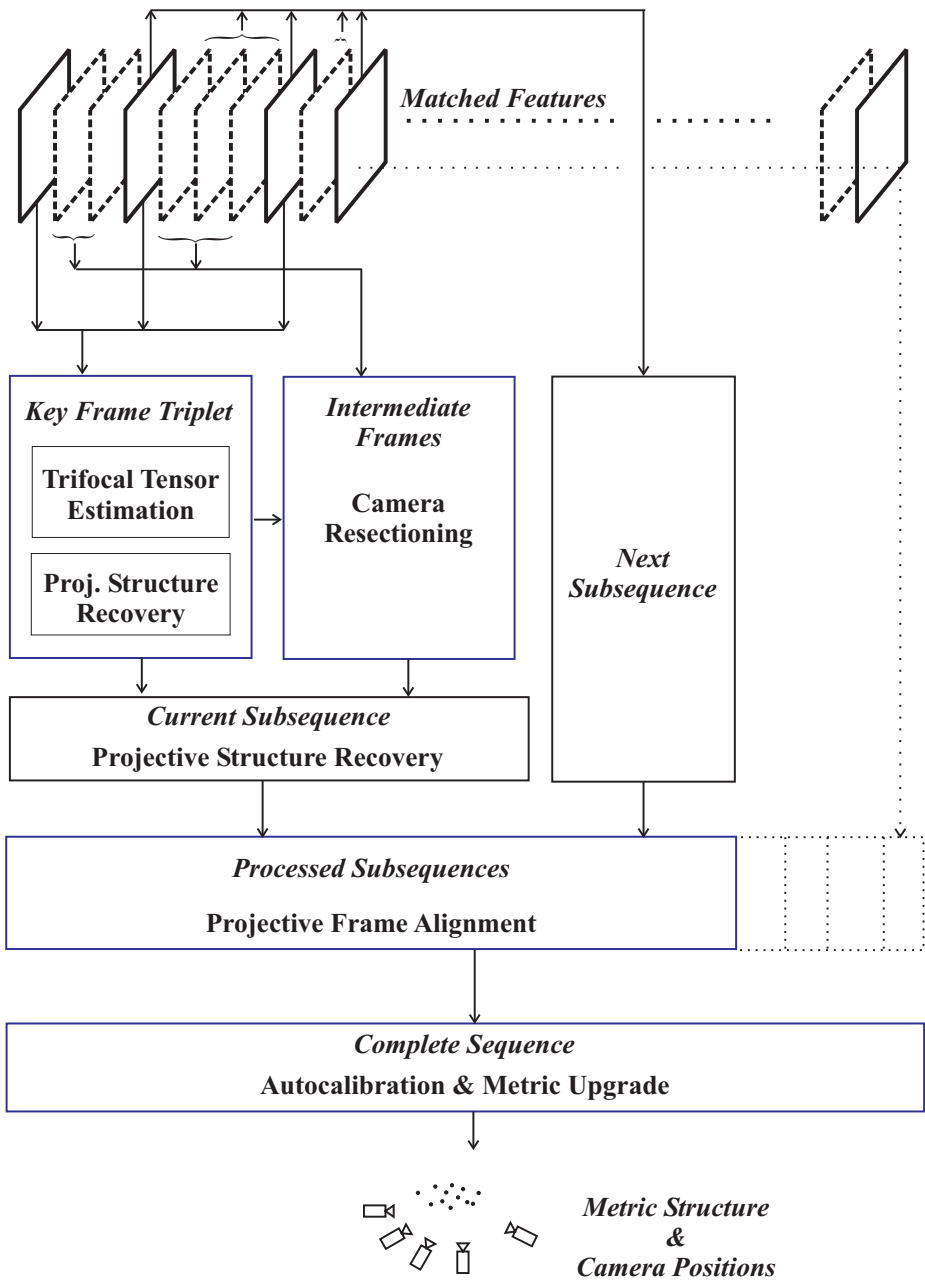
## 1 Introduction

Reliable analysis of image sequences captured by uncalibrated cameras is arguably the most significant progress in the recent years in computer vision. As the result of the analysis a 3D representation of the scene is obtained, which then can be used to acquire 3D models, generate new viewpoints, insert and delete objects, or determine the ego-motion for visual navigation. The technology became mature enough to support successful commercial ventures, such as REALVIZ or 2D3.

We follow a feature based approach toward uncalibrated image sequence analysis, in contrast with the brightness-based direct methods which consider the information from all the pixels in the image. Given an image sequence, first salient features are extracted from each frame and tracked across frames to establish correspondences. The analysis itself is a hierarchical process starting from groups of two or three images. Four main processing modules can be distinguished (Fig. 1).

1. Projective structure recovery from the key frames.
2. Insertion of the intermediate frames through camera resectioning.
3. Alignment of the independently processed subsequences.
4. Autocalibration and metric upgrade of the global reconstruction.





**Fig. 1.** The computational modules of an uncalibrated image sequence analysis system.

There is a large variety of methods proposed for each processing step. This paper will consider the most widely used techniques, described in [1], [4], [13], [14]. A nonlinear minimization problem has to be solved in each of the four modules, and most often bundle adjustment, e.g. [5], [21], is employed. Bundle adjustment is based on a sparse Levenberg-Marquardt procedure and minimizes the reprojection error over the whole set of unknown parameters, i.e., the camera matrices and the 3D structure. When the reprojection errors are normally distributed, bundle adjustment yields the optimal, maximum likelihood estimates.

The performance of any nonlinear optimization depends on the quality of the initial solution. Should this solution be too far from the true value, satisfactory convergence of the nonlinear procedure is no longer guaranteed. The traditional way to obtain the initial solution in the vision problems discussed here, is to apply a total least squares (TLS) procedure to a linearized constraint. It is well known, however, that this simple solution is biased since it fails to correctly account for the noise process that affects the *linearized* measurements, e.g. [8, p.77]. An empirical technique to improve the reliability of the linear solution is to first perform a normalizing transformation of the data [6].

In a more theoretical approach, the linearization process is analyzed and the estimation problem is put on solid theoretical foundations. The estimates obtained at the output of such methods are unbiased up to first order approximation. Kanatani's renormalization [10, pp.267–294] was the first technique from this class and was applied to a large variety of computer vision tasks. The heteroscedastic errors-in-variables (HEIV) model based estimation [11], defines the estimation problem somewhat differently than renormalization and has better numerical behavior. In this paper we show that by replacing the initial TLS based estimation step with a statistically more rigorous technique is advantageous and does not increase the amount of total computations. In fact, in some of the cases it can eliminate the need for bundle adjustment.

In Section 2 the different approaches toward solving the minimization problems arising in uncalibrated image sequence analysis are discussed. In Section 3 the performance of the four main modules are examined under two different initialization strategies: TLS and HEIV. The performance is assessed for synthetic data, while in Section 4 two standard image sequences are processed.

## 2 Nonlinear and Linear Minimization Techniques

Let  $\mathbf{m}_j, j = 1, \dots, n$ , be the available measurements, i.e., assumed to be the unknown true values additively corrupted with normal noise having covariance  $\mathbf{C}_{m_j}$ . In the estimation process the true values are replaced with the corrected measurements  $\hat{\mathbf{m}}_j$ , and the optimal (maximum likelihood) estimates can be obtained by minimizing

$$\mathcal{J}_M = \frac{1}{2} \sum_{j=1}^n (\mathbf{m}_j - \hat{\mathbf{m}}_j)^\top \mathbf{C}_{m_j}^+ (\mathbf{m}_j - \hat{\mathbf{m}}_j) \quad (1)$$

where  $\mathbf{C}_{m_j}^+$  is the pseudoinverse.

In the most straightforward approach, the dependence of the corrected measurements  $\hat{\mathbf{m}}_j$  on the parameter estimates  $\hat{\beta}$  is considered explicitly through a nonlinear vector

valued function, i.e.,  $\hat{\mathbf{m}}_j = \mathbf{f}_j(\hat{\boldsymbol{\beta}})$ . The resulting unconstrained nonlinear optimization problem is called *bundle adjustment*, and it is solved using the Levenberg-Marquardt method taking also into account the sparseness of the problem [21]. For example, if the  $\hat{\mathbf{m}}_j$ -s are the measured image points corresponding to the unknown 3D points, projected with cameras whose parameters are also unknown, the criterion (1) represents the sum of squared geometric distances under the suitable Mahalanobis metric.

An alternative way of capturing the a priori geometrical information is to consider an implicit relation (constraint), between  $\hat{\mathbf{m}}_j$  and  $\hat{\boldsymbol{\beta}}$ , i.e.,  $\mathbf{h}(\hat{\mathbf{m}}_j, \hat{\boldsymbol{\beta}}) = 0$ . The minimization criterion (1) becomes

$$\mathcal{J}_M = \frac{1}{2} \sum_{j=1}^n (\mathbf{m}_j - \hat{\mathbf{m}}_j)^\top \mathbf{C}_{m_j}^+ (\mathbf{m}_j - \hat{\mathbf{m}}_j) + \sum_{j=1}^n \boldsymbol{\eta}_j^\top \mathbf{h}(\hat{\mathbf{m}}_j, \hat{\boldsymbol{\beta}}) \quad (2)$$

where  $\boldsymbol{\eta}_j$  are the Lagrange multipliers. In most of the problems which arise in uncalibrated image sequence analysis, this constraint can be written as

$$\mathbf{h}(\hat{\mathbf{m}}_j, \hat{\boldsymbol{\beta}}) = \boldsymbol{\Phi}(\hat{\mathbf{m}}_j) \boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) = \mathbf{0}, \quad j = 1, \dots, n. \quad (3)$$

The data enters through the *carrier* matrix  $\boldsymbol{\Phi}$ , while the parameters are mapped through the vector valued *linearized parameter* function  $\boldsymbol{\theta}$ . The linear manifold structure of (3) is a consequence of the underlying projective geometry. Note that without loss of generality we can have  $\|\boldsymbol{\theta}\| = 1$ .

The existence of (3) motivated the use of a simple linear approximation to obtain  $\hat{\boldsymbol{\theta}}$ , the estimate of the linearized parameters. This estimate can then be used as initial solution for bundle adjustment. The *total least squares* (TLS) technique minimizing

$$\mathcal{J}_{TLS} = \sum_{j=1}^n \left\| \boldsymbol{\Phi}(\mathbf{m}_j) \hat{\boldsymbol{\theta}} \right\|^2 \quad (4)$$

i.e., the algebraic distance from the hyperplane with unit normal  $\hat{\boldsymbol{\theta}}$ , is most often employed. The TLS estimator, however, is optimal only when *all* the rows  $\phi_k$  of the carrier matrix are corrupted with the same noise process, which must have covariance  $\sigma^2 \mathbf{I}$  [22, Sec. 8.2]. This is not true for the estimation problems under consideration even when all  $\mathbf{C}_{m_j} = \sigma^2 \mathbf{I}$ , since the elements of the carrier matrix are nonlinear functions in the measurements  $\mathbf{m}_j$ .

Analyzing the structure of the carrier matrix  $\boldsymbol{\Phi}$  reveals that the noise process which has to be considered when (2) is minimized, is point dependent, i.e., heteroscedastic. The *heteroscedastic errors-in-variables* (HEIV) estimator described in [11] takes into account the nature of the noise process and finds  $\hat{\boldsymbol{\theta}}$  by solving iteratively the generalized eigenproblem

$$\nabla_{\hat{\boldsymbol{\theta}}} \mathcal{J}_M = [\mathbf{S}(\hat{\boldsymbol{\theta}}) - \mathbf{C}(\hat{\boldsymbol{\theta}})] \hat{\boldsymbol{\theta}} = \mathbf{0} \quad \text{subject to} \quad \|\hat{\boldsymbol{\theta}}\| = 1 \quad (5)$$

where

$$\mathbf{S}(\hat{\boldsymbol{\theta}}) = \sum_{j=1}^n \boldsymbol{\Phi}(\mathbf{m}_j)^\top \hat{\boldsymbol{\Sigma}}_j^+ \boldsymbol{\Phi}(\mathbf{m}_j) \quad \hat{\boldsymbol{\Sigma}}_j = \hat{\boldsymbol{\theta}}^\top \mathbf{J}_{\boldsymbol{\Phi}|\hat{\mathbf{m}}_j}^\top \mathbf{C}_{m_j} \mathbf{J}_{\boldsymbol{\Phi}|\hat{\mathbf{m}}_j} \hat{\boldsymbol{\theta}} \quad (6)$$

is the scatter matrix. The Jacobian matrices  $\mathbf{J}_{\Phi|\hat{\mathbf{m}}_j} = \partial\Phi(\hat{\mathbf{m}}_j)/\partial\hat{\mathbf{m}}_j$  can be easily computed since most of the elements of the carrier matrix have a multilinear structure. The expression of the weighted covariance matrix is

$$\mathbf{C}(\hat{\boldsymbol{\theta}}) = \sum_{j=1}^n \sum_{k,l} \eta_{kj} \eta_{lj} \left[ \frac{\partial \phi_k(\hat{\mathbf{m}}_j)}{\partial \hat{\mathbf{m}}_j} \right]^\top \mathbf{C}_{m_j} \left[ \frac{\partial \phi_l(\hat{\mathbf{m}}_j)}{\partial \hat{\mathbf{m}}_j} \right] \quad \eta_j = \hat{\boldsymbol{\Sigma}}_j^+ \Phi(\mathbf{m}_j) \hat{\boldsymbol{\theta}} \quad (7)$$

where  $\phi_k$  is the  $k^{th}$  row of  $\Phi$ . The corrected measurements are

$$\hat{\mathbf{m}}_j = \mathbf{m}_j - \mathbf{C}_{m_j} \mathbf{J}_{\Phi|\hat{\mathbf{m}}_j} \hat{\boldsymbol{\theta}} \hat{\boldsymbol{\Sigma}}_j^+ \Phi(\mathbf{m}_j) \hat{\boldsymbol{\theta}} \quad (8)$$

and analytical expressions for the covariances of the estimated parameters  $\mathbf{C}_{\hat{\boldsymbol{\theta}}}$  and the corrected measurements  $\mathbf{C}_{\hat{\mathbf{m}}_j}$  are also available [11, Sec. 5.2].

In spite of being computed iteratively, the performance of the HEIV estimation is not critically dependent of the initial choice of  $\hat{\boldsymbol{\theta}}$  in (5). Indeed, in most of the cases using a random initial  $\hat{\boldsymbol{\theta}}$  suffices. A more accurate starting value can be obtained by approximating the initial  $\mathbf{S}(\hat{\boldsymbol{\theta}})$  and  $\mathbf{C}(\hat{\boldsymbol{\theta}})$  from the available measurements  $\mathbf{m}_j$  and their covariances  $\mathbf{C}_{m_j}$  (usually taken as  $\sigma^2 \mathbf{I}$ ) [11, Sec. 5.6]. After each iteration the measurements are corrected (8) and the Jacobian matrices are updated followed by  $\mathbf{S}(\hat{\boldsymbol{\theta}})$  and  $\mathbf{C}(\hat{\boldsymbol{\theta}})$ . Convergence is usually reached after 3-4 iterations. It can be shown that  $\hat{\boldsymbol{\theta}}$  is an unbiased estimate at the first order approximation [11, Sec. 5.2].

The minimization criterion solved by the HEIV estimator is similar to that of the Sampson distance [8, Sec. 15.4.3]. However, traditionally when the Sampson distance is used, the solution is still obtained through the Levenberg Marquardt algorithm and the Jacobian matrices are not updated at each step [8, pp. 387–388].

The parameter of interest in the optimization is  $\beta$  and not  $\boldsymbol{\theta}$ . Since at each iteration of the HEIV estimation procedure the covariance of the current linearized parameter estimates  $\mathbf{C}_{\hat{\boldsymbol{\theta}}}$  is available,  $\hat{\boldsymbol{\theta}}$  can be further refined by imposing the constraint of its nonlinear dependence on  $\beta$  (3). This is achieved by projecting  $\hat{\boldsymbol{\theta}}$  under the metric induced by  $\mathbf{C}_{\hat{\boldsymbol{\theta}}}$  on the nonlinear manifold in the space of  $\beta$

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\| \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}(\beta) \right\|_{\mathbf{C}_{\hat{\boldsymbol{\theta}}}}^2 \quad (9)$$

which is solved by linearization [11, Sec. 5.10]. The parameter estimate  $\hat{\boldsymbol{\theta}}$  can now be updated as  $\hat{\boldsymbol{\theta}}^{(u)} = \boldsymbol{\theta}(\hat{\beta})$  and this value is used in the next iteration of the HEIV estimator.

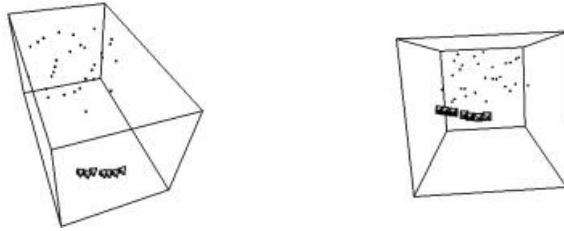
### 3 Uncalibrated Image Sequence Analysis System

The reconstruction of the 3D scene in the system described in this paper is based on point features detected in the images. The Harris corner detector was used, since it provides the most stable features under a wide range of operating conditions [16]. The correspondences across frames are established by the traditional normalized cross-correlation technique.

The first step in the analysis is to break down the sequence into several small groups of *key frames*. Since the “local” estimation of the projective structure employs the trifocal tensor, each group has three key frames. Given the first frame, the second and third frames are chosen to satisfy the trade-off between increasing the baseline of the group and having enough reliably tracked features. Adjacent key frames triplets have two frames in common (Fig. 1).

The three key frames delimit sets of contiguous *intermediate frames* in the sequence. Because the salient points were tracked also through these frames, the projection matrices for each intermediate frame can be computed by camera resectioning. The projective structure is then refined for the entire subsequence. The same process is applied independently to the next triplet of key frames, and the two subsequences are aligned by bringing them into the same projective basis.

After the entire available image sequence was processed and aligned, given that the camera motion is not degenerate, the metric structure of the 3D scene is recovered by imposing additional constraints on the internal camera parameters.



**Fig. 2.** Synthetic data. Two views of the typical configuration.

The role of the two initialization methods in the four computational modules will be assessed using synthetic data. Thirty 3D points were uniformly distributed in a cube and kept in the field of view in each of the 7 frames of size 512x512. The projected points were corrupted by normal noise with standard deviation  $\sigma = 0.5$  pixel units. Every second frame was taken as a key frame, thus having two subsequences of five frames each. The performance of the four modules was recorded in 100 trials. Between the trials the measurement noise is changed, and the position of the cameras was slightly perturbed by a random displacement. Two views of a typical experimental configuration is shown in Fig. 2. Note the small baseline of the camera movement which increases the difficulty of the processing. To assess the performance of an individual module, the output of bundle adjustment from the previous module was used.

### 3.1 Projective Structure Recovery from the Key Frames

The first computational module of the uncalibrated image sequence analysis system recovers the projective structure defined by triplets of key frames. The employed geometric constraint is based on the trifocal tensor which describes, independently of the scene structure, the intrinsic properties of the group of three images [17].

The incidence relation between the three point projections  $\{x, x', x''\}$  corresponding to the same 3D point can be written using the estimated  $3 \times 3 \times 3$  trifocal tensor  $T$  as

$$[x']_{\times} \left( \sum_{i=1}^3 x_i T_i \right) [x'']_{\times} = \mathbf{0}_{3 \times 3} \quad (10)$$

where  $[v]_{\times}$  is the skew-symmetric matrix such that  $v \times u = [v]_{\times} u$  and the  $3 \times 3$  matrices  $T_i$  are the correlation slices of the trifocal tensor [2]. The trifocal tensor is related to the projection matrices of the three frames  $P = [I|0]$ ,  $P' = [A|e']$  and  $P'' = [B|e'']$  by

$$T_i = a_i e''^{\top} - e' b_i^{\top}. \quad (11)$$

The constraint is satisfied by the *true* values of the quantities involved. For the estimation the relation (10) can be easily rewritten under the form (3)

$$\Phi_t(\hat{m}_t) \theta_t(\hat{\beta}_t) = \mathbf{0}_9 \quad \|\theta_t\| = 1 \quad (12)$$

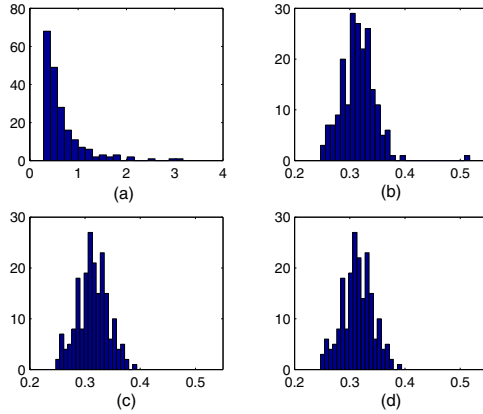
where the elements of the carrier matrix are products of the corrected image coordinates  $\hat{m}_t = [\hat{x}_1, \hat{x}_2, \hat{x}'_1, \hat{x}'_2, \hat{x}''_1, \hat{x}''_2]^{\top} \in \mathbb{R}^6$  and the components of the unconstrained parameter vector  $\theta_t \in \mathbb{R}^{27}$  are the estimates of the trifocal tensor elements. Each point correspondence contributes with 4 independent constraints [8, pp.417–421].

It can be shown that the trifocal tensor has only 18 degrees of freedom [2], and different parametrizations can be employed to constrain the 27 values of  $\theta_t$  to represent a tensor [18]. We have used the 24 parameters of the projection matrices  $P'$  and  $P''$  for parametrization, thus  $\beta_t \in \mathbb{R}^{24}$ . Note that only 22 parameters are significant due to the scale ambiguity of the projection matrices.

Since not all point correspondences are correct, the estimation process must be implemented robustly. Instead of the traditional RANSAC approach we have used one of its variants MLESAC [19], and minimized the transfer error, i.e., the robust sum of squared distances between the measurements and the corrected points. Subsequent computations were based only on the inliers. We have found that when the percentage of erroneous matches is small (say under 20%) a global M-estimation procedure is already satisfactory. This condition can be assured by using a high correlation score threshold.

The optimal, Gold Standard method for the recovery of the projective structure from a triplet of the key frames, is to apply bundle adjustment over the camera parameters and the 3D position of each feature [3], [18]. The initial solution is computed by recovering the camera matrices from the tensor (11), and using this information to obtain the 3D coordinates of each point by triangulation. The initial solution was computed with either TLS using normalized image coordinates [6] or HEIV. In the latter case, the corrected measurements  $\hat{m}_t$  are also available (8) and the camera parameters are obtained using the estimation process (9).

The performance was assessed through the reprojection error, i.e., the root-mean-squared (RMS) residual error which is proportional to the square root of the optimization criterion value (1). In Fig. 3 the histograms of the reprojection errors are shown for the different processing methods. As expected, the linear TLS solution is of poor quality being strongly biased (Fig. 3a), though bundle adjustment succeeds to eliminate this bias (Fig. 3b). The HEIV solution (Fig. 3c), on the other hand, already returns the same



**Fig. 3.** Reprojection errors for the trifocal tensor estimation from the key frames. (a) TLS initial solution. Note the different scale from HEIV. (b) Bundle adjustment initiated with TLS. (c) HEIV initial solution. (d) Bundle adjustment initiated with HEIV.

estimate as the Gold Standard method, subsequent bundle adjustment is not necessary since it will not yield any improvement (Fig. 3d). The number of bundle adjustment iterations for the TLS initialization was on average 5.3 but with high variation ( $\sigma_{it} = 6.5$ ) while using the HEIV initialization no additional iterations were needed. See [12] for a detailed discussion about using HEIV method for the trifocal tensor estimation.

### 3.2 Insertion of the Intermediate Frames

To complete the projective structure estimation for the entire subsequence defined by the three selected key frames, the information provided by the intermediate frames must be also integrated. Will denote with  $\mathbf{X}$  the projective coordinates of the 3D points, whose image was tracked *through* all the intermediate frames. From the processing of the key frames, the estimates of these 3D points are already available. Thus, using camera resectioning [8, pp. 166–170] the initial solution for the camera matrices of the intermediate frames can be determined.

The projective image formation relation  $\mathbf{x} \sim \mathbf{P}\mathbf{X}$  can be rewritten as

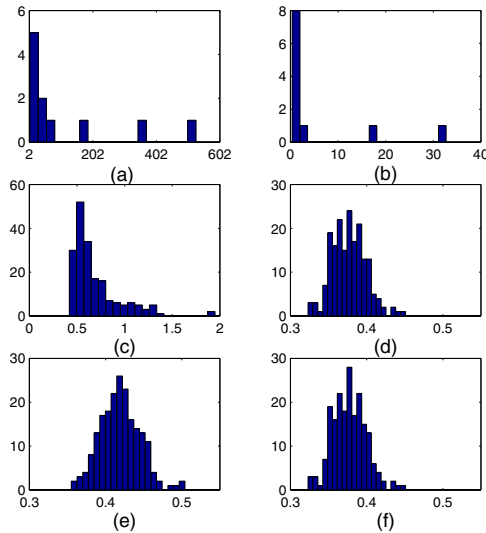
$$\begin{bmatrix} \mathbf{0}_4^\top & -x_3\mathbf{X}^\top & x_2\mathbf{X}^\top \\ x_3\mathbf{X}^\top & \mathbf{0}_4^\top & -x_1\mathbf{X}^\top \\ -x_2\mathbf{X}^\top & x_1\mathbf{X}^\top & \mathbf{0}_4^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}^{(1)} \\ \mathbf{p}^{(2)} \\ \mathbf{p}^{(3)} \end{bmatrix} = \mathbf{0}_3 \quad (13)$$

where we denote by  $\mathbf{p}^{(k)\top}$  the  $k^{th}$  row of the camera matrix  $\mathbf{P}$ . Each measurement contributes with two linear independent equations, and thus the images of at least six 3D points must be available. From the constraint (13), for the estimation we obtain an expression which has the form (3)

$$\Phi_r(\hat{\mathbf{m}}_r)\hat{\boldsymbol{\theta}}_r = \mathbf{0}_3 \quad (14)$$

where the carrier matrix  $\Phi_r$  has as elements products between the corrected projective image and 3D coordinates,  $\hat{\mathbf{m}}_r = [\hat{\mathbf{x}}^\top, \hat{\mathbf{X}}^\top]^\top$ , and  $\hat{\boldsymbol{\theta}}_r = \text{vec}[\hat{\mathbf{P}}^\top]$  are the elements of the projection matrix to be estimated.

The projection matrices are estimated for each of the intermediate frames initially by TLS and HEIV. Note that the HEIV based estimation takes into account that the “measurements” of the 3D points  $X_i$  (estimated in the first module) have covariances  $C_{X_i}$ . The entire subsequence, defined by the key frames and the intermediate frames, is passed to the bundle adjustment which refines globally the camera parameters and the 3D points coordinates.



**Fig. 4.** Reprojection errors for the entire subsequence. (a) TLS initial solutions, large errors. (b) Bundle adjustment initiated with the estimates in (a). (c) TLS initial solution. Note the different scale from HEIV. (d) Bundle adjustment initiated with TLS. (e) HEIV initial solution. (f) Bundle adjustment initiated with HEIV.

Results obtained with TLS initialization are presented in Figs. 4a and 4c, while the output of the corresponding bundle adjustment is shown in Figs. 4b and 4d. A few of the TLS initializations fail yielding RMS errors larger than 2 (Fig. 4a). Bundle adjustment did not succeed to recover from all of these cases (Fig. 4b). The bias of the TLS initial solution is also visible in Fig. 4c but it is removed after bundle adjustment (Fig. 4d). The HEIV initial solution is unbiased and yields smaller errors than the TLS (Fig. 4e). Since bundle adjustment is a global procedure, the errors are further reduced (Fig. 4f). It should be emphasized that after bundle adjustment both initializations give the same results (except the few failures of TLS) but fewer iterations were needed for bundle adjustment to converge for the HEIV initialization (average 3.5 with  $\sigma_{it} = 0.8$ ) than using the TLS initialization (average 5.1 but with large  $\sigma_{it} = 6.1$ ).

In the same framework we can also approach the triangulation procedure, i.e. finding the location of the 3D projective coordinates of a point knowing its projection in several images *and* the camera matrices. While it was not used in the performance comparisons with synthetic data, triangulation is an important step in the analysis of real image



sequences to obtain additional point correspondences and augment the available structure [7].

For triangulation, in (13) the parameters become the 3D projective point coordinates, and the “measurements” are the image points and camera matrices

$$\Phi_g(\mathbf{m}_g)\boldsymbol{\theta}_g = \begin{bmatrix} x_1\mathbf{p}^{(3)\top} - x_3\mathbf{p}^{(1)\top} \\ x_2\mathbf{p}^{(3)\top} - x_3\mathbf{p}^{(2)\top} \end{bmatrix} \mathbf{X} = \mathbf{0}_2. \quad (15)$$

The HEIV estimation takes into account the nonlinearities present in the carrier matrix, i.e., the products of image coordinates and camera matrix elements.

### 3.3 Alignment of the Independently Processed Subsequences

After a subsequence was processed, the newly obtained structure must be aligned with the already recovered structure. This can be achieved since there are at least two frames overlapping with the previous subsequence.

Assume that the frame  $j$  is shared by both subsequences and an image point  $\mathbf{x}_{ij}$  from this frame corresponds to the 3D point having the coordinates  $\mathbf{X}_i$  and  $\mathbf{X}'_i$  in the projective base of the two subsequences. Then the homography  $\mathbf{H}$  that aligns the subsequences must obey

$$\mathbf{x}_{ij} \sim \mathbf{P}_j \mathbf{X}_i \sim \mathbf{P}'_j \mathbf{H} \mathbf{H}^{-1} \mathbf{X}'_i \quad \text{or} \quad \mathbf{P}_j \sim \mathbf{P}'_j \mathbf{H} \quad \mathbf{X}_i \sim \mathbf{H}^{-1} \mathbf{X}'_i \quad (16)$$

where  $\mathbf{P}_j$  and  $\mathbf{P}'_j$  are the projective matrices of the frame  $j$  in the two bases.

Different methods allowing linear solutions for  $\mathbf{H}$ , based on direct 3D point registration, which is not meaningful in a projective framework, enforcing camera consistency, or a combination of these two were proposed [4]. We use the reprojection error between  $\mathbf{P}'_j \mathbf{H} \mathbf{X}_i$  and the corresponding image coordinates  $\mathbf{x}_{ij}$ . From (16) this constraint is

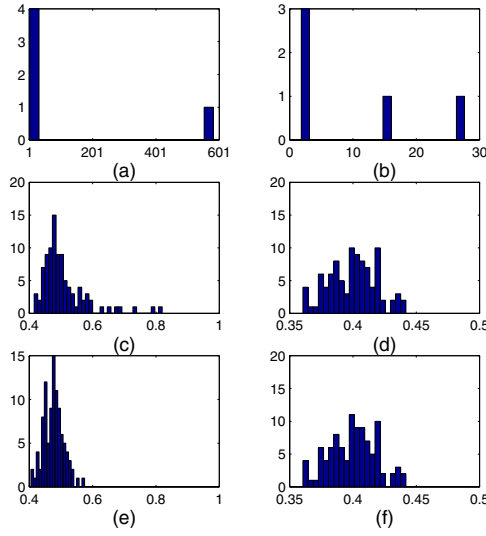
$$\mathbf{x}_{ij} \sim \mathbf{P}'_j \mathbf{H} \mathbf{X}_i \quad (17)$$

which can be expressed for estimation as

$$\Phi_h(\hat{\mathbf{m}}_h) \hat{\boldsymbol{\theta}}_h = \mathbf{0}_2 \quad (18)$$

where the elements of the carrier matrix  $\Phi_h$  are triple products of the corrected image coordinates, projection matrix elements and projective coordinates of the 3D points,  $\hat{\mathbf{m}}_h^\top = [\hat{\mathbf{x}}^\top, \text{vec}[\hat{\mathbf{P}}^\top]^\top, \hat{\mathbf{X}}^\top]$  and  $\hat{\boldsymbol{\theta}}_h$  contains the 16 components of the homography  $\mathbf{H}$  to be estimated.

The TLS initialization started from the results of the TLS based bundle adjustment from the previous module. Thus the few cases yielding large residual errors were also considered. Some of these cases were successfully processed by alignment, however, other new ones were introduced (Figs. 5a and 5b). The HEIV initialization used the estimated covariance matrices  $\mathbf{C}_{\hat{\mathbf{P}}}$ , and  $\mathbf{C}_{\hat{\mathbf{X}}}$  of the projection matrices and 3D points. No failures were obtained and its performance (Fig. 5e) is further refined by the global bundle adjustment (Fig. 5f). The average number of iterations using the TLS initialization was 6.4 with  $\sigma_{it} = 9.42$  while using the HEIV initialization the average was 3.9 with  $\sigma_{it} = 1.6$ .



**Fig. 5.** Reprojection errors for alignment of subsequences. (a) TLS initial solutions, large errors. (b) Bundle adjustment initiated with the estimates in (a). (c) TLS initial solution. (d) Bundle adjustment initiated with TLS. (e) HEIV initial solution. (f) Bundle adjustment initiated with HEIV.

### 3.4 Autocalibration and Metric Upgrade of the Global Reconstruction

The autocalibration method used in this paper is based on the dual absolute quadric  $\Omega^*$  and its relation to the dual image of the absolute conic  $\omega_j^*$  [20]

$$\omega_j^* \sim K_j K_j^\top \sim P_j \Omega^* P_j^\top \quad (19)$$

where  $K_j$  are the internal camera parameters for frame  $j$ . When additional knowledge about the internal parameters is available, such as no skew, known principal point or aspect ratio, then relation (19) can be used to obtain constraints on the dual absolute quadric [9], [14], [15]. If we assume that the aspect ratio is one, the skew is zero and the principal point is in the center of the image, then (19) yields four linear independent equations

$$\begin{aligned} p_j^{(1)\top} \Omega^* p_j^{(1)} &= p_j^{(2)\top} \Omega^* p_j^{(2)} \\ p_j^{(i)\top} \Omega^* p_j^{(k)} &= 0 \quad (i, k) \in \{(1, 2), (1, 3), (2, 3)\} \end{aligned} \quad (20)$$

which can be rearranged for estimation as in (3)

$$\Phi_a(\hat{m}_a) \hat{\theta}_a = 0_4 \quad (21)$$

where the carriers  $\Phi$  have as elements double products of the projection matrix elements,  $\hat{m}_a = \text{vec}[\hat{P}^\top] \in \mathbb{R}^{12}$  and  $\hat{\theta}_a \in \mathbb{R}^{10}$  contains the dual absolute quadric elements to be estimated. Because of the symmetry only 10 such elements are needed.

If one of the projection matrices is chosen as reference  $P_0 = [I|0_3]$  then  $\Omega^*$  becomes

$$\Omega^* = \begin{bmatrix} K_0 K_0^\top & -K_0 K_0^\top \pi_\infty \\ -\pi_\infty^\top K_0 K_0^\top & -\pi_\infty^\top K_0 K_0^\top \pi_\infty \end{bmatrix} \quad (22)$$

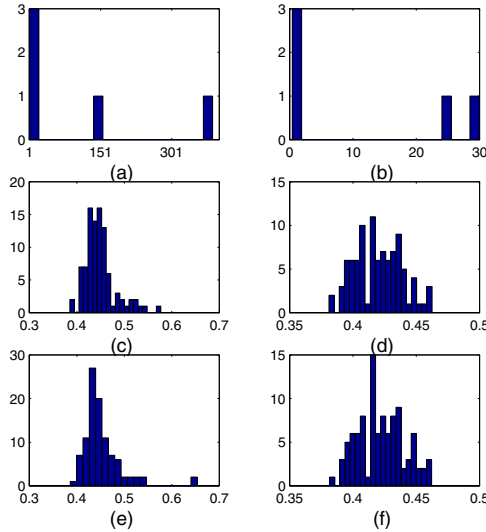
where  $\pi_\infty$  defines the plane at infinity. Thus  $\Omega^*$  can be parametrized by maximum 8 parameters (3 from the plane at infinity and the rest from  $K_0$ ). The following transformation brings the recovered projective structure into a metric reconstruction

$$H = \begin{bmatrix} K_0 & 0 \\ -\pi_\infty K_0^\top & 1 \end{bmatrix}. \quad (23)$$

Following [15] the TLS initial solution was further refined by solving with Levenberg-Marquardt the nonlinear least squares problem

$$\sum_j \left\| \frac{K_j K_j^\top}{\|K_j K_j^\top\|_F} - \frac{P_j \Omega^* P_j^\top}{\|P_j \Omega^* P_j^\top\|_F} \right\|_F^2 \quad (24)$$

where  $\|\cdot\|_F$  is the Frobenius norm of a matrix. The refinement of HEIV initialization was based on the nonlinear correction (9) which provided the estimates of the parametrization (22).



**Fig. 6.** Reprojection errors for metric upgrade of the entire structure. (a) Refined TLS initial solutions, large errors. (b) Bundle adjustment initiated with the estimates in (a). (c) Refined TLS initial solution. (d) Bundle adjustment initiated with refined TLS. (e) HEIV initial solution. (f) Bundle adjustment initiated with HEIV.

The projective reconstruction is upgraded to a metric reconstruction using the homography computed from (23) and bundle adjustment is employed over all the available 3D

points, internal and external camera parameters. The rotation matrices were parametrized with quaternions.

The results before and after metric bundle adjustment are presented in Fig. 6. The HEIV based initial solution (Fig. 6e) has similar performance to the combined TLS and nonlinear LS solution for the majority of the data (Fig. 6c) while not having the spurious large residual errors (Fig. 6a).

## 4 Experimental Results with Image Sequences

The system using the HEIV initialization was run on two well known real image sequences. Fig. 7 shows two images from the *MOVI house* image sequence and two poses of the reconstructed scene and camera positions. The sequence has 118 frames of a scene taken by moving the objects on a turntable. Significant illumination changes appear in the sequence because the objects were moved with respect to the light source. Note also that the density of the frames is not uniform. The reconstruction was computed automatically and without imposing constraints on the camera motion. After metric upgrade, the cameras that were close in 3D were used to establish additional correspondences which helped to improve the alignment of the entire sequence. It can be seen that the camera positions are lying on a planar circular path while keeping the scene in the field of view. The reconstructed position in 3D space of the scene features obey the rectangular shape of the house and lay on circular surfaces for the can and cup.

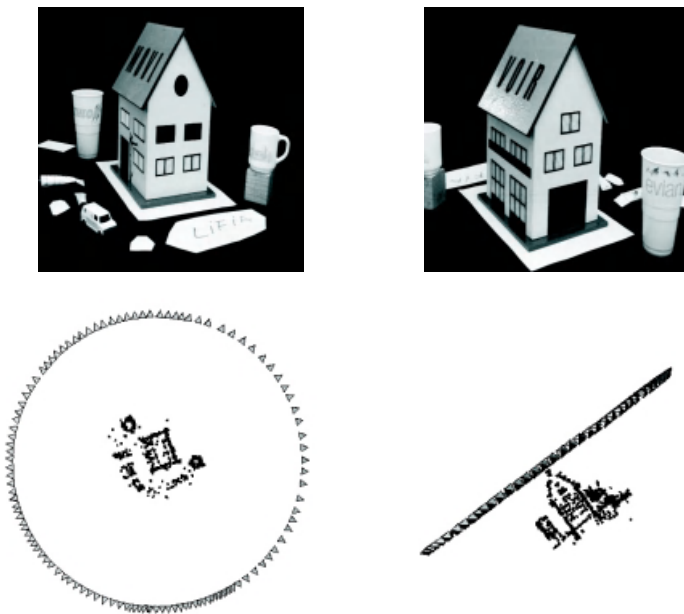


Fig. 7. Metric reconstruction of the MOVI house sequence.



Fig. 8. Metric reconstruction of the castle sequence

The second image sequence processed was the *castle* sequence. In Fig. 8 two images from the sequence and the reconstructed structure are shown. The sequence is 27 frames long and contains also some small nonrigid elements. The metric reconstruction successfully recovers the main features of the scene

## 5 Conclusion

We have presented a detailed investigation of the importance of using a statistically accurate initialization procedure in all the processing modules of an uncalibrated image sequence analysis system. The reliability of the system is further increased, and the failures for difficult data may be avoided.

**Acknowledgment.** The support of the NSF grant IRI 99-87695 is gratefully acknowledged. We thank Dr. Bogdan Matei from Sarnoff Corporation for insightful discussions.

## References

1. P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV 1996*, volume II, pages 683–695, Cambridge, UK, April 1996. Springer.
2. N. Canterakis. A minimal set of constraints for the trifocal tensor. In D. Vernon, editor, *Computer Vision – ECCV 2000*, volume I, pages 84–99, Dublin, Ireland, 2000. Springer.

3. O. Faugeras and T. Papadopoulos. A nonlinear method for estimating the projective geometry of 3 views. In *6th International Conference on Computer Vision*, pages 477–484, Bombay, India, January 1998.
4. A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV 1998*, volume I, pages 311–326, Freiburg, Germany, June 1998. Springer.
5. R. I. Hartley. Euclidean reconstruction from uncalibrated views. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, pages 237–256, 1994.
6. R. I. Hartley. In defence of the 8-point algorithm. In *5th International Conference on Computer Vision*, pages 1064–1070, Cambridge, MA, June 1995.
7. R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68:146–157, 1997.
8. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
9. A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *1997 IEEE Conference on Computer Vision and Pattern Recognition*, pages 438–443, San Juan, Puerto Rico, June 1997.
10. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, 1996.
11. B. Matei. *Heteroscedastic Errors-In-Variables Models in Computer Vision*. PhD thesis, Department of Electrical and Computer Engineering, Rutgers University, 2001. Available at <http://www.caip.rutgers.edu/riul/research/theses.html>.
12. B. Matei, B. Georgescu, and P. Meer. A versatile method for trifocal tensor estimation. In *8th International Conference on Computer Vision*, volume II, pages 578–585, Vancouver, Canada, July 2001.
13. P.F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *5th International Conference on Computer Vision*, pages 314–320, Cambridge, Massachusetts, June 1995.
14. M. Pollefeys. *Self-calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, K. U. Leuven, 1999.
15. M. Pollefeys. Self calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *International J. of Computer Vision*, 32:7–25, 1999.
16. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Computer Vision and Image Understanding*, 78:151–172, 2000.
17. A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 17:779–780, 1995.
18. P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, August 1997.
19. P.H.S. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *6th International Conference on Computer Vision*, pages 727–732, Bombay, India, January 1998.
20. B. Triggs. Autocalibration and the absolute quadric. In *1997 IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, San Juan, Puerto Rico, June 1997.
21. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment — A modern synthesis. In B. Triggs, A. Zisserman, and R. Szelisky, editors, *Vision Algorithms: Theory and Practice*, pages 298–372. Springer, 2000.
22. S. Van Huffel and J. Vanderwalle. Analysis and properties of GTLS in problem  $AX \approx B$ . *SIAM Journal on Matrix Analysis and Applications*, 10:294–315, 1989.

# Linear Multi View Reconstruction with Missing Data

Carsten Rother and Stefan Carlsson \*

Computational Vision and Active Perception Laboratory (CVAP)  
Dept. of Numerical Analysis and Computer Science  
KTH, SE-100 44 Stockholm, Sweden  
{carstenr, stefanc}@nada.kth.se

**Abstract.** General multi view reconstruction from affine or projective cameras has so far been solved most efficiently using methods of factorizing image data matrices into camera and scene parameters. This can be done directly for affine cameras [18] and after computing epipolar geometry for projective cameras [17]. A notorious problem has been the fact that these factorization methods require all points to be visible in all views. This paper presents alternative algorithms for general affine and projective views of multiple points where a) points and camera centers are computed as the nullspace of one linear system constructed from all the image data b) only three points have to be visible in all views. The latter requirement increases the flexibility and usefulness of 3D reconstruction from multiple views. In the case of projective views and unknown epipolar geometry, an additional algorithm is presented which initially assumes affine views and compensates iteratively for the perspective effects. In this paper affine cameras are represented in a projective framework which is novel and leads to a unified treatment of parallel and perspective projection in a single framework. The experiments cover a wide range of different camera motions and compare the presented algorithms to factorization methods, including approaches which handle missing data.

**Keywords:** Structure from Motion, Linear Multiple View Reconstruction, Missing Data, Affine and Projective Cameras.

## 1 Introduction

Efficient 3D reconstruction from multiple camera views is a problem of great importance in computer vision with far reaching applications. It has also received considerable attention over the years as seen from the number of publications [2, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21] and books [4, 1] devoted to the topic. It is generally accepted that for parallel projection the factorization method of

---

\* This work was supported by the Swedish Foundation for Strategic Research in the VISIT program.

Tomasi-Kanade [18] is numerically the most satisfying. It is optimal under the assumption of isotropic Gaussian noise in the image data. For perspective projection, the projective factorization of Sturm-Triggs [17,19] has been demonstrated to be one of the most numerically efficient methods, see e.g. [5]. This method is similar to affine factorization but requires known epipolar geometry. Both these methods have a major disadvantage, however, in the fact that they require all points to be visible in all views. This of course limits their usefulness in most common multiple view situations where points eventually will be occluded as the camera viewpoint changes. Some suggestions to overcome this problem has been made [9,18], but they require careful analysis and selection of image data in order to be used. Alternative methods for handling missing data for affine [10] and projective views [20] use the so-called closure constraints. The idea is to obtain the camera's motion linearly and simultaneously from a series of bi- or tri-focal tensors.

Ideally, an algorithm for multiple view reconstruction should utilize *all* available image data directly in an efficient manner. Obviously, a minimum overlap of views is necessary for the computation of relative camera positions [14].

In [15] a linear algorithm for multi view reconstruction was presented which requires four coplanar reference points to be visible in all views. All image data, except for image data from points on this reference plane, is used directly to recover points and camera centers simultaneously. The key idea is to map the reference plane to infinity which transforms the projective multi camera situation to the case of purely translating calibrated cameras. In this paper we will demonstrate that there are more multiple view situations which can be transformed to this mathematically simpler structure of translating calibrated cameras. Namely, these are exactly the cases for which affine and projective factorization can be applied:

- general affine cameras
- general projective cameras with known relative epipolar geometry.

No assumption about the scene structure is needed. The main differences of our approach to the bilinear factorization methods are:

- the selection of a finite plane as the plane at infinity
- the allowance of arbitrary missing data, with three points visible in all views
- the computation of the null space of one image data matrix.

The fact that a finite plane will be mapped to infinity is a potential problem for numerical calculations. However, we will demonstrate practically and experimentally that this problem can be handled. Additionally, we present an iterative algorithm for the case of projective cameras and unknown epipolar geometry that initially assumes affine views and compensates iteratively for the perspective effects. A similar idea has been suggested by [4,6,19] to circumvent the pre-estimation of the epipolar geometry for projective factorization. In this paper affine cameras are represented in a projective framework. This is novel and leads to a unified treatment of parallel and perspective projection in a single framework.



## 2 Structure, Motion and the Infinite Homography

General perspective projection of a 3D point  $P_i$  onto the 2D image point  $p_{ij}$  can be described in homogeneous coordinates as:

$$p_{ij} \sim H_j (I \mid -\bar{Q}_j) P_i \sim H_j (\bar{P}_i - \bar{Q}_j), \quad (1)$$

where  $H_j (I \mid -\bar{Q}_j)$  represents the  $3 \times 4$  projection matrix of camera  $j$ . Non-homogeneous coordinates are denoted with a bar, e.g.  $\bar{Q}_j$ , and homogeneous coordinates without a bar, e.g.  $p_{ij}$ . The symbol “ $\sim$ ” means equality up to scale. Let us consider the homography  $H_j$  in more detail. A point  $P = (X, Y, Z, 0)^T$ , which lies on the plane at infinity  $\pi_\infty$ , is mapped by eqn. (1) onto the image plane  $\pi_j$  as:

$$p_{ij} \sim H_j \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (2)$$

Therefore,  $H_j$  can be considered as the *infinite homography*<sup>1</sup> between the plane at infinity  $\pi_\infty$  and the image plane  $\pi_j$ . From eqn. (1) we see that if  $H_j$  is known, we are left with a linear and symmetric relationship between non-homogeneous points and camera centers:

$$p_{ij}^* \sim H_j^{-1} p_{ij} \sim \bar{P}_i - \bar{Q}_j. \quad (3)$$

This suggests the following approach for structure and motion recovery:

1. Determine the infinite homographies  $H_j$
2. Reconstruct points and camera centers.

Section 3 will discuss several ways to determine  $H_j$  for affine views, projective views and scenes containing a reference plane.

If  $H_j$  is known, eqn. (3) can be transformed into three projection relations, where the unknown scale is eliminated by taking ratios:

$$\begin{aligned} -y_{ij}^* \bar{X}_i + x_{ij}^* \bar{Y}_i + y_{ij}^* \bar{A}_j - x_{ij}^* \bar{B}_j &= 0 \\ -w_{ij}^* \bar{X}_i + x_{ij}^* \bar{Z}_i + w_{ij}^* \bar{A}_j - x_{ij}^* \bar{C}_j &= 0 \\ -w_{ij}^* \bar{Y}_i + y_{ij}^* \bar{Z}_i + w_{ij}^* \bar{B}_j - y_{ij}^* \bar{C}_j &= 0 \end{aligned} \quad (4)$$

with  $p_{ij}^* = (x_{ij}^*, y_{ij}^*, w_{ij}^*)^T$ ,  $\bar{P}_i = (\bar{X}_i, \bar{Y}_i, \bar{Z}_i)^T$  and  $\bar{Q}_j = (\bar{A}_j, \bar{B}_j, \bar{C}_j)^T$ . Therefore, each scene point  $\bar{P}_i$  visible in view  $j$  provides three linear relations of the form (4) which can be put into a set of linear equations (SLE). For  $n$  points and  $m$  views the SLE has the form (explicit in [15]):

$$L h = 0 \text{ with } h = (\bar{X}_1, \bar{Y}_1, \bar{Z}_1, \dots, \bar{X}_n, \bar{Y}_n, \bar{Z}_n, \bar{A}_1, \bar{B}_1, \bar{C}_1, \dots, \bar{A}_m, \bar{B}_m, \bar{C}_m)^T. \quad (5)$$

<sup>1</sup> Note, the definition of the infinite homography is slightly different to [4,1].

The Singular Value Decomposition (SVD) of  $L$  shows that  $L$  has a four dimensional null-space. However, three of the four singular vectors of the null-space have the trivial form:  $\bar{P}_i = \bar{Q}_j = (1, 0, 0)^T$ ,  $\bar{P}_i = \bar{Q}_j = (0, 1, 0)^T$  and  $\bar{P}_i = \bar{Q}_j = (0, 0, 1)^T$ . Therefore, the summation of all four singular vectors of the null-space gives the non-trivial solution for all camera centers and points. However, points on the plane at infinity  $\pi_\infty$  increase the dimensionality of the null-space of  $L$  (see [15]). Therefore, the projection relation of those points have to be excluded from the SLE. Since the infinite homography  $H_j$  is known, those points can be reconstructed directly with eqn. (2). How such points are detected automatically and how the SLE is formulated in an optimal way will be discussed in section 4. Let us summarize the main advantages of this approach:

- *One linear system containing all image data*
- *Missing data can be handled*
- *Points and cameras are determined simultaneously.*

### 3 Determine the Infinite Homographies

It was shown in [15] that the infinite homographies  $H_j$  can be determined if the scene contains a reference plane visible in all views. However, in this section we will show that  $H_j$  can be determined for affine or projective cameras and *general scenes* without constraints on the scene structure. Additionally, for projective cameras the epipolar geometry has to be known. Let us begin with the reference plane case.

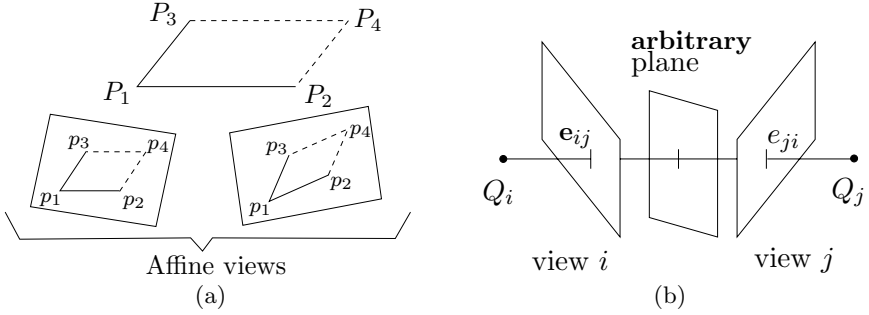
#### 3.1 Reference Plane

In order to determine  $H_j$ , it was assumed in [15] that four coplanar scene points  $P_1, P_2, P_3, P_4$  are visible in all views. The coordinates of these reference points were chosen in a canonical way in the projective space  $P^3$ . Furthermore, the image basis  $p_{ij}$  was mapped to the normalized projective image basis  $p_{ij}^*$  with  $p_{1j}, p_{2j}, p_{3j}, p_{4j}$  as basis points. This corresponds to the following mapping of the reference points onto the image plane  $j$ :

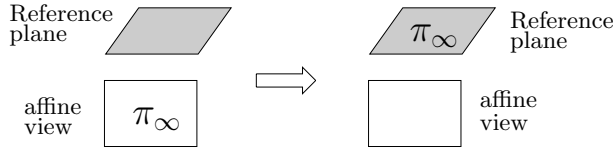
$$\begin{array}{cccc}
 P_1 & P_2 & P_3 & P_4 \\
 - & - & - & - \\
 1 & 0 & 0 & 1 \\
 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0
 \end{array}
 \longrightarrow
 \begin{array}{cccc}
 p_{1j}^* & p_{2j}^* & p_{3j}^* & p_{4j}^* \\
 - & - & - & - \\
 1 & 0 & 0 & 1 \\
 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1
 \end{array}
 . \quad (6)$$

We see that all the reference points lie on the plane at infinity. This means that in the particular chosen projective space, the reference plane is the plane at infinity. This transformation was noted in [21] as a fundamental requirement for obtaining the linear relationship between points and camera centers (see eqn. (3)). The infinite homography of each image  $j$  can now be derived from eqn. (2) as  $H_j = I$ . Alternatively,  $H_j$  could be derived from the inter-image homographies induced by the reference plane.

### 3.2 Affine Cameras



**Fig. 1.** Determine a fourth coplanar point for affine (a) and projective (b) views.



**Fig. 2.** Moving the plane at infinity  $\pi_\infty$  from its ‘true’ location to the reference plane.

Let us assume that three reference points  $P_1, P_2, P_3$  are visible in all views. These three points uniquely define a reference plane. The basic idea is to deduce a fourth ‘virtual’ reference point which lies on the reference plane as well. Let us define the coplanar 3D point  $P_4$  as  $P_4 = P_3 + P_2 - P_1$  (see fig. 1 (a)). Since affine cameras perform a parallel projection on scene points, the affine image of  $P_4$  in view  $j$  is  $p_{4j} = p_{3j} + p_{2j} - p_{1j}$ . Alternatively, the fourth point could be chosen as the centroid of the three reference points.

However, how are affine cameras embedded in the projective framework derived in the previous section? Let us reconsider the mapping of a general projective camera as in eqn. (1):

$$p_{ij} \sim \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ v_1 & v_2 & v_3 & v_4 \end{pmatrix} P_i. \quad (7)$$

The last row of the camera matrix is the *principle plane*  $\pi_{prin} = (v_1, v_2, v_3, v_4)^T$  of the camera which contains the camera center and is parallel to the image plane. In a projective space where the plane at infinity is at its true location, the principle plane of an affine camera is the plane at infinity, i.e.  $\pi_{prin} = \pi_\infty = (0, 0, 0, 1)^T$  (fig. 2 left). However, we have seen in the previous section that in order to determine  $H_j$ , the reference plane has to be the plane at infinity in the particular chosen projective space (fig. 2 right). This means that in this particular projective space all camera centers lie on a plane  $\pi_{prin}$  which is different to  $\pi_\infty$ . Eqn.

(7) can now be transformed into eqn. (1) with non-homogeneous coordinates for the camera centers and scene points. From the four coplanar reference points, the infinite homographies  $H_j$  can be derived with eqn. (2) and (6). The reconstructed cameras provide the principle plane  $\pi_{prin}$ , which contains all camera centers. Finally, by mapping  $\pi_{prin}$  to  $\pi_\infty$  the projective reconstruction transforms into an affine reconstruction.

How does this approach compare to other affine reconstruction methods? In our approach 6 parameters of each affine camera are determined directly by the infinite homographies. The remaining 2 unknown parameters, which represent the direction of an affine camera, are reconstructed simultaneously with the scene points. In contrast to this, affine factorization [18] determines 2 parameters of each affine camera in forehand. The remaining 6 parameters of each camera are determined simultaneously with the scene points. However, this method does not allow missing data. It has been shown [6,12] that all 8 unknown parameters of an affine camera could be determined directly by choosing a special affine basis in the scene and in the image. However, from an numerical point of view this is less favourable.

### 3.3 Projective Cameras

Let us assume that the three reference points  $P_1, P_2, P_3$  have canonical coordinates in the projective space and in the image as in eqn. (6). The infinite homography for each view  $j$  is then described as:

$$H_j = \begin{pmatrix} a_j & 0 & 0 \\ 0 & b_j & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (8)$$

The arbitrary scale of the matrix is fixed by setting  $H_j(3,3) = 1$ . The variables  $a_j$  and  $b_j$  are unknown in each view  $j$  and can be considered as the mapping of point  $(1, 1, 1, 0)^T$  into view  $j$ :  $H_j(1, 1, 1)^T = (a_j, b_j, 1)^T$ . Let us assume that the epipolar geometry is known, i.e we have the fundamental matrices between each pair of views which have at least seven points in common. We denote the epipole  $e_{ij} = (e_{ijx}, e_{ijy}, e_{ijw})^T$  as the projection of camera center  $j$  into view  $i$  (see fig. 1(b)). The inter-image homography from view  $i$  to view  $j$  via a certain plane is defined as  $H_{ij} = H_i^{-1} H_j$  where  $H_i, H_j$  denote the respective homographies as defined in eqn. (1). Since the epipols between two views are in correspondence via any plane in the scene (see fig. 1(b)), we may write:

$$e_{ji} \sim H_{ij} e_{ij} \Leftrightarrow H_i e_{ji} \sim H_j e_{ij}. \quad (9)$$

Taking equation (8) and (9) we obtain two constraints between views  $i$  and  $j$ :

$$\begin{aligned} a_i e_{jix} e_{ijw} - a_j e_{ijx} e_{jiw} &= 0 \\ b_i e_{jiy} e_{ijw} - b_j e_{ijy} e_{jiw} &= 0. \end{aligned} \quad (10)$$

All the  $a_i$ 's and  $b_i$ 's may now be determined separately but simultaneously. Each pair of images  $i$  and  $j$ , which are linked by a fundamental matrix, gives an linear

equation in  $a_i, a_j$  and  $b_i, b_j$  respectively. With  $m$  images we obtain two sets of linear equations:

$$\begin{aligned} L_a h_a &= 0 \text{ with } h_a = (a_1, \dots, a_m)^T \text{ and} \\ L_b h_b &= 0 \text{ with } h_b = (b_1, \dots, b_m)^T. \end{aligned} \quad (11)$$

The last singular vector of the SVD of  $L_a$  and  $L_b$  gives the solution for  $h_a$  and  $h_b$  respectively. The vector  $h_a$  and  $h_b$  have an arbitrary scale which corresponds to the fact that the fourth unknown reference point on the reference plane has two degrees of freedom.

The advantage of deriving the infinite homographies in this way is that all homographies are determined in one step which implies that the complete information given by the geometry is used simultaneously.

### 3.4 Known Structure and Cameras

For an iterative structure and motion algorithm, we would like to update the infinite homography  $H_j$  on the basis of known 3D scene points and cameras. This means that  $P_i$  and  $\bar{Q}_j$  are known and we obtain:

$$p_{ij} \sim H_j (I \mid -\bar{Q}_j) P_i \sim H_j p'_{ij} \quad (12)$$

where  $p'_{ij}$  is the projection of point  $P_i$  by camera  $(I \mid -\bar{Q}_j)$ . Since  $p_{ij}$  and  $p'_{ij}$  are known, the infinite homography  $H_j$  can be determined for each image  $j$  individually with e. g. the normalized 8-point algorithm (see [3]).

### 3.5 Choice of Reference Points

In practice more than three points might be visible in all views of a multiple view situation. Naturally the question arises of how to find the optimal three reference points. Let us consider the criteria for good reference points. Firstly, a camera center must not lie on the reference plane. This means that the three reference points must not be collinear in any view. Secondly, in the presence of noise the infinite homography is determined more accurately if the projected reference points are far apart in the image. Since the two criteria are not contradictive, we choose as reference points those three points which are “least collinear”. This is done by considering the distance between one reference point to the line defined by the other two reference points.

## 4 Structure and Motion with Infinite Homographies

We have seen that with the knowledge of  $H_j$  the relationship between known image points  $p^*_{ij}$  and unknown points  $\bar{P}_i$  and camera centers  $\bar{Q}_j$  is linear (see eqn. (3)). Furthermore, eqn. (3) shows that changing the image basis by a homography  $B$  and individually scaling the image points  $p^*_{ij}$  by  $s_{ij}$  does not alter this relationship:

$$p'_{ij} \sim s_{ij} B p^*_{ij} \sim B \bar{P}_i - B \bar{Q}_j \sim \bar{P}_i - \bar{Q}_j'. \quad (13)$$

How to choose  $B$  and  $s_{ij}$  in an optimal way will be discussed in this section.

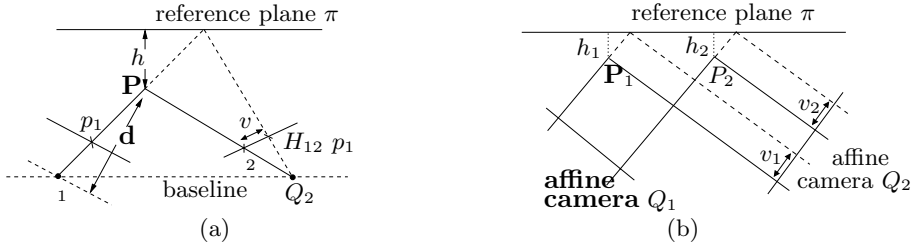
If  $B$  and  $s_{ij}$  are known,  $p'_{ij}$  can be derived and we obtain a set of linear equations (SLE) as in section (2):

$L h = 0$  with

$$h = (\bar{X}'_1, \bar{Y}'_1, \bar{Z}'_1, \dots, \bar{X}'_n, \bar{Y}'_n, \bar{Z}'_n, \bar{A}'_1, \bar{B}'_1, \bar{C}'_1, \dots, \bar{A}'_m, \bar{B}'_m, \bar{C}'_m)^T. \quad (14)$$

Since points which are on or “close” to the reference plane potentially increase the numerical stability of the reconstruction, the projection relations of such points have to be excluded from the SLE. However, how can these points be detected automatically? One idea is to exclude successively points from the SLE which are close to the reference plane. Therefore, a ranking of all points on the basis of their *distance to the reference plane* has to be known.

#### 4.1 Distance between Points and Reference Plane



**Fig. 3.** Parallax geometry for (a) projective and (b) affine cameras.

Let us consider a configuration with two cameras  $Q_1, Q_2$ , a 3D point  $P$  and a reference plane  $\pi$  where  $P$  does not lie on  $\pi$  (fig. 3(a) depicts a top view). The inter-image homography from the first to the second view via the reference is as defined in the previous section:  $H_{12} = H_1^{-1} H_2$ . The residual parallax vector in the second view is given as  $v = p_2 - H_{12} p_1$ . Obviously,  $v$  is null if  $P$  lies on  $\pi$ . However,  $v$  vanishes as well if  $P$  lies on the baseline of the two views. Therefore, the distance of a point to the reference plane can not be determined directly from its parallax vector. Let us define  $\gamma_i = \frac{h_i}{d_i}$ , where  $h_i$  is the perpendicular distance of  $P_i$  to the reference plane and  $d_i$  is the depth of  $P_i$  with respect to the first view (see fig. 3(a)). It is known [8] that the relative depth  $\frac{\gamma_1}{\gamma_2}$  of two points  $P_1$  and  $P_2$  can be derived directly from their parallax vectors  $v_1, v_2$ . This means that the relative distance  $\frac{h_1}{h_2}$  of two points depends on both their parallax vectors and their depths. However, if we assume parallel projection,  $d_i$  is constant and we obtain the relative distance of two points as:

$$\frac{\gamma_1}{\gamma_2} = \frac{h_1}{h_2} = \frac{v_1}{v_2}. \quad (15)$$

Fig. 3(b) depicts a configuration with affine cameras where  $h_1 = h_2$  and therefore  $v_1 = v_2$ . We will use eqn. (15) as an approximation for projective cameras.

The original task was to determine a unique function  $dis(P_i)$  which represents the distance between a point  $P_i$  and the reference plane. Eqn. (15) supplies a distance function  $dis_{j_1 j_2}(\cdot)$  between each pair of views  $j_1, j_2$ , which is unique up to scale. A unique function  $dis(\cdot)$  can be obtained by recursively merging the set of functions  $dis_{j_1 j_2}(\cdot)$ . Finally,  $dis(\cdot)$  is scaled so that the maximal distance of a point to the reference plane is equal to one, i.e.  $dis(\cdot) \in [0, 1]$ .

## 4.2 The Choice of the Image Basis

It has been shown in [3] that the normalization of image coordinates can dramatically influence the result of a computation based on image coordinates. Normalization means that the centroid of all image coordinates is at the origin and the average distance of an image point to the origin is equal to  $\sqrt{2}$ . If we consider eqn. (13), normalization would involve to determine for each view  $j$  an individual matrix  $B_j$ , which represents the normalization. However, such a  $B_j$  would destroy the linear relationship between points and camera centers. Therefore, the matrix  $B$  has to be determined independently of a certain view  $j$ . We define:

$$B = \frac{1}{m} \sum_{j=1}^m B_j / \|B_j\|_2, \quad (16)$$

where  $\|\cdot\|_2$  is the Frobenius norm of a matrix and  $m$  is the number of views.

## 4.3 Weighting the Set of Equations

Let us consider a point  $P_1$  which is closer to the reference plane than another point  $P_2$ . Since the reference plane is the plane at infinity in the chosen projective space, the coordinates of the reconstructed point  $\bar{P}_1$  are larger than the ones of  $\bar{P}_2$ . This means that in the presence of noise, the point with larger coordinates is reconstructed more accurately. In order to eliminate this favoring of certain points we suggest to choose<sup>2</sup> the scale factors in eqn. (13) as  $s_{ij} = dis(P_i)$  where  $dis(\cdot) \in [0, 1]$ . This means that points which are closer to the reference plane are inhibited. The same applies to the equations in the SLE of such a point.

## 5 Outline of the Algorithms

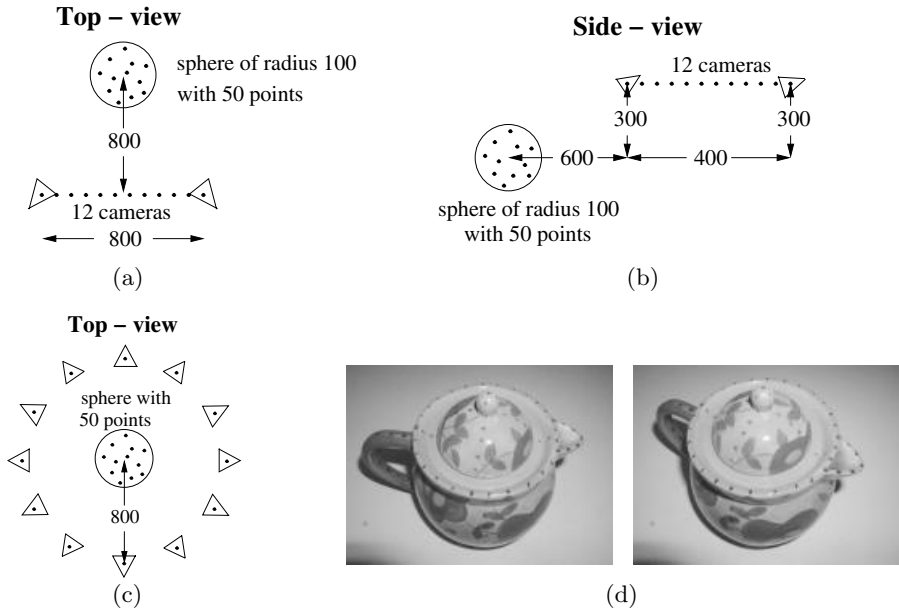
The different ideas of the previous sections result in three algorithms: **A-*alg.*** (for affine cameras), **P-*alg.*** (for projective cameras) and **AtoP-*alg.*** (an iterative version for projective cameras). The **AtoP-*alg.*** assumes initially affine cameras and compensates iteratively for the perspective effects. However, there is no guarantee that the algorithm will converge to a global minimum. The algorithms are composed of the following steps:

<sup>2</sup> This particular choice of the scale factors  $s_{ij}$  is motivated by the mapping  $(0, 1)^T \rightarrow (1, 0)^T$  and  $(1, 1)^T \rightarrow (1, 1)^T$  in the projective space  $P^1$ .

1. Detect optimal three reference points (sec. 3.5)
2. Determine  $H_j$  (sec. 3)
- A-alg, AtoP-alg:** assume affine cameras; **P-alg:** assume projective cameras
3. Determine distances between points and reference plane (sec. 4.1)
4. Exclude iteratively points from the SLE
5. Determine matrix  $B$  (sec. 4.2)
7. Determine scales  $s_{ij}$  and image points  $p'_{ij} = s_{ij} \|BH_j^{-1}p_{ij}\|_2$  (sec. 4.3)
8. Obtain  $\bar{P}'_i, \bar{Q}'_j$  by SVD (eqn. (14)) and  $P_i$  on the ref. plane (eqn. (2))
9. Only **AtoP-alg.:** Update  $H_j$  (sec. 3.4) and go to 3.
- Stop if the RMS-error is either unchanged or increases
10. Take the best result on the basis of RMS-error
11. Undo the basis change:  $\bar{P}_i = B^{-1} \bar{P}'_i$  and  $\bar{Q}_j = B^{-1} \bar{Q}'_j$

The quality of the reconstruction is evaluated in terms of the Root-Means-Square (RMS) error. However, other criteria could be used.

## 6 Experiments



**Fig. 4.** The four different configurations: lateral movement – LAT (a), translational movement towards the scene – TOW (b), circular movement – CIR (c) and 2 images of a real sequence – TEA (d).

In order to demonstrate the performance of the algorithms, they were applied to a wide range of different camera motions and scene settings. Fig. 4 depicts four of them: three synthetic configurations (a-c) and a real sequence (d). Each synthetic configuration consists of 50 points distributed uniformly in a sphere



and 12 cameras pointing towards the center of the sphere. They differ in the type of camera motion: lateral – LAT (a), circular – CIR (c) and translational towards the scene – TOW (b). The distances (in units) between cameras and points are as in fig. 4. Additionally, a real sequence of 8 images (see fig. 4(d)) was utilized. The camera circled around the teapot, similar to the CIR-configuration. In order to conduct synthetic experiments on this sequence, a reconstruction of the cameras and 96 points of the teapot (see fig. 8) served as the basis for a synthetic configuration (TEA). If not stated explicitly, all the points of the TEA-configuration are visible in all views, i.e. no missing data.

## 6.1 Synthetic Data

The synthetic experiments were carried out with respect to different levels of Gaussian noise:  $\sigma = 0, 0.2, \dots, 3.0$  (standard deviation). In order to obtain average performance, the following two steps were conducted 20 times for each noise level: a) randomly determine 50 scene points b) add Gaussian noise on the reprojected 3D points. In case of projective cameras, the internal calibration matrix was chosen as  $\text{diag}(500, 500, 1)$ . Affine cameras were derived from the projective cameras by moving the center of projection to infinity where the image size remained fixed (see [4]).

The computed reconstructions were evaluated in terms of the Root-Mean-Square (RMS) error between reprojected 3D points and 2D image data (potentially corrupted by noise). The performance of the three algorithms presented in this paper: **A-alg.**, **P-alg.** and **AtoP-alg.** is compared to affine factorization of Tomasi-Kanade (**TK-alg.**) [18] and projective factorization of Sturm-Triggs (**ST-alg.**) [17,19]. In [17] it is suggested to derive the initial “projective depths” from epipolar geometry. Other authors, e.g. [6,4,5], have shown that initialising all the “projective depths” to one and reestimating them by reprojection produce good results as well. This more simple approach was used in this paper.

**Different Configurations.** Let us consider the performance of the A-alg. and TK-alg. for different configurations (fig. 5 (a,b)). In this case, the scene is viewed by affine cameras. The performance of the TK-alg. is equally good for all configurations and close to identical with the theoretical minimum, i.e. Cramer-Rao lower bound (not shown). The differences between the results of the TK-alg. and the A-alg. are not large but noticeable. Furthermore, the A-alg. performed worse for the TOW-configuration than for the other three configurations. Since the TOW-configuration has the shortest baseline relative to the scene (see fig. 4 (b)) this result can be expected.

Fig. 5 (c,d) shows the results of the P-alg. and ST-alg. with respect to different configurations. In contrast to the previous section projective cameras were used. As in the case of affine factorization, the ST-alg. is equally good for all configurations and close to the theoretical minimum. The difference between the results of the ST-alg. and the P-alg. are obvious. However, for practical noise levels, e.g.  $\sigma = 1.0$ , the results of the P-alg. are still acceptable. A comparison between the A-alg. (a,b) and P-alg. (c,d) shows that the results of the A-alg.

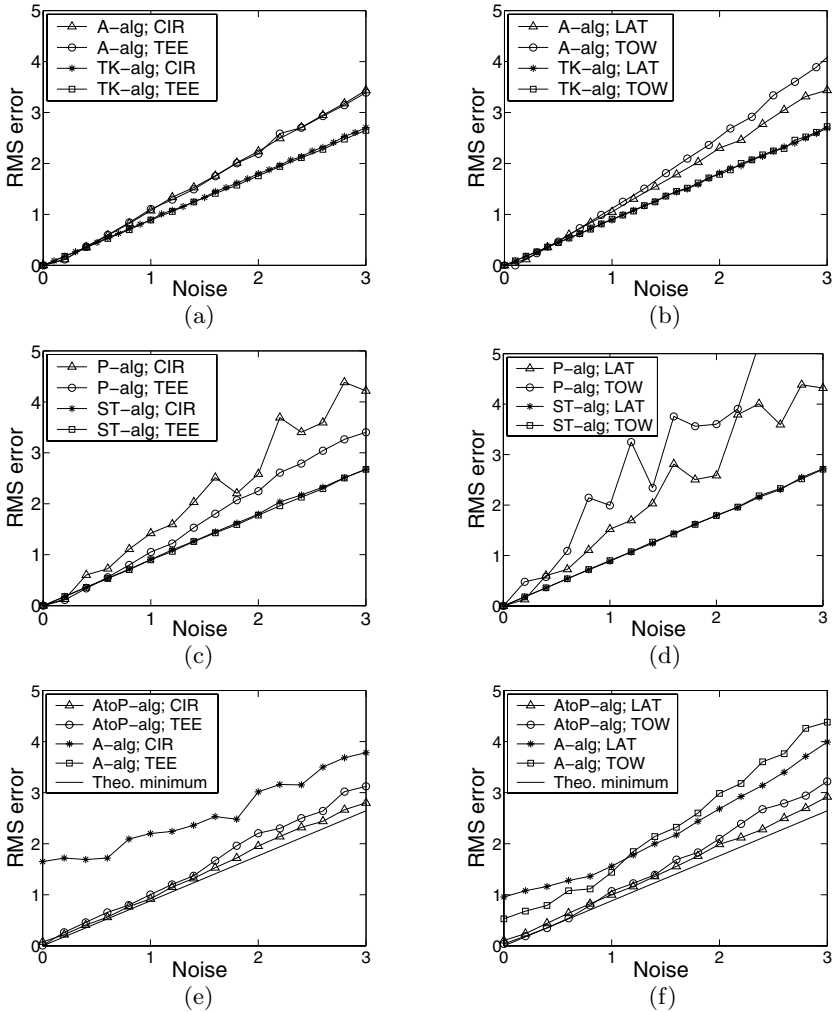
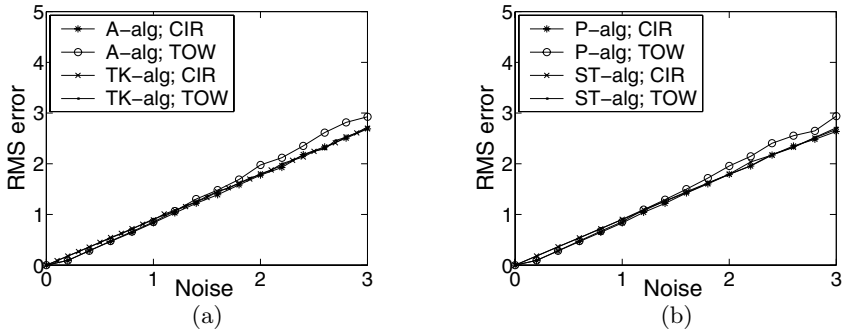


Fig. 5. Graphs in respect to different configurations.

are better and the curves for the A-alg. are more linear. The only difference between the A-alg. and P-alg. is the derivation of the infinite homographies. A more detailed analyses confirmed that this derivation in the case of projective cameras is fairly sensitive to noise in the epipols and reference points.

Finally, fig. 5 (e,f) depicts the performance of the AtoP-alg. for different configurations with projective cameras. Additionally, the results of the A-alg. are shown, which serve as the initialisation for the iterative AtoP-alg. The theoretical minimum is displayed as well. The results of the A-alg. on the TEA-configuration were off the scale (RMS-error between 18.4 and 19.1). It stands out, that for all configurations the initial reconstruction of the A-alg. can be significantly improved by the AtoP-alg. Particularly, in the case of no noise, i.e.  $\sigma = 0$ , the AtoP-alg. converged for all configuration close to the theoretical minimum.

However, for higher noise levels, e.g.  $\sigma = 3.0$ , the AtoP-alg. did not always converge close to the theoretical minimum, e.g. TOW-configuration.



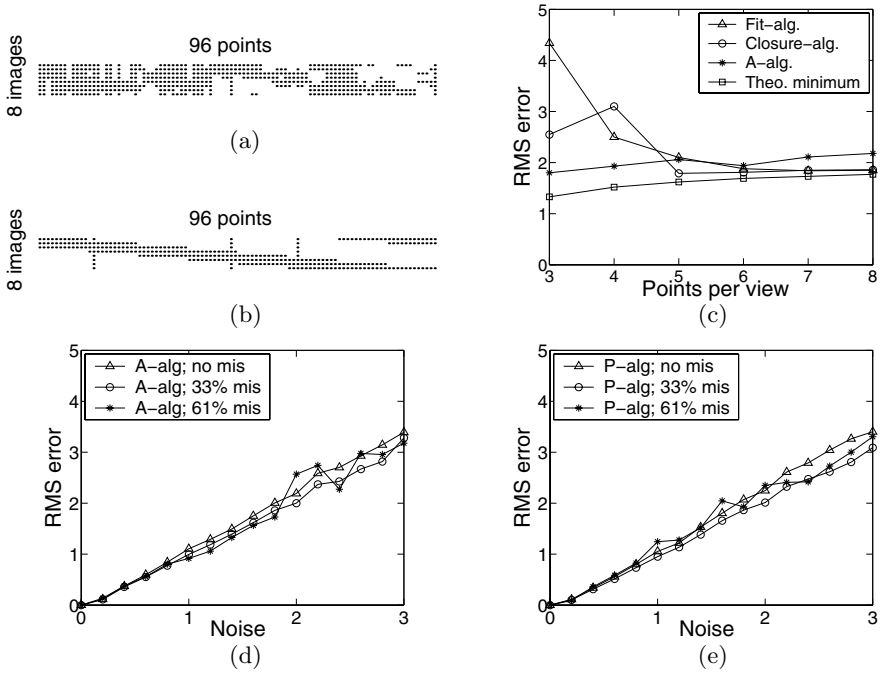
**Fig. 6.** Graphs for the case of perfect reference points.

**Choice of Reference Points.** In this section we will repeat some of the experiments of the previous section. However, Gaussian noise will be added to all image points *except* for the three reference points. Fig. 6 depicts the results for the case of affine (a) and projective cameras (b). In contrast to the previous section, the performance of the A-alg. and TK-alg. (fig. 6 (a)) is close to identical. The same applies to the performance of the P-alg. and ST-alg. (fig. 6 (b)). Further experiments on the AtoP-alg. and other configurations confirmed this performance. This leads to the conclusion that, independent of the configuration, the *choice of reference points is crucial* for the three presented algorithms.

**Missing Data.** In the previous experiments was assumed that all 96 points of the TEA-configuration are visible in all 8 images, i.e. no missing data. However, in practice some points might be occluded in several views. Fig. 7 (a) shows the “true” *visibility matrix* of the TEA-configuration. An element of this matrix is set (black dot) if the respective point is visible in the respective view. It turns out, that 33% of the entries are not set. If the correspondence between successive frames were obtained by tracking, the final visibility matrix might look like in fig. 7 (b). Each point, except for the three reference points, is only visible in three successive views. The amount of missing data increases to 61%.

If all points are visible in all views, the optimal reference points correspond to points on the body of the teapot (fig. 4 (d)). In the case of missing data (33% or 61%), points on the rim and handle were detected as the best reference points.

Let us consider the performance of the A-alg. (fig. 7(d)) and the P-alg. (fig. 7 (e)) on these three types of visibility matrices: no missing data, 33% missing data (fig. 7(a)) and 61% missing data (fig. 7(b)). The first observation is that the performance of the A-alg. and P-alg. differs only slightly in respect to the different cases of missing data. Further experiments confirmed the conclusion that the performance of the three novel algorithms is “fairly” independent to the amount of missing data. A more detailed analyses shows that both algorithms performed less stable for the case of 61% missing data (zigzag shape of the



**Fig. 7.** Visibility matrices (a,b) and graphs (c-e) for the case of missing data.

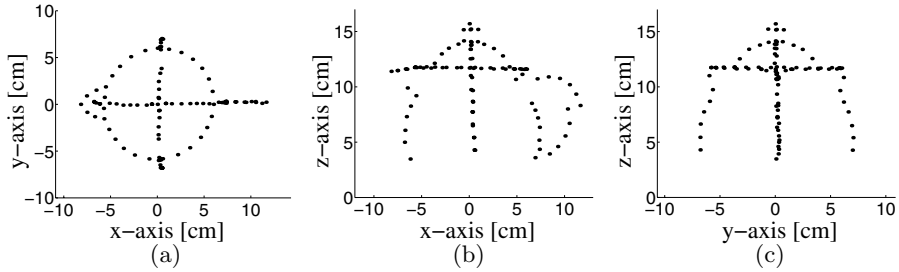
curves). This performance can be expected since “only” three successive views (fig. 7(b)), i.e. short baseline, provide information about the 3D position of a certain point in the scene.

In the last experiment we compare the A-alg. to two alternative methods which handle missing data for affine views. Jacobs algorithm [9] fits a matrix of rank 3 to the data matrix with missing elements (**Fit-alg.**)<sup>3</sup>. Kahl and Heyden [10] use the centred affine tensors between successive two and three views to obtain all camera matrices simultaneously by using the so-called closure constraints (**Closure-alg.**). The main advantage of these methods is that the image data is used in a uniform manner, i.e. no selection of specific reference points. In contrast to our approach, the image data is not used directly to obtain structure and motion simultaneously. Fig. 7(c) shows the result for a noise level of  $\sigma = 2.0$ . The visibility matrix in fig. 7(b) was used, where the number of points per view varied. If a point is visible in more than 5 views the alternative methods performed slightly better, which might be due to noise in the reference points. However, in the case of more missing data, i.e. only 3 or 4 points per view, the alternative methods performed worse in this experiment. In case of the Closure-alg. an explanation might be that the data is not sufficient to obtain good tensors.

## 6.2 Real Data

The 8 images of the teapot (see fig. 4 (d)) served as a real image sequence. On the basis of this, 96 corresponding image points were selected manually which

<sup>3</sup> We used the code available at <http://www.neci.nj.nec.com/homepages/dwj/>.



**Fig. 8.** Top (a), side (b) and front (c) view of the reconstructed teapot (see fig. 4 (d)).

results in the “true” visibility matrix as in fig. 7 (a). The reconstruction obtained with the P-alg. had an RMS error of 2.84 between reprojected 3D points and select image points. Fig. 8 shows the top (a) side (b) and front (c) view of the reconstruction which was metric rectified. Only those scene points which lie on the contour in the top, side or front view of the teapot were reconstructed. The AtoP-alg. performed with an RMS error of 2.06 where the initial reconstruction determined with the A-alg. had a RMS error of 15.82.

## 7 Summary and Conclusion

We have presented two linear methods for the simultaneous computation of 3D points and camera positions from multiple affine and projective views. This is achieved by computing the nullspace of one linear system constructed from *all image data*. In case of affine views the only requirement is that *three points are visible in all views*. Additionally, for projective views the epipolar geometry has to be known. In case of unknown epipolar geometry, a third iterative algorithm for projective views has been presented. The treatment of affine and projective cameras in a single, unified projective framework is a further, novel contribution.

The only other methods that use all image data directly are the factorization algorithms for affine [18] and projective [17,19] views. However, in contrast to our approach, they require all points to be visible in all views. Since points become inevitably occluded in practice, we consider the presented methods as a major and novel contribution to the problem of structure from motion. Alternative reconstruction methods for handle missing data have been presented [9,10,20], which have the advantage that data is used in a uniform manner, i.e no selection of reference points. However, in contrast to our approach the image data is not used directly to obtain structure and motion simultaneously.

The experiments, which covered a wide range of different camera motions and scene settings, have shown that the presented algorithms perform very well for practical noise levels. If the reference points are chosen carefully the performance of the presented algorithms compared to affine and projective factorization methods is close to identical. Furthermore, the use of all available image data, which is not available for factorization methods, is a most important numerical stabilising factor in this approach.

## References

1. Faugeras, O. and Luong, Q.-T. 2001. *The Geometry of Multiple Images*. The MIT Press.
2. Fitzgibbon, A. W. and Zisserman, A. 1998. Automatic camera recovery for closed or open image sequences. In *Europ. Conf. Comp. Vis.*, Freiburg, Germany, pp. 311-326.
3. Hartley, R. 1997. In defence of the 8-point algorithm. In *IEEE Trans. on Pattern Anal. and Machine Intell.*, 19(6), pp. 580-593.
4. Hartley, R. and Zisserman, A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.
5. Hartley, R., Dano, N. and Kaucic, R. 2001. Plane-based Projective Reconstruction. In *Int. Conf. Comp. Vis.*, Vancouver, Canada, pp. 420-427.
6. Heyden, A., Berthilsson, R. and Sparr, G. 1999. An iterative factorization method for projective structure and motion from image sequences. In *Image and Vision Computing*, 17(13), pp. 981-991.
7. Heyden, A. and Kahl, F. 2000. Direct Affine Reconstruction. *Int. Conf. Pattern Recog.*, Barcelona, Spain, pp. 885-888.
8. Irani, M. and Anandan, P. 1996. Parallax geometry of pairs of points for 3d scene analysis. In *Europ. Conf. Comp. Vis.*, Cambridge, UK, pp. 17-30.
9. Jacobs, D. 1997. Linear Fitting with Missing Data for Structure-from-Motion. In *IEEE Conf. Comp. Vis. and Pattern Recog.*, San Juan, Puerto Rico, pp. 206-212.
10. Kahl, F. and Heyden, A. 1999. Affine structure and motion from points, lines and conics. In *Int. J. Computer Vision*, 33(3):163-180.
11. Koch, R., Pollefeys, M. and VanGool, L. 1998. Multi viewpoint stereo from uncalibrated video sequences. In *Europ. Conf. Comp. Vis.*, Freiburg, Germany, pp. 55-65.
12. Koenderink, J.J. and van Doorn, A.J. 1991. Affine structure from motion. In *J. Opt. Soc. Am. A*, 8(2), pp. 377-385.
13. Oliensis, J. 1995. Multiframe Structure from Motion in Perspective. In *Workshop on Representations of Visual Scenes*, Boston, USA, pp. 77-84.
14. Quan, L., Heyden, A. and Kahl, F. 1999. Minimal Projective Reconstruction with Missing Data. In *IEEE Conf. Comp. Vis. and Pattern Recog.*, Fort Collins, Colorado, pp. 210-216.
15. Rother, C. and Carlsson S. 2001. Linear Multi View Reconstruction and Camera Recovery. In *Int. Conf. Comp. Vis.*, Vancouver, Canada, pp. 42-51.
16. Schaffalitzky, F., Zisserman, A., Hartley, R. I. and Torr, P.H.S. 2000. A Six Point Solution for Structure and Motion. In *Europ. Conf. Comp. Vis.*, Dublin, Ireland, pp. 632-648.
17. Sturm, P. and Triggs, B. 1996. A factorization based algorithm for multi-image projective structure and motion. In *Europ. Conf. Comp. Vis.*, Cambridge, U.K., pp. 709-719.
18. Tomasi, C. and Kanade, T. 1992. Shape and Motion from Image Streams under Orthography: a Factorization Method. In *Int. J. Computer Vision*, 9(2):137-54.
19. Triggs, B. 1996. Factorization methods for projective structure and motion. In *IEEE Conf. Comp. Vis. and Pattern Recog.*, San Francisco, CA, pp. 845-851.
20. Triggs, B. 1997. Linear projective reconstruction from matching tensors. In *Image and Vision Computing*, 15(8), pp. 617-625.
21. Triggs, B. 2000. Plane + Parallax, Tensors and Factorization. In *Europ. Conf. Comp. Vis.*, Dublin, Ireland, pp. 522-538.

# Model-Based Silhouette Extraction for Accurate People Tracking

Ralf Plaenkers and Pascal Fua\*

Computer Graphics Lab (LIG)

Computer Graphics Lab, EPFL, CH-1015 Lausanne, Switzerland

`pascal.fua@epfl.ch`

**Abstract.** In this work, we introduce a model-based approach to extracting the silhouette of people in motion from stereo video sequences. To this end, we extend a purely stereo-based approach to tracking people proposed in earlier work. This approach is based on an implicit surface model of the body. It lets us accurately predict the silhouette’s location and, therefore, detect them more robustly. In turn these silhouettes allow us to fit the model more precisely. This allows effective motion recovery, even when people are filmed against a cluttered unknown background. This is in contrast to many recent approaches that require silhouette contours to be readily obtainable using relatively simple methods, such as background subtraction, that typically require either engineering the scene or making strong assumptions.

We demonstrate our approach’s effectiveness using complex and fully three-dimensional motion sequences where the ability to combine stereo and silhouette information is key to obtaining good results.

## 1 Introduction

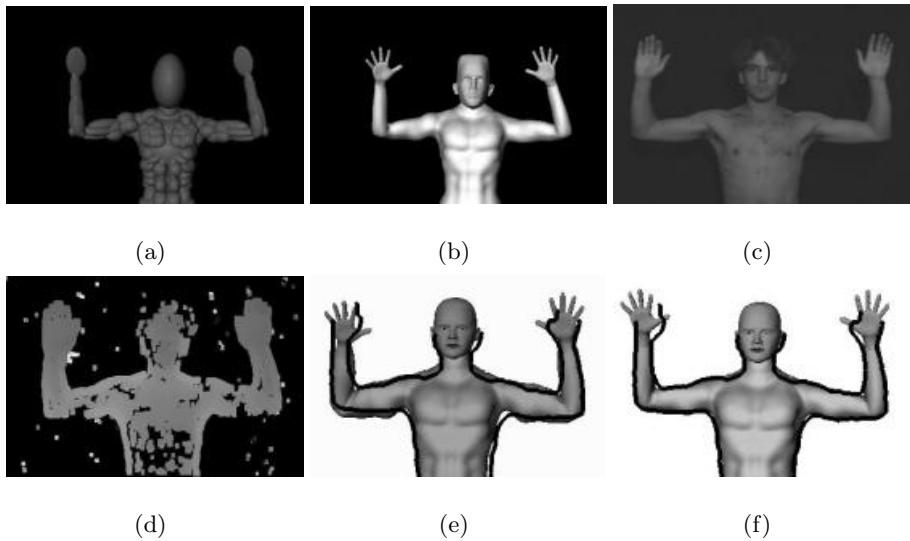
In recent years, much work has been devoted to tracking people from video sequences. Many of the techniques that have been proposed rely on extracting silhouettes and fitting body models to them. See [1,11,13] for recent reviews. While this may be practical in some cases—for example, because the background is both static and known, thus allowing background subtraction—silhouette extraction is in general a difficult task.

Here, we present a model-based approach to silhouette extraction that allows us to overcome this problem and to simultaneously recover 3-D body shape and motion as well as 2-D outlines. We use the “articulated soft objects” we proposed in earlier work [14] to represent and track human bodies: We use a conventional articulated skeleton but replace the simple geometric primitives—typically, cylinders or ellipsoids—that are usually attached to it by implicit volumetric primitives. Each one defines a field function and the skin is taken to be a level set of the sum of these fields. This implicit surface formulation has three key strengths:

- **Effective use of stereo and silhouette data:** Defining surfaces implicitly allows us to define a distance function of data points to models that is both differentiable and computable without search.

---

\* This work was supported in part by the Swiss Federal Office for Education and Science



**Fig. 1.** Models and silhouettes. (a) Metaballs attached to an articulated skeleton. (b) Skin surface computed by ray casting. (c) One image of a stereo pair used to estimate the parameters of the model in (b). (d) Corresponding disparity map. (e) The real body outlines overlaid on the skin surface. In this case the model was fitted using stereo only. As a result, it ends up too far from the actual data points and the system compensates by incorrectly enlarging the primitives. (f) Using the silhouettes during the fitting process provides stricter constraints that yield a better result.

- **Accurate shape description by a small number of parameters:** Varying a few parameters yields models that can match different body shapes and allow both shape and motion recovery.
- **Explicit modeling of 3-D geometry:** Geometry can be taken into account to predict the expected location of image features and occluded areas, thereby making the extraction algorithm more robust.

Our model is depicted by Figure 1(a,b). In an earlier publication [14], we used it to demonstrate robust tracking using stereo only. Here, we show that its quality and 3-D nature allow us to effectively predict silhouette locations from frame to frame, constrain the search and, thus, reliably extract them from a cluttered background without having to make *a priori* assumptions about it. These silhouettes are used in turn to constrain model reconstruction, thereby further increasing its precision and reliability.

This is in contrast to many recent approaches that rely either on noisy edge or region information, such as image gradient and optical flow, e.g. [3,9], or on silhouette contours that are assumed to be readily available or obtainable through statistical background subtraction techniques e.g. [12,6,8,7], both of which require either an engineering of the scene or a static background. Furthermore, as noted in [9], the simple rigid volumetric primitive-based models that are often used tend to be a source of inaccuracy. Our approach addresses both of these issues: Our improved models yield a better accuracy that, in turn, gives us the predictive power required for effective silhouette extraction.



Combining stereo and silhouette data is valuable because they are complementary sources of information. The former works best when a body part faces the cameras but becomes unreliable when the surface slants away, which is precisely where silhouettes can be used. Figure 1(e,f) illustrates this complementarity. In this example, we used a single stereo pair. In Figure 1(e) only stereo-data, in the form of a cloud of 3-D points derived from the disparity map, was used. The stereo data is too noisy and shallow to sufficiently constrain the model. As a result, the fitting algorithm tends to move it too far away from the 3-D data and to compensate by inflating the arms to keep contact with the point cloud. Using the silhouettes in addition to the stereo data, however, sufficiently constrains the fitting problem to obtain the much improved result of Figure 1(f).

In the remainder of this paper, we first introduce both our human body model and the optimization framework we use to instantiate its degrees of freedom given stereo and silhouette data. We then turn to our model-based automated silhouette extraction approach and show that it yields precise outlines and that combining silhouettes and stereo yields much better results than using stereo alone.

## 2 Articulated Model and Surfaces

The human body model we use in this work [17] incorporates a highly effective multi-layered approach for constructing and animating realistic human bodies. The first layer is a skeleton that is a connected set of segments, corresponding to limbs and joints. A joint is the intersection of two segments, which means it is a skeleton point around which the limb linked to that point may move.

Smooth implicit surfaces, also known as *metaballs* or *soft objects*, form the second layer [2]. They are used to simulate the gross behavior of bone, muscle, and fat tissue. The metaballs are attached to the skeleton and arranged in an anatomically-based approximation. The head, hands and feet are explicit surfaces that are attached to the body. For display purposes a third layer, a polygonal skin surface, is constructed by ray casting [17].

The body shape and position are controlled by a *state vector*  $\Theta$ , which is a set of parameters controlling joint locations and limb sizes. In this section, we first describe this state vector in more detail and, then, our implicit surface formulation.

### 2.1 State Vector

Our goal is to use video-sequences to estimate our model's shape and derive its position in each frame. Let us therefore assume that we are given  $N$  consecutive video frames and introduce position parameters for each frame.

Let  $B$  be the number of body parts in our model. We assign to each body part a variable length and width coefficient. These dimensions change from person to person but we take them to be constant within a particular sequence. This constraint could be relaxed, for example to model muscular contraction.

The model's *shape* and *position* are then described by the combined state vector

$$\Theta = \{\Theta^w, \Theta^l, \Theta^r, \Theta^g\} , \quad (1)$$

where  $\Theta$  is broken into sub-vectors that control the following model components:

- Shape
  - $\Theta^w = \{\theta_b^w \mid b = 1..B\}$ , the width of body parts.
  - $\Theta^l = \{\theta_b^l \mid b = 1..B\}$ , the length of body parts.
- Motion
  - $\Theta^r = \{\theta_{j,f}^r \mid j = 1..J, f = 1..N\}$ , the rotational degree of freedom of joint  $j$  of the articulated skeleton for all frames  $f$
  - $\Theta^g = \{\theta_f^g \mid f = 1..N\}$ , the six parameters of global position and orientation of the model in the world frame for all frames  $f$

The size and position of the metaballs is relative to the segment they are attached to. A length parameter not only specifies the length of a skeleton segment but also the shape of the attached metaballs in the direction of the segment. Width parameters only influence the metaballs' shape in the other directions.

## 2.2 Metaballs

Metaballs [2] are generalized algebraic surfaces that are defined by a summation over  $n$  3-dimensional Gaussian density distributions, each called a *primitive*. The final surface  $\mathcal{S}$  is found where the density function  $F$  equals a threshold  $T$ , taken to be 0.5 in this work:

$$\mathcal{S} = \left\{ [x, y, z]^T \in \mathbb{R}^3 \mid F(x, y, z) = T \right\} , \quad (2)$$

$$F(x, y, z) = \sum_{i=1}^n f_i(x, y, z) , \quad (3)$$

$$f_i(x, y, z) = \exp(-2d_i(x, y, z)) , \quad (4)$$

where  $d_i$  represents the algebraic ellipsoidal distance described below. For simplicity's sake, in the remainder of the paper, we will omit the  $i$  index for specific metaball sources wherever the context is unambiguous.

## 2.3 3-D Quadratic Distance Function

We use ellipsoidal primitives because they are simple and, at the same time, allow accurate modeling of human limbs with relatively few primitives because metaballs result in a smooth surface, thus keeping the number of parameters low. To express simply the transformations of these implicit surfaces that is caused by their attachment to an articulated skeleton, we write the ellipsoidal distance function  $d$  of Eq. 4 in matrix notation as follows. For a specific metaball and a state vector  $\Theta$ , we define the  $4 \times 4$  matrix

$$\mathbf{Q}_\Theta = \mathbf{L}_{\Theta^w, l} \cdot \mathbf{C}_{\Theta^w, l} . \quad (5)$$

where  $\mathbf{L}$  and  $\mathbf{C}$  are radii and position of the primitive respectively. The skeleton induced transformation  $\mathbf{S}_\theta$  is introduced as the rotation-translation matrix from the world frame to the frame to which the metaball is attached. These matrices will be formally defined in the appendix.

Given the  $\mathbf{Q}_\theta$  and  $\mathbf{S}_\theta$  matrices, we combine the quadric and the articulated skeleton transformations by writing the distance function of Eq. 3 as:

$$d(\mathbf{x}, \theta) = \mathbf{x}^T \cdot \mathbf{S}_\theta^T \cdot \mathbf{Q}_\theta^T \cdot \mathbf{Q}_\theta \cdot \mathbf{S}_\theta \cdot \mathbf{x} . \quad (6)$$

This formulation will prove key to effectively computing the Jacobians required to implement the optimization scheme of Section 3.

We can now compute the global field function  $F$  of Eq. 3 by plugging Eq. 6 into the individual field functions of Eq. 4 and adding up these fields for all primitives. In other words, the field function from which the model surface is derived can be expressed in terms of the  $\mathbf{Q}_\theta$  and  $\mathbf{S}_\theta$  matrices, and so can its derivatives as will be shown in the appendix. These matrices will therefore constitute the basic building blocks of our optimization scheme's implementation.

### 3 Optimization Framework

Our goal is to instantiate the degrees of freedom of our model so that it conforms as faithfully as possible to the image data derived from motion sequences such as the ones shown in Figure 3 and Figure 4. The expected output of our system is the instantiated state vector  $\theta$  of Eq. 1 that describes the model's shape and motion. This is a highly non-linear problem: The model consists of an articulated set of implicit surfaces. As a result it contains rotations in Euclidean space as well as quadratic and exponential distance functions. Simplifying the volumetric models, replacing the perspective transform by an orthographic one, and using a different representation for rotational joints can be used to linearize parts of the problem [3]. Such approaches, however, tend to lose in generality. Therefore, we chose to use a non-linear least squares estimator (LSE) to minimize the distance between the observations and the model. We implemented a variant of the standard Levenberg-Marquart least-squares solver [15] that can handle large number of unknowns by using sparse matrices.

In practice, we use the data to write  $n_{obs}$  observation equations of the form

$$F(\mathbf{x}, \theta) = T - \epsilon_i , \quad 1 \leq i \leq n_{obs} , \quad (7)$$

where  $F$  is the global field function of Eq. 3,  $T$  is the threshold of Eq. 2,  $\mathbf{x}$  is a data point, and  $\epsilon_i$  is an error term. We then minimize  $v^T P v$  where  $v = [\epsilon_1, \dots, \epsilon_{n_{obs}}]$  is the vector of residuals and  $P$  is a diagonal weight matrix associated to the observations. Because  $F$  is both well-defined and differentiable, these observations and their derivatives can be estimated both simply and without search using the matrix formalism of Section 2.3. This is valuable because our least-squares solver takes advantage of differential information for faster and more robust optimization, as do most powerful optimizers. The computation is outlined briefly

in the appendix and we refer the interested reader to our earlier publication [14] for additional details.

We now turn to the detailed implementation of the 3-D point and 2-D silhouette observations which are the main cues we obtain from the image sequences.

### 3.1 3-D Point Observations

Disparity maps such as those of Figure 1(d) are used to compute clouds of noisy 3-D points. Each one is used to produce one observation of the kind described by Eq. 7. Minimizing the corresponding residuals tends to force the fitted surface to be as close as possible to these points.

The properties of the chosen distance function allow the system to naturally deal with outliers and to converge even from rough initializations or estimates. The smooth shape of the inverted exponential that is used in our field function is responsible for both effects. It approaches zero asymptotically and, thus, provides an upper limit on the error resulting from distance between model and observation.

### 3.2 2-D Silhouette Observations

A silhouette point in the image defines a line of sight to which the surface must be tangential. Let  $\theta \in \Theta$  be an element of the state vector. For each value  $\theta$ , we define the implicit surface

$$\mathcal{S}(\theta) = \{[x, y, z]^T \in \mathbb{R}^3, F(x, y, z, \theta) = T\} . \quad (8)$$

Assuming that the line of sight is tangential to  $\mathcal{S}(\theta)$ , let  $[x(\theta), y(\theta), z(\theta)]$  be the contact point that is both on the line and on the surface. By definition,  $[x(\theta), y(\theta), z(\theta)]$  satisfies two constraints:

1. The point is on the surface, therefore  $F(x(\theta), y(\theta), z(\theta), \theta) = T$ .
2. The normal to  $\mathcal{S}(\theta)$  is perpendicular to the line of sight at  $[x(\theta), y(\theta), z(\theta)]$ .

We integrate silhouette observations into our framework by performing an initial search along the line of sight to find the point  $\mathbf{x}$  that is closest to the model in its current configuration. This point is used to add one of the observations described by Eq. 7. By construction, the point on the ray with the lowest value of field function  $F$  satisfies the second constraint as depicted by Figure 2(a).

In order to keep the second constraint satisfied during the optimization process, the Jacobian has to be constructed accordingly. A change in model position or size induces a motion of  $\mathbf{x}$  along the ray in order to remain the closest point on the ray with respect to the model. This involves computing first and second order derivatives for the Jacobian entries as will be discussed in the appendix. We have already seen in Figure 1 that silhouettes are crucial to constrain the search space. In Figure 2(b) we show a similar behavior in a 2-D context.

## 4 Robust Silhouette Tracker

Because our models are fully three-dimensional, given a position estimate at a particular time, we can predict where we expect to see silhouettes in a particular



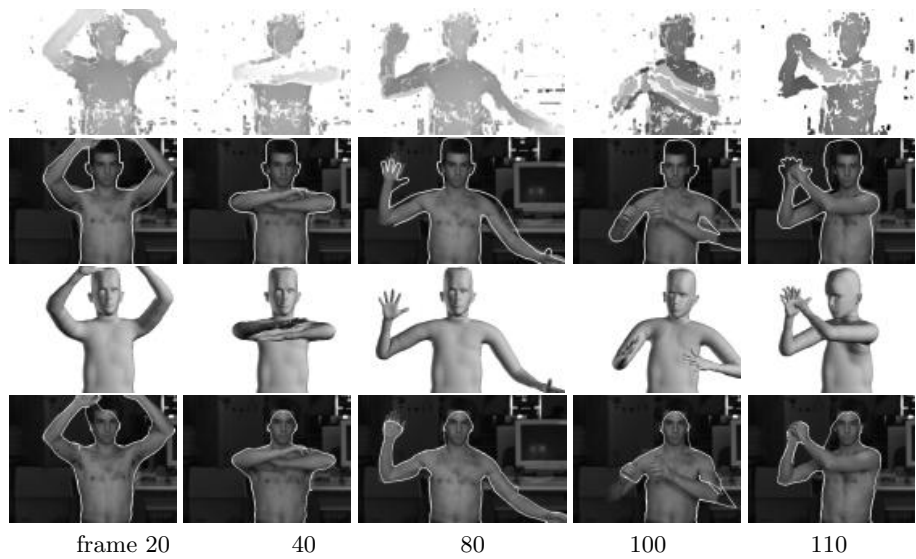
**Fig. 2.** Silhouette constraints. (a) Two lines of sight going through the camera optical center. In both cases, at the point that minimizes  $F$  along the line, the gradient of  $F$  is perpendicular to the line. But only in the case of the dashed line that is actually tangent to the surface, is this point on the surface and thus satisfies both criteria of Section 3.2. (b) The top row depicts the results of fitting an ellipse to slightly noisy 2-D points, shown as white crosses. The computed ellipses are much too large. The bottom row depicts the fit to the same points, but under the constraint that the ellipse be tangent to the white lines. The results are much improved.

image and we use this prediction to constrain the search for the silhouettes to restricted areas in the image. In Section 4.1, we discuss this approach when run *a posteriori*, that is, by first tracking the person’s body using stereo only, computing the model’s outline in each frame and using it as an initial guess for the silhouette’s location. This “naive” approach yields good results if the initial tracking was accurate enough but fails otherwise. In Section 4.3, we therefore introduce a more effective approach in which, for each frame, we use the model derived from the previous frame to estimate the silhouette and immediately use this silhouette to guide the recovery of the new current model position. We will show that this increases the tracker’s robustness and prevents it from making mistakes in ambiguous situations.

#### 4.1 Simple Model-Based Approach for Silhouette Extraction

Given a first fit of the model to the data, one can take advantage of it to extract additional information from the images. In this case, we can first track using stereo alone. In the absence of gross tracking errors, the projected model’s outlines can be used as an initial estimate for the silhouette location. We ran a number of experiments in which we used this estimate to initialize an active contour. We found that, in practice, when the background is cluttered, the active contour does not in general converge towards the desired outline because it gets stuck into undesirable local minima. To remedy this situation, instead of running the snake on the original image, we first filtered it using the technique described in Section 4.2 and, then, used the result to deform the contour.

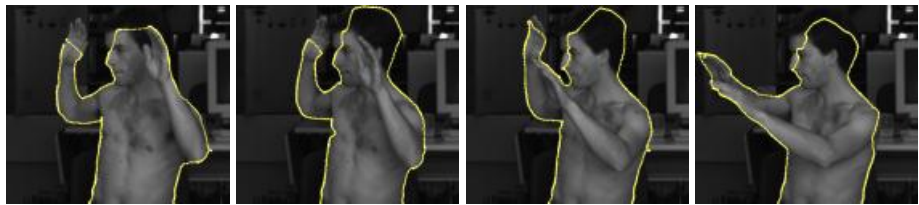
Figures 3 and 4 depict the results of running this algorithm on the 600 frames of two different video sequences. For most frames, deforming the predicted outline using the filtered the image resulted in an outline that was very close to the true silhouette. We saw errors in only 19 frames. Given the fact that the motions are complex and that the background is cluttered, this shows the robustness of our algorithm. Note also, that we did not use an image of the background without the subject to perform any kind of background subtraction.



**Fig. 3.** Extracting the subject’s silhouette using the simple model-based approach of Section 4.1. Frames 20, 40, 80, 100 and 110 of the a 300-frame sequence are shown. In the upper row are the disparity maps, in the second row are the outlines of the projected model, which is shown in the third row. This outline was fitted to the filtered image gradient in the last row. The system correctly tracks until frame 100 and then recovers before frame 110. The more sophisticated approach of Section 4.3 will overcome the error in the 100th frame.

The result for frame 100 in Figure 3 shows typical problem areas. The errors around the left arm are most interesting. The model was relatively far away from the actual position. A second circular silhouette curve at the inside of the model’s arm was extracted as well. It corresponds to nothing in the original image and evolves arbitrarily during snake optimization. The problem is compounded by the fact that the table in the background is very close to the subject. It is not removed totally by the low level filtering steps but it is considered as foreground instead.

A different problem occurred around the subject’s right elbow. It is cut because the model projected slightly inside the real contour so that the smoothness coefficient of the snake forced it to retract from the sections of high curvature.



**Fig. 4.** Automatically extracted silhouette contours in a few frames of another 300-frame sequence.

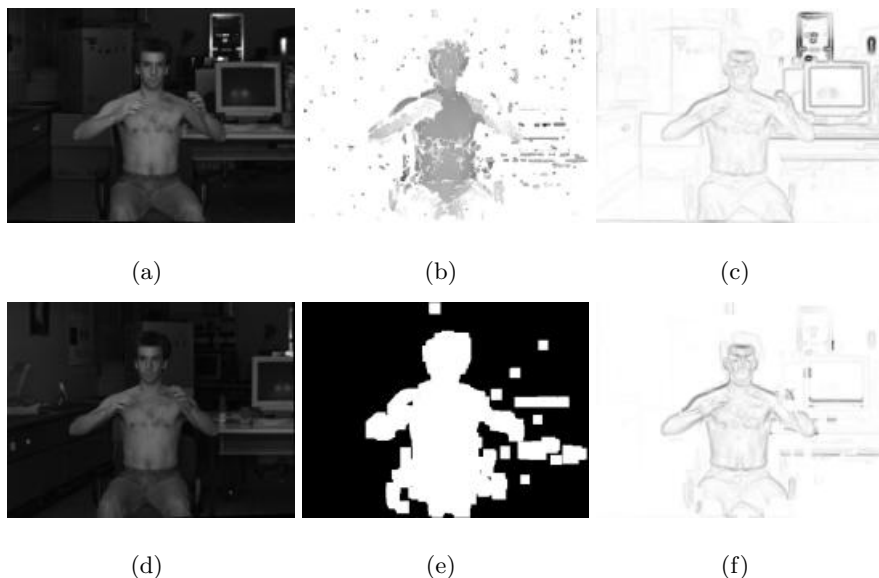
Tuning the ratio of the snake parameters can delay the onset of such problems, but not prevent them altogether.

This particular example clearly demonstrates the problems that arise when trying to extract higher-level information from image data in an unsupervised environment. Using the model is a tremendous advantage but, still, it is not enough to ensure correct behavior. We address this problem in Section 4.3.

## 4.2 Disparity-Based Gradient Filtering

As discussed above, in the presence of a cluttered and unknown background, even a good initial body-outline estimate does not guarantee convergence of an active contour towards the correct solution. To improve the snake algorithm's performance, we filter the image to eliminate edges that are unlikely to be silhouettes but can still trap the active contour into an undesirable local minimum. To this end we combine gradient and depth information as follows.

The main input to our system is 3-D data derived from disparity maps. It is therefore natural to also use those maps to perform a form of background subtraction. This, however, is non trivial: Because of noise, absence of texture, specularities and non fronto-parallel surfaces, these maps are typically neither dense nor reliable enough to robustly separate foreground and background. Furthermore, objects located at similar distances as the subject may not be eliminated that way. This is certainly true of the disparity map of Figure 5(b).



**Fig. 5.** Disparity-based gradient filtering. (a) and (d) is a original stereo pair and (b) is the corresponding disparity map which is binarized and morphologically “cleaned” in (e). (c) is the gradient magnitude of (a), and (f) the result of filtering (c) with (e).

We can nevertheless use such disparity maps to find image pixels with a large probability to correspond to the subject: When computing the map, we chose a range of disparities that eliminates distances that correspond to the far background. Those areas for which no acceptable disparity value can be found appear mostly in black. The correlation-based stereo algorithm we use [10] can be limited to a specified reconstruction volume and a Right-Left Check eliminates most erroneous matches. Therefore, by binarizing of the disparity map and cleaning it up using morphological operators, we obtain binary masks such as the one shown in Figure 5(e). Disparity computation was done only on a given volume excluding the far background. However, objects close to the subject are still included and parts of the subject are missing due to bad texture or shadows.

Applying the filter to the gradient image eliminates most parts of the background as shown in Figure 5(f). Finally, we perform a hysteresis thresholding that is comparable to the one used in Canny’s edge-detector [4]: Only gradient image pixels are accepted that are above an upper threshold or which are above a lower threshold and have a point that passes the previous test in their immediate neighborhood.

### 4.3 Joint Shape and Silhouette Extraction

To solve problems such as those shown in Figure 3, we take the temporal evolution of the contour into account, instead of entirely relying on a good fit of the model. We therefore modify the scheme for silhouette extraction as follows:

1. Silhouette of previous frame serves as initialization for current frame
2. Optimize using active contours on disparity-filtered gradient image
3. Fit body model to stereo data constraint by current silhouette estimate
4. Optimize silhouette of fitted model using again active contours

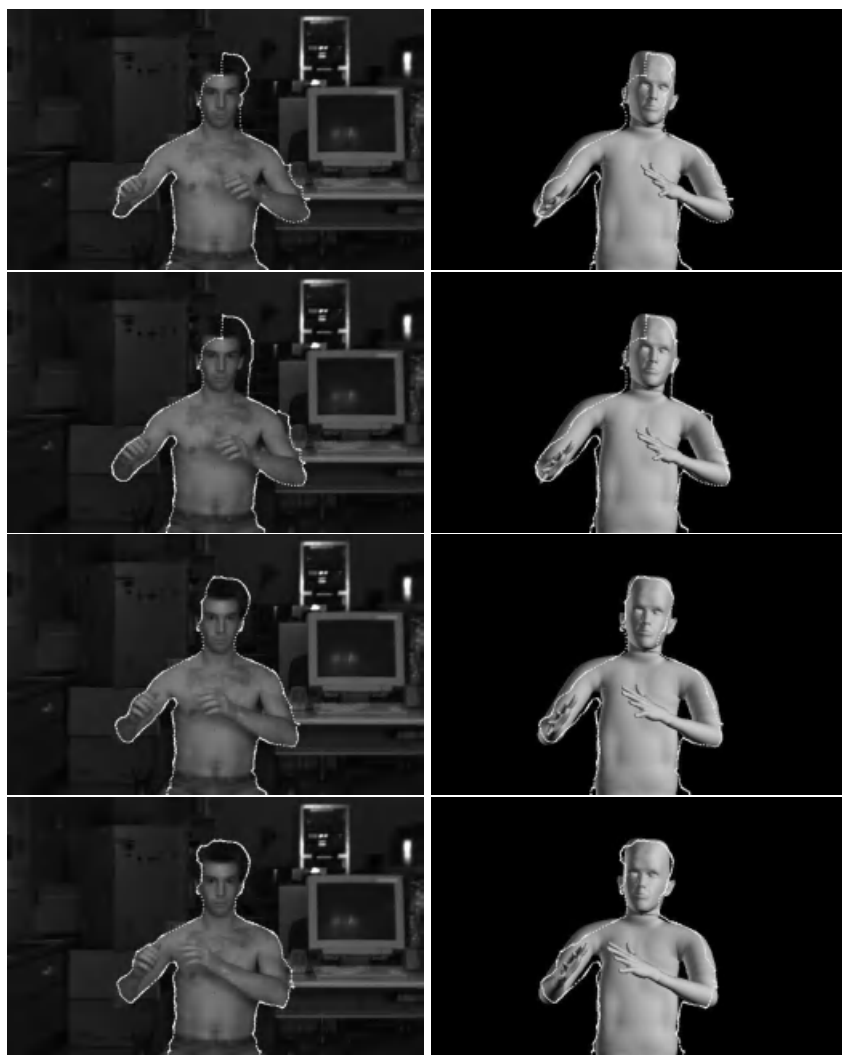
The initial guess of the silhouette position is now taken to be the result of the silhouette extraction in the previous frame. Because, at frame rate, differences between successive images are small the actual silhouette is close to the previous one. Assuming a correctly recovered position of the silhouette in the previous frame we can directly feed this silhouette as initialization to an active contour algorithm for the current frame.

Again, this step alone does not ensure robust silhouette extraction. The active contour may still miss the silhouette outline of the person. We therefore rerun our fitting algorithm on the 3-D stereo data augmented by this—possibly incorrect—silhouette outline. Thanks to correct segments of the new silhouette, the implicit treatment of outliers, and the strong stereo information, in our experience the system is able to find the correct pose of the model.

In a last step the model in its correct pose is projected into the camera frame. Using the model silhouette as input to a second run of an active contour algorithm results in the final silhouette outline of the person. Results of this model-based silhouette tracker are shown in Figure 6. The original image together with the extracted silhouette of the person are shown in the left column.



The recovered model pose as well as the silhouettes are shown in the right column. These results show that this new algorithm is able to overcome the errors that occurred during the simple model-based approach of Section 4.1.



**Fig. 6.** Tracking results using the model-based silhouette extraction method of Section 4.3. Frames 97, 98, 99 and 100 of the sequence in Figure 3 are shown. Snake-optimized contours are overlaid on the original images on the left as well as the resulting model on the right.

## 5 Conclusion

We have presented a model-based technique for robustly extracting silhouette contours of people in motion. By integrating the model early on in the processing

pipeline our system overcomes the common pitfalls of low-level image processing. We do not have to engineer the background and the system could robustly extract the silhouette of the person even in a dynamic scene. The explicit modeling of 3-D geometry lets us predict the expected location of image features and occluded areas, thereby making the extraction algorithm more robust. These silhouettes are used in turn to constrain model reconstruction, thereby further increasing its precision and reliability.

A key component of our implementation is our ability to analytically and precisely compute all required derivatives. This helps the optimizer to accurately estimate the actual variation of the distance between model and data according to parameter change and, thus, to minimize it with a minimal number of iterations. In contrast to most other work where the derivatives are obtained by perturbing the parameters and re-evaluating the distance function, computing the Jacobian analytically necessitates fewer computations. Furthermore, the derivatives in our modular matrix notation contain many identical parts and intermediate computation results can be reused to further speed-up the process.

In future work, we intend to test our system on sequences with highly cluttered and dynamic background. Our results are promising but more tests have to be effected to analyze the limits of the system. Currently, no provisions for occlusion and limb self-occlusion are given. The depth information from the disparity maps was sufficient to successfully track in our test sequences but explicit occlusion detection would be needed to robustly track more complex motions.

## Appendix: Differentiating the Global Field Function

To illustrate the efficient computation of the Jacobian entries for the least squares estimator we present the computation of the metaball distance function as well as its first and second order derivatives with respect to a rotational joint paramter.

To express simply the transformations of the implicit surfaces caused by their attachment to an articulated skeleton, we write the ellipsoidal distance function  $d$  of Eq. 3 in matrix notation as follows. This formulation will prove key to effectively computing the Jacobians required to implement the optimization scheme regarding the different data constraints.

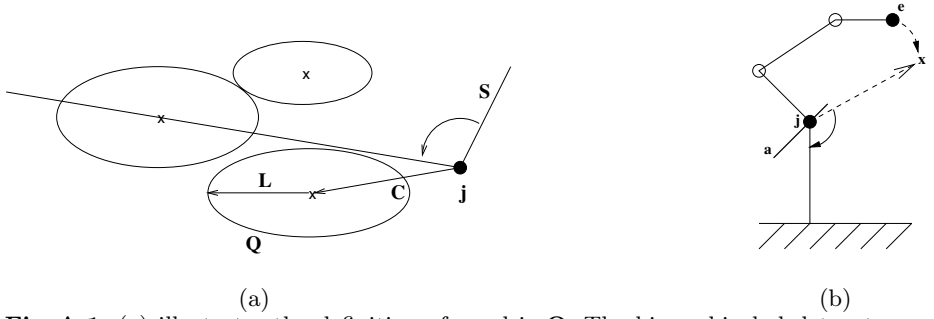
Recall that the distance function of Equation 3 can be written as follows:

$$d(\mathbf{x}, \Theta) = \mathbf{x}^T \cdot \mathbf{S}_\Theta^T \cdot \mathbf{Q}_\Theta^T \cdot \mathbf{Q}_\Theta \cdot \mathbf{S}_\Theta \cdot \mathbf{x} .$$

$d(\mathbf{x}, \Theta)$  defines an ellipsoidal quadratic distance field. For a given primitive and the state vector  $\Theta$ , the  $4 \times 4$  matrix  $\mathbf{Q}_\Theta$  contains the location of the center  $\mathbf{C}$  and the scaling  $\mathbf{L}$  along the principal axes respectively. Figure A.1(a) illustrates the concept.

$\mathbf{L}\mathbf{C}_{\Theta^w, l} = \mathbf{L}_{\Theta^w, l} \cdot \mathbf{C}_{\Theta^w, l}$  is the scaling and translation along the principal axes:

$$\mathbf{L}\mathbf{C}_{\Theta^w, l} = \begin{bmatrix} \frac{1}{\theta^w l_x} & 0 & 0 & -\theta^w c_x \\ 0 & \frac{1}{\theta^w l_y} & 0 & -\theta^w c_y \\ 0 & 0 & \frac{1}{\theta^l l_z} & -\theta^l c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$



**Fig. A.1.** (a) illustrates the definition of quadric  $Q$ . The hierarchical skeleton transformation  $S$  up to joint  $j$  is followed by the centroid transformation  $C$  and by  $L$  describing the quadric's volume. (b) depicts the articulated structure  $S$ . The distance between model surface  $e$  and observation  $x$  is to be minimized subject to rotation  $\theta^r$  around axis  $a$  of joint  $j$ .

where  $L = (l_x, l_y, l_z)$  are the radii of an ellipsoid, i.e. half the axis length along the principal directions and  $C = (c_x, c_y, c_z)$  is the primitive's center. Coefficients  $\theta^l$  and  $\theta^w$  from the state vector  $\Theta$  control relative *length* and *width* of a metaball. They are shared among groups of metaballs according to segment assignment.  $S_{\Theta^{l,r}}$  is the skeleton induced transformation, a  $4 \times 4$  rotation-translation matrix from the world frame to the frame to which the metaball is attached. Given rotation  $\theta^r \in \Theta$  of a joint  $J$ , we write:

$$S_{\Theta} = E \cdot J_{\theta^r} = E \cdot R_{\theta^r} \cdot J_{0^r}, \quad (A.1)$$

where  $E$  is the homogenous  $4 \times 4$  transformation from the joint frame to the metaball frame.  $J_{\theta^r}$  is the transform from world frame to joint frame, including the rotation parameterized by  $\theta^r$  and  $R_{\theta^r}$  is the homogenous rotation matrix of  $\theta^r$  around axis  $a$  with  $J_{0^r} = R_{\theta^r}^{-1} \cdot J_{\theta^r}$ . A configuration of an articulated structure is depicted by Figure A.1(b).

We can now compute the global density function  $F$  of Equation 3 by plugging Equation 6 into the individual field functions of Equation 4 and adding up these fields for all primitives. In other words, the field function from which the model surface is derived can be expressed in terms of the  $Q_{\Theta}$  matrices, and so can its derivatives as will be seen later. These matrices will therefore constitute the basic building blocks of our optimization scheme's implementation.

### A.1 Derivatives of the Distance Function

The generic derivative of the distance function wrt. to any parameter  $\theta \in \Theta$  from the state vector can be computed as:

$$\frac{\partial}{\partial \theta} d(x, \Theta) = 2 \cdot x^T \cdot S_{\Theta}^T \cdot Q_{\Theta}^T \cdot \left[ \frac{\partial}{\partial \theta} Q_{\Theta} \cdot S_{\Theta} \right] \cdot x \quad (A.2)$$

The computation includes parts of the distance function and intermediate results can be cached and reused.

The model pose is exclusively defined by the state of the underlying articulated structure  $\mathbf{S}$  of Eq. A.1. A rotational parameter  $\theta^r$  defines the angle between two body parts. For differentiation classical robotics methods can be applied, such as [5]. Combined with the quadric the rotational derivative of Eq. A.2 can be shown to be:

$$\left[ \frac{\partial}{\partial \theta^r} \mathbf{Q}_\theta \cdot \mathbf{S}_\theta \right] \cdot \mathbf{x} = \mathbf{Q} \cdot \mathbf{E} \cdot \mathbf{a} \times \mathbf{j}\mathbf{x} . \quad (\text{A.3})$$

with rotational axis  $\mathbf{a}$  and  $\mathbf{j}\mathbf{x} = \mathbf{J}_\theta \cdot \mathbf{x}$  being the vector from joint center to observation.  $\mathbf{E}$  is the transformation from the joint frame to the metaball frame. See Figure A.1(b) for an illustration. Equation A.3 can be efficiently implemented because it only consists of a simple vector cross-product transformed into the quadric's frame. For more details we refer the interested reader to our earlier publication [14].

## A.2 Silhouette Constraint

As introduced in Section 3.2 the motion of the silhouette point  $\mathbf{x}$  along the ray has to be taken into account during optimization. For non-articulated implicit surfaces this has been shown by [16]. This involves computing first and second order derivatives for the Jacobian entries. This turns out to be prohibitively expensive when done in a brute force manner. In contrast, our modular matrix formulation allows an elegant description of these derivatives because they retain their modularity and can be computed similarly to the first order derivatives. Again, intermediate results of the function evaluation as well as the computation of the first order derivatives can be reused to speed up the process.

Second order derivatives with respect to spatial coordinates can be computed according to Eq. A.4 and those with respect to a spatial coordinate as well as a parameter of the state vector according to Eq. A.5. Please see our earlier publication [14] for a full derivation of the Jacobian for the field functions.

$$\frac{\partial^2 d}{\partial x_i \partial x_j} = 2 * \frac{\partial \mathbf{x}^T}{\partial x_i} \cdot \mathbf{S}_\theta^T \cdot \mathbf{Q}_\theta^T \cdot \mathbf{Q}_\theta \cdot \mathbf{S}_\theta \cdot \frac{\partial \mathbf{x}}{\partial x_j} , \quad (\text{A.4})$$

$$\frac{\partial^2 d}{\partial x_i \partial \theta} = 2 * \mathbf{x}^T \cdot \left( \left[ \frac{\partial}{\partial \theta} \mathbf{S}_\theta^T \cdot \mathbf{Q}_\theta^T \right] \cdot \mathbf{Q}_\theta \cdot \mathbf{S}_\theta + \mathbf{S}_\theta^T \cdot \mathbf{Q}_\theta^T \cdot \left[ \frac{\partial}{\partial \theta} \mathbf{Q}_\theta \cdot \mathbf{S}_\theta \right] \right) \cdot \frac{\partial \mathbf{x}}{\partial x_i} (\text{A.5})$$

These derivatives are necessary to correctly integrate 2-D silhouette observations in a 3-D optimization framework. When replacing them with simpler, first order ones we experienced incorrect estimation of the closest point on the ray and the optimizer did not converge. Using the complete derivatives allows the optimizer to more precisely estimate the system's reaction to parameter change and, thus, to find the optimal state with a minimal number of iterations.

## References

1. J.K. Aggarwal and Q. Cai. Human motion analysis: a review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

2. J. F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.
3. Ch. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 1998.
4. J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986.
5. J.J. Craig. *Introduction to robotics: mechanics and control*, chapter 5. Electrical and Computer Engineering. Addison-Wesley, 2nd edition, 1989.
6. L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Third International Workshop on Co-operative Distributed Vision*, November 1999.
7. Q. Delamarre and O. Faugeras. 3D Articulated Models and Multi-View Tracking with Silhouettes. In *International Conference on Computer Vision*, Corfu, Greece, September 1999.
8. J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *CVPR*, Hilton Head Island, SC, 2000.
9. T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.
10. P. Fua. From Multiple Stereo Views to Multiple 3-D Surfaces. *International Journal of Computer Vision*, 24(1):19–35, August 1997.
11. D.M. Gavrilu. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1), January 1999.
12. I. Kakadiaris and D. Metaxas. 3D Human Body Model Acquisition from Multiple Views. In *International Conference on Computer Vision*, 1995.
13. T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), March 2001.
14. R. Plänkers and P. Fua. Articulated Soft Objects for Video-based Body Modeling. In *International Conference on Computer Vision*, pages 394–401, Vancouver, Canada, July 2001.
15. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes, the Art of Scientific Computing*. Cambridge U. Press, Cambridge, MA, 1986.
16. S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3-d. object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1183–1196, December 1994.
17. D. Thalmann, J. Shen, and E. Chauvineau. Fast Realistic Human Body Deformations for Animation and VR Applications. In *Computer Graphics International*, Pohang, Korea, June 1996.

# On the Non-linear Optimization of Projective Motion Using Minimal Parameters

Adrien Bartoli

INRIA Rhône-Alpes  
655, av. de l'Europe  
38334 St. Ismier cedex, France.  
Adrien.Bartoli@inria.fr  
[www.inrialpes.fr/movi/people/Bartoli](http://www.inrialpes.fr/movi/people/Bartoli)

**Abstract.** I address the problem of optimizing projective motion over a minimal set of parameters. Most of the existing works overparameterize the problem. While this can simplify the estimation process and may ensure well-conditioning of the parameters, this also increases the computational cost since more unknowns than necessary are involved.

I propose a method whose key feature is that the number of parameters employed is minimal. The method requires singular value decomposition and minor algebraic manipulations and is therefore straightforward to implement. It can be plugged into most of the optimization algorithms such as Levenberg-Marquardt as well as the corresponding sparse versions. The method relies on the orthonormal camera motion representation that I introduce here. This representation can be locally updated using minimal parameters.

I give a detailed description for the implementation of the two-view case within a bundle adjustment framework, which corresponds to the maximum likelihood estimation of the fundamental matrix and scene structure. Extending the algorithm to the multiple-view case is straightforward. Experimental results using simulated and real data show that algorithms based on minimal parameters perform better than the others in terms of the computational cost, i.e. their convergence is faster, while achieving comparable results in terms of convergence to a local optimum. An implementation of the method will be made available.

## 1 Introduction

The problem of recovering structure and motion from images is one of the central challenges for computer vision. The use of image feature correspondences (e.g. points, lines) through different views and the study of geometrical aspects of the image formation process have led to numerous techniques acting in metric, affine or projective space, depending on whether camera calibration is fully, partially or not available.

Most of the time, a sub-optimal solution is obtained using linear techniques for motion then for structure recovery [3] or jointly [16] and subsequently refined. One of the most efficient techniques for such a structure and motion optimization is bundle adjustment. It involves minimizing a non-linear cost function based on the discrepancy between reprojected and original image features [15,17]. The behaviour of such techniques, in terms of convergence properties to a local optimum of the cost function and

computational cost, greatly depends on the algebraic representation of the problem, i.e. of structure and motion, and in particular, numerical conditioning and whether the number of parameters employed is minimal. For bundle adjustment, preserving the original noise model on image features is also crucial.

While bundle adjustment is theoretically well-defined (it corresponds to the maximum likelihood estimator, see e.g. [17]) there are still practical optimization problems since the employed cost functions have many local minima [13] where optimization processes may get trapped.

I address the problem of representing motion. The goal is to obtain minimal estimators, i.e. where the number of parameters considered for optimization is minimal. I focus on the projective case, i.e. when camera calibration is unknown. This topic is of primary importance since an accurate projective reconstruction is necessary to subsequently succeed in self-calibration.

Consider two perspective views of a rigid scene. The image signature of their relative position is the projective two-view motion or the epipolar geometry, usually described by the  $(3 \times 3)$  rank-2 and homogeneous fundamental matrix [10,18]. A fundamental matrix has 7 degrees of freedom. Therefore, 7 parameters should be enough to optimize the projective two-view motion.

However, it has been seen that there does not exist a universal parameterization of the fundamental matrix using 7 parameters. This is due to the non-linear rank-2 constraint and the free-scale ambiguity. Existing works may fall into the following categories:

- overparameterization, e.g. the 12 entries of a perspective camera matrix [7]. More unknowns than necessary are involved to simplify the representation;
- multiple minimal parameterizations, 3 in [2] or 36 in [18];
- minimal parameterizations combined to image transformations [2,19] to reduce the number of parameterizations.

Other techniques optimize over the 9 parameters of the fundamental matrix while adding the non-linear rank-2 constraint and the normalization constraint as virtual measurements.

This paper makes the following contributions.

Firstly, I address the projective two-view motion case in §2. I introduce what I call the orthonormal representation of projective two-view motion. Based on this, I show how one can non-linearly estimate the projective two-view motion using a minimal number of 7 parameters. An important point is that this method does not depend upon the optimization technique considered.

Secondly, I illustrate the use of this method in a bundle adjustment framework based on [8] in §3. The result is a minimal maximum likelihood estimator for the fundamental matrix as well as scene structure. The reader who is interested into practical issues only should refer directly to this section.

Thirdly, I extend the framework to multiple views in §4 where I introduce the orthonormal representation of projective multiple-view motion. I derive, similarly to the two-view case, a means to perform optimization over a minimal set of parameters.

Finally, experimental results on simulated and real data are shown in §§5 and 6 respectively. They show that algorithms based on minimal motion parameters, and in

particular on the orthonormal representation, perform better than the others, in terms of computational cost, while achieving equivalent performances in terms of convergence properties. This is followed by my conclusions in §7.

## 2 The Projective Two-View Motion

### 2.1 Preliminaries

Let us consider two  $(3 \times 4)$  uncalibrated perspective camera matrices. Due to homogeneity, each one has 11 degrees of freedom. Since there is a 15-degrees of freedom coordinate frame ambiguity on structure and motion, the projective two-view motion has  $11 \cdot 2 - 15 = 7$  degrees of freedom.

The corresponding  $(3 \times 3)$  fundamental matrix  $F$  has 9 entries but 7 degrees of freedom since it is homogeneous and has rank 2. It allows one to extract projection matrices for the two views while fixing the coordinate frame. These projection matrices constitute a realization of the fundamental matrix. Among the 15-parameter family of realizations, a common choice is the canonic projection matrices  $P$  and  $P'$  [11]:

$$P \sim (I_{(3 \times 3)} \quad \mathbf{0}_{(3 \times 1)}) \text{ and } P' \sim (H^* \quad \mathbf{e}'), \quad (1)$$

where  $\mathbf{e}'$  is the second epipole defined as the right kernel of  $F$ ,  $F^T \mathbf{e}' \sim \mathbf{0}_{(3 \times 1)}$  and  $H^* \sim [\mathbf{e}']_{\times} F$  is the canonic plane homography. This defines the canonic coordinate frame which is unique, provided normalization constraints for  $H^*$  and  $\mathbf{e}'$ . It will be seen in §2.3 that  $\|\mathbf{e}'\|^2 = 1$  and  $\|H^*\|^2 = \gamma$ , where  $\gamma$  is an unknown constant scalar, is a convenient choice for my method. Note that  $\sim$  means “equal up to scale” and  $[\cdot]_{\times}$  is the cross-product skew-symmetric  $(3 \times 3)$ -matrix. All entities are represented in homogeneous coordinates, i.e. are defined up to scale.

### 2.2 Relation to Previous Work

Most of the previous work on minimally parameterizing projective two-view motion deals with directly parameterizing the epipolar geometry. The fundamental matrix  $F$  is decomposed into the epipoles  $\mathbf{e}$  and  $\mathbf{e}'$  and the epipolar transformation, which is a 1D projective transformation relating the epipolar pencils, represented by an homogeneous  $(2 \times 2)$ -matrix  $g$ .

Representing these entities with minimal parameters requires eliminating their homogeneity. This is usually done by normalizing each of them so that their largest entry is unity, which yields 3 possibilities for each epipole and 4 for the epipolar transformation, so  $3 \cdot 3 \cdot 4 = 36$  possible parameterizations.

Obtaining the fundamental matrix [or any other 2D entity such as the extended epipolar transformation [2] or the canonic plane homography  $H^*$ ] from  $\mathbf{e}$ ,  $\mathbf{e}'$  and  $g$  requires then the use of 9 distinct parameterizations to model all cases [18]. These cases coincide with 9 of the 36 previous ones.

In [10], the author proposes to restrict the two-view configurations considered to the cases where both epipoles are finite and can therefore be expressed in affine coordinates. Due to the homogeneity of the epipolar transformation, 4 distinct parameterizations are



still necessary for  $\mathbf{g}$ . A unique parameterization can then be used to form the fundamental matrix.

The method has been extended in [18] to the general case, i.e. when the epipoles can be either finite or infinite. In this case, it is shown that the full 36 distinct parameterizations are necessary. This leads to a cumbersome and error-prone implementation of the optimization process.

In [2,19], the method has been revised so as to reduce the number of parameterizations using image transformations. In [2], the image transformations used are metric and the number of distinct parameterizations restricted to 3 plus one bilinear constraint on the entries of  $\mathbf{g}$ , while in [19], the transformations used are projective, which allows one to reduce the number of parameterizations to 1. The main drawback is that in the transformed image space, the original noise model on the image features is not preserved. A means to preserve it, up to first order approximation, has been proposed in [19] for the gradient-weighted criterion, which is not the one used for bundle adjustment.

### 2.3 The Orthonormal Representation

*Derivation.* I introduce what I call the orthonormal representation of projective two-view motion. I consider the fundamental matrix representation of the motion. Any  $(3 \times 3)$  rank-2 matrix is a fundamental matrix, i.e. represents a motion. Conversely, a projective two-view motion is represented by a unique fundamental matrix (up to scale). Therefore, deriving a minimal representation of projective two-view motion from its fundamental matrix representation implies considering two constraints; the rank-2 constraint and a normalization constraint, which fixes the relative scale of the fundamental matrix. Previous work has shown that these constraints are quite tricky to enforce directly on the fundamental matrix [2,10,18,19].

To overcome this problem, instead of considering directly the fundamental matrix, I rather analyze its singular value decomposition  $\mathbf{F} \sim \mathbf{U}\Sigma\mathbf{V}^T$ , where  $\mathbf{U}$  and  $\mathbf{V}$  are  $(3 \times 3)$  orthonormal matrices and  $\Sigma$  a diagonal one containing the singular values of  $\mathbf{F}$ . The orthonormal representation is then derived while enforcing the constraints:

- *rank-2*: since  $\mathbf{F}$  is a rank-2 matrix,  $\Sigma \sim \text{diag}(\sigma_1, \sigma_2, 0)$  where  $\sigma_1$  and  $\sigma_2$  are strictly positive scalars and  $\sigma_1 \geq \sigma_2 > 0$ ;
- *normalization*: since  $\mathbf{F}$  is an homogeneous entity, I can scale its singular value decomposition such that  $\mathbf{F} \sim \mathbf{U} \cdot \text{diag}(1, \sigma, 0) \cdot \mathbf{V}^T$  where  $\sigma = \sigma_2/\sigma_1$  and  $0 < \sigma \leq 1$  ( $\sigma_1 \neq 0$  since  $\mathbf{F}$  is rank-2).

This decomposition shows that any projective two-view motion can be represented by two  $(3 \times 3)$  orthonormal matrices and a scalar.

This gives the orthonormal representation of projective two-view motion as:

$$(\mathbf{U}, \mathbf{V}, \sigma) \in \mathbb{F} \text{ where } \mathbb{F} \equiv O(3)^2 \times \{\sigma \mid 0 < \sigma \leq 1\}, \quad (2)$$

where  $O(3)$  is the Lie group of  $(3 \times 3)$  orthonormal matrices. This representation is minimal in that it has  $3 + 3 + 1 = 7$  degrees of freedom. It can easily be computed from the singular value decomposition of the fundamental matrix. Note that  $\sigma = 1$  may correspond to the case of an essential matrix, i.e. when cameras are calibrated.

Any element of  $\mathbb{F}$  represents a unique two-view motion since it can be used to recompose a unique fundamental matrix (see next paragraph), i.e. a  $(3 \times 3)$  matrix where both the rank-2 and a normalization constraints have been enforced. However, a fundamental matrix has more than one orthonormal representations. For instance, given an orthonormal representation  $(U, V, \sigma) \in \mathbb{F}$ , one can freely switch the signs of  $\mathbf{u}_3$  or  $\mathbf{v}_3$  while leaving the represented motion invariant. However, the space of fundamental matrices and the orthonormal representation are both 7-dimensional, which allows for minimal estimation.

*Recovering 2D entities.* The fundamental matrix corresponding to an orthonormal representation  $(U, V, \sigma) \in \mathbb{F}$  can be recovered by simply recomposing the singular value decomposition:

$$F \sim \mathbf{u}_1 \mathbf{v}_1^T + \sigma \mathbf{u}_2 \mathbf{v}_2^T, \quad (3)$$

where  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are the columns of  $U$  and  $V$  respectively. Among all potential applications of the orthonormal representation, I will use it for bundle adjustment. Therefore, I will need to extract projection matrices from the fundamental matrix. This can be achieved directly from the orthonormal representation by recovering the second epipole and the canonic plane homography of [11], equation (1).

The second epipole is the last column of  $U$ :  $\mathbf{e}' \sim \mathbf{u}_3$ , while the canonic plane homography can be formulated as  $H^* \sim [\mathbf{e}']_{\times} F \sim [\mathbf{u}_3]_{\times} (\mathbf{u}_1 \mathbf{v}_1^T + \sigma \mathbf{u}_2 \mathbf{v}_2^T)$ . Since  $U$  is an  $O(3)$  matrix,  $[\mathbf{u}_3]_{\times} \mathbf{u}_1 = \pm \mathbf{u}_2$  and  $[\mathbf{u}_3]_{\times} \mathbf{u}_2 = \mp \mathbf{u}_1$  which yields the canonic plane homography as  $H^* \sim \mathbf{u}_2 \mathbf{v}_1^T - \sigma \mathbf{u}_1 \mathbf{v}_2^T$  and the second projection matrix as:

$$P' \sim (\mathbf{u}_2 \mathbf{v}_1^T - \sigma \mathbf{u}_1 \mathbf{v}_2^T \mid \mathbf{u}_3). \quad (4)$$

Normalization constraints discussed in §2.1 are clearly satisfied. This guarantees that the same canonic basis will be used through the optimization process.

## 2.4 Estimation Using Minimal Parameters

In this section, I use the previously described orthonormal representation of projective two-view motion to locally update a “base” estimate using the minimum 7 parameters.

*Update using minimal parameters.* Before going further, let us examine the case of 3D rotations. There does not seem to exist a minimal, complete and non-singular parameterization of the 3-dimensional set of rotations in 3D space. For example, consider the  $(3 \times 3)$  rotation matrix  $R \in SO(3)$ . Minimal representations, such as the 3 Euler angles  $\boldsymbol{\theta}$  lead to singularities. However, one can find representations that are locally minimal and non-singular, e.g. in a neighbourhood of  $R = I_{(3 \times 3)}$ , i.e.  $\boldsymbol{\theta} = \mathbf{0}_{(3 \times 1)}$ . Therefore, most of the estimation processes of 3D rotations do not minimally parameterize the current estimate, but rather locally update its overparameterization. A typical example is to use a  $(3 \times 3)$  rotation matrix representation that is locally updated as  $R \leftarrow R \cdot R(\boldsymbol{\theta})$  where  $R(\boldsymbol{\theta})$  is any minimal and locally non-singular parameterization of 3D rotations, e.g. Euler angles. This method is used in [6] for the non-linear recovery of metric structure and motion, where  $R$  is updated and  $\boldsymbol{\theta}$  reset to zero after each iteration of the optimizer.

The same scenario arises for the projective two-view motion. There does not seem to exist a minimal, complete and non-singular parameterization of the corresponding 7-dimensional space. Consequently, I propose locally updating a given estimate using minimal parameters.

Let us consider the orthonormal representation  $(U, V, \sigma) \in \mathbb{F}$  of a given projective two-view motion. Each matrix  $(U, V) \in O(3)^2$  can be locally updated using 3 parameters by considering the method described above for 3D rotations. The scalar  $\sigma \in \{\sigma \mid 0 < \sigma \leq 1\}$  is completely included into the parameterization. A means to ensure  $0 < \sigma \leq 1$  is described below.

*Completeness.* A first remark that immediately follows about the above-proposed method is whether all two-view configurations are covered. This arises from the fact that  $U$  and  $V$  are  $O(3)$  matrices, which may have positive or negative determinants, and are updated using  $SO(3)$  matrices,  $R(\mathbf{x})$  and  $R(\mathbf{y})$  respectively, which have only positive determinants. Therefore, the signs of  $\det(U)$  and  $\det(V)$  are not allowed to change during the optimization process. I claim that this is not a problem and that any fundamental matrix  $F$  can be reached from any initial guess  $F_0$ , even if  $\text{sign}(\det(U_0)) \neq \text{sign}(\det(U))$ ,  $\text{sign}(\det(V_0)) \neq \text{sign}(\det(V))$  or both. To prove this claim, I show that any fundamental matrix  $F$  represented by  $(U, V, \sigma)$  has alternative representations where the signs of either  $\det(U)$  or  $\det(V)$ , or both, have been switched. This is due to the non-uniqueness of the singular value decomposition, see e.g. [14]. Consider the recomposition equation (3) and observe that  $\mathbf{u}_3$  and  $\mathbf{v}_3$ , the third columns of  $U$  and  $V$  respectively, do not affect the result. Therefore, they are only constrained by the orthonormality of  $U$  and  $V$ . Hence, their signs can be arbitrarily switched, which accordingly switches the sign of the determinant of the corresponding matrix. For example,  $\mathbf{u}_3 \leftarrow -\mathbf{u}_3$  switches the sign of  $\det(U)$  while leaving invariant the represented fundamental matrix. This concludes the proof.

*Implementation details.* Through the iterations,  $\sigma$  may go outside of its boundaries. This is not a problem since the corresponding motion is still valid.

There are several possibilities to ensure the boundaries of  $\sigma$  such as using linear constraints. I propose to enforce these boundaries at each iteration while leaving the current estimate invariant. However, I have found during my experiments of §§5 and 6 that in practice, this does not affect the behaviour of the underlying optimization process. A way to proceed is:

- if  $\sigma < 0$  then  $\sigma \leftarrow -\sigma$ ,  $\mathbf{u}_2 \leftarrow -\mathbf{u}_2$ ,  $\mathbf{u}_3 \leftarrow -\mathbf{u}_3$ ;
- if  $\sigma > 1$  then  $\sigma \leftarrow \frac{1}{\sigma}$ ,  $\text{swap}(\mathbf{u}_1, \mathbf{u}_2)$ ,  $\mathbf{u}_3 \leftarrow -\mathbf{u}_3$ ,  $\text{swap}(\mathbf{v}_1, \mathbf{v}_2)$ ,  $\mathbf{v}_3 \leftarrow -\mathbf{v}_3$ .

One can easily check that these changes on the orthonormal representation leave the represented motion invariant.

### 3 Bundle Adjustment

In this section, I show how the orthonormal representation can be used for bundle adjustment of point features seen in two views. This is summarized in table 1. Similar results could be derived for other criteria, such as the minimization of the distances between points and epipolar lines or the gradient-weighted criterion [10,18].

**Table 1.** Implementing my minimal estimator within the bundle adjustment Levenberg-Marquardt-based framework given in [8], p.574 (algorithm A4.1). Note that  $r$  is the number of residuals and that the second projection matrix have to be extracted from the orthonormal representation using equation (4) (for e.g. computing the error vector).

**Two-view projective bundle adjustment** expressed within the framework of [8], p.574 (algorithm A4.1). The initial guess of the fundamental matrix is denoted by  $F_0$ .

*Add the following steps:*

- (i') Initialize the orthonormal representation  $(U, V, \sigma)$  by a scaled singular value decomposition of  $F_0$ :

$$F_0 \sim U \cdot \text{diag}(1, \sigma, 0) \cdot V^T.$$

- (ii') Turn the full  $(r \times 12)$  camera Jacobian matrix  $A = \bar{A}$  into the minimal  $(r \times 7)$  Jacobian matrix of the orthonormal representation:

$$A \leftarrow A \cdot A^{\text{ortho}},$$

where  $A^{\text{ortho}}$  is given by equations (5,6);

*Change the parameter update step as:*

- (viii) Update the orthonormal representation as:

$$U \leftarrow U \cdot R(\mathbf{x}) \quad V \leftarrow V \cdot R(\mathbf{y}) \quad \sigma \leftarrow \sigma + \delta_\sigma,$$

where  $\delta_a^T = (\mathbf{x}^T \mathbf{y}^T \delta_\sigma)$  are the 7 motion update parameters, update the structure parameters by adding the incremental vector  $\delta_b$  and compute the new error vector;

*Add the last step:*

- (xi) Return the computed  $F$  using equation (3) as:

$$F \sim \mathbf{u}_1 \mathbf{v}_1^T + \sigma \mathbf{u}_2 \mathbf{v}_2^T.$$

**Cost function.** Bundle adjustment consists in solving the following optimization problem, see e.g. [12,17,18]:

$$\min_{\mathbf{a}, \mathbf{b}} \mathbf{r}^T \mathbf{r},$$

where:

- $\mathbf{a}$  and  $\mathbf{b}$  are respectively motion and structure parameters (or parameters used to update them);
- $\mathbf{r}$  is the vector of residual errors;
- $r_i^2 = d^2(\mathbf{q}_i, P\mathbf{Q}_i) + d^2(\mathbf{q}'_i, P'\mathbf{Q}_i)$  is the  $i$ -th point residual error ( $d$  is the 2D Euclidean distance) corresponding to its reprojection error;
- $\mathbf{q}_i$  and  $\mathbf{q}'_i$  are corresponding image points for the first and second images;
- $\mathbf{Q}_i$  are 3D reconstructed points and depend upon  $\mathbf{b}$ ;

- $P$  and  $P'$  are projection matrices corresponding to the current motion estimate represented by  $\mathbf{a}$ . They must correspond to a realization of the fundamental matrix. I have shown in §2.3, equation (4), how the canonic realization can be directly obtained from the orthonormal representation.

*Analytical differentiation.* Newton-type optimization methods, such as the widely used Levenberg-Marquardt, necessitate computing the Jacobian matrix  $J = (A \mid B)$  of the residual vector  $\mathbf{r}$  with respect to motion and structure parameters  $\mathbf{a}$  and  $\mathbf{b}$ . While this can be achieved numerically using e.g. finite differences [14], it may be better to use an analytical form for both computational efficiency and numerical accuracy. I focus on the computation of  $A = \frac{\partial \mathbf{r}}{\partial \mathbf{a}}$  since  $B = \frac{\partial \mathbf{r}}{\partial \mathbf{b}}$  only depends upon structure parameterization. I decompose it as  $A_{(r \times 7)} = \tilde{A}_{(r \times 12)} \cdot A_{(12 \times 7)}^{\text{ortho}}$  where:

- $r$  is the number of residuals;
- only the 12 entries of  $P'$  are considered since  $P$  is fixed in the canonic reconstruction basis (1);
- $\tilde{A} = \frac{\partial \mathbf{r}}{\partial \mathbf{p}'}$  ( $\mathbf{p}' = \text{vect}(P')$  where  $\text{vect}(\cdot)$  is the row-wise vectorization) depends on the chosen realization of the fundamental matrices, i.e. on the coordinate frame employed. I have chosen the canonic projection matrices (1). This Jacobian matrix is employed directly for the overparameterization of [8]. Deriving its analytical form is straightforward;
- $A^{\text{ortho}} = \frac{\partial \mathbf{p}'}{\partial \mathbf{a}}$  is related to the orthonormal motion representation.

I therefore concentrate on deriving a closed-form expression for  $A^{\text{ortho}}$ . If the minimal method of e.g. [18] were used, 36 different Jacobian matrices corresponding to each parameterization would have to be derived.

One of the advantages of my update scheme shown in table 1 and based on the orthonormal representation is that there exists a simple closed-form expression for  $A^{\text{ortho}}$ .

Let us consider the orthonormal representation  $(U, V, \sigma)$ . In this case, the motion update parameters are minimal and defined by  $\mathbf{a} = (x_1 \ x_2 \ x_3 \ y_1 \ y_2 \ y_3 \ \sigma)^T$ , where  $\mathbf{x} = (x_1 \ x_2 \ x_3)^T$  and  $\mathbf{y} = (y_1 \ y_2 \ y_3)^T$  are used to update  $U$  and  $V$  respectively. Since  $U$  and  $V$  are updated with respect to the current estimate,  $A^{\text{ortho}}$  is evaluated at  $(U, V, \sigma)$ , i.e. at  $\mathbf{a} = \mathbf{a}_0 = (\mathbf{0}_{(6 \times 1)} \ \sigma)^T$ . Let  $\tilde{U} = U \cdot R(\mathbf{x})$  and  $\tilde{V} = V \cdot R(\mathbf{y})$  be the updated  $U$  and  $V$ . Equation (4) is used to derive a closed-form expression of the second canonic projection matrix after updating, i.e. corresponding to the orthonormal representation  $(\tilde{U}, \tilde{V}, \sigma)$ . By expanding, differentiating and evaluating this expression at  $\mathbf{a}_0$ , I obtain:

$$A^{\text{ortho}} = \frac{\partial \mathbf{p}'}{\partial \mathbf{a}} = \frac{\partial \mathbf{p}'}{\partial (x_1 \ \dots \ y_3 \ \sigma)} = \left( \left( \frac{\partial \mathbf{p}'}{\partial x_1} \right) \ \dots \ \left( \frac{\partial \mathbf{p}'}{\partial y_3} \right) \ \left( \frac{\partial \mathbf{p}'}{\partial \sigma} \right) \right), \quad (5)$$

where:

$$\frac{\partial \mathbf{p}'}{\partial \mathbf{a}} = \begin{cases} \text{vect}(\mathbf{u}_3 \mathbf{v}_1^T \mid -\mathbf{u}_2) \cdot \partial x_1 \\ \text{vect}(-\sigma \mathbf{u}_1 \mathbf{v}_3^T \mid \mathbf{0}) \cdot \partial x_2 \\ \text{vect}(\sigma \mathbf{u}_3 \mathbf{v}_2^T \mid \mathbf{u}_1) \cdot \partial x_3 \\ \text{vect}(-\mathbf{u}_2 \mathbf{v}_3^T \mid \mathbf{0}) \cdot \partial y_1 \\ \text{vect}(-\mathbf{u}_1 \mathbf{v}_1^T - \sigma \mathbf{u}_2 \mathbf{v}_2^T \mid \mathbf{0}) \cdot \partial y_2 \\ \text{vect}(\mathbf{u}_2 \mathbf{v}_2^T + \sigma \mathbf{u}_1 \mathbf{v}_1^T \mid \mathbf{0}) \cdot \partial y_3 \\ \text{vect}(-\mathbf{u}_1 \mathbf{v}_2^T \mid \mathbf{0}) \cdot \partial \sigma. \end{cases} \quad (6)$$

## 4 The Multiple-View Case

In this section, I extend my projective two-view motion modelisation to multiple views. I analyse how to model additional views. I propose the orthonormal representation of projective multiple-view motion. As in the two-view case, this serves to devise elements for optimizing projective multiple-view motion over minimal parameters.

### 4.1 Modeling Additional Views

Once two views have been modeled, the coordinate frame is fixed. Therefore, an additional view does not have any free gauge and its complete projection matrix has to be modeled. Let  $P$  be such a  $(3 \times 4)$  projection matrix. Since it is homogeneous, it has 11 degrees of freedom. This can be seen in several other ways. For example, one can consider the metric decomposition  $P \sim K(R \ t)$ . However, this decomposition is not available here since I deal with uncalibrated cameras. One may also interpret  $P \sim (H \ \alpha e)$ , where  $H$  is a  $(3 \times 3)$ -matrix, as a plane homography  $H$  with respect to the reference view, i.e. thus for which the projection matrix is  $(I_{(3 \times 3)} \ 0_{(3 \times 1)})$ ,  $e$  a 3-vector that represents an epipole with the same view and  $\alpha$  a scalar that accounts for the relative scale between  $H$  and  $e$ . Therefore,  $P$  has  $8 + 2 + 1 = 11$  degrees of freedom. This interpretation is related to that of plane+parallax, see e.g. [9].

### 4.2 Relation to Previous Work

A common strategy for optimizing an homogeneous entity such as  $P$  is to overparameterize it by using all its entries. A normalization constraint is then softly imposed by using an hallucinated measurement, e.g. on the norm of  $P$  as  $\|P\|^2 - 1 = 0$ . The drawback of this method is that more parameters than necessary are estimated, which increases the computational cost of the estimation process and may cause numerical instabilities. One could also renormalize  $P$  after each iteration as  $P \leftarrow P/\|P\|^2$ . Alternatively, one could fix one entry of  $P$  to a given value, e.g. 1, but this representation would have singularities.

The main drawback of these techniques is that a unique minimal parameterization does not suffice to express all cases. This leads to the necessity for multiple expressions of e.g. the Jacobian matrix for Newton-type optimizers, which might complexify implementation issues.

### 4.3 The Orthonormal Representation

The orthonormal representation of  $P \sim (H \ \alpha e)$  can be derived as follows. Let  $s = \alpha e$ . This inhomogeneous 3-vector is a scaled version of  $e$  which has 3 degrees of freedom since it also encapsulates the relative scale  $\alpha$  between  $H$  and  $e$ . Therefore,  $s \in \mathbb{R}^3$  and it can be directly parameterized by its 3 elements.

Consider now the homogeneous  $(3 \times 3)$ -matrix  $H$ . As in the case of the fundamental matrix, see §2.3, I examine its singular value decomposition  $H \sim U \Sigma V^T$  where  $U$  and  $V$  are  $(3 \times 3)$  orthonormal matrices and  $\Sigma$  a diagonal one containing the singular values of  $H$ . Since  $H$  may be singular, see e.g. in §2.1 the canonic plane homography of [11], but

must not be rank-1 or null,  $\Sigma \sim \text{diag}(1, \sigma_1, \sigma_2)$ , where  $0 < \sigma_1 \leq 1$  and  $0 \leq \sigma_2 \leq \sigma_1$ . Therefore, the orthonormal representation of  $H$  writes as:

$$(U, V, \sigma_1, \sigma_2) \in \mathbb{H} \text{ where } \mathbb{H} \equiv \mathbb{F} \times \{\sigma_2 \mid 0 \leq \sigma_2 \leq \sigma_1\},$$

and  $F \equiv O(3)^2 \times \{\sigma_1 \mid 0 < \sigma_1 \leq 1\}$ , see equation (2). As a byproduct, one can observe that I have derived the orthonormal representation  $\mathbb{H}$  of 2D homographies which can be used as a starting point to devise minimal 8-parameter estimators for these transformations. Finally, I obtain the orthonormal representation of  $P$ , denoted by  $\mathbb{P}$  as:

$$(U, V, \sigma_1, \sigma_2, s) \in \mathbb{P} \text{ where } \mathbb{P} \equiv \mathbb{H} \times \mathbb{R}^3.$$

It is minimal in that it has  $3 + 3 + 1 + 1 + 3 = 11$  degrees of freedom.

#### 4.4 Optimization With Minimal Parameters

Mimicking the method of §2.4 for the projective two-view motion case, I obtain a way to minimally estimate projective multiple-view motion. Given a set of camera matrices, I represent two of them using the orthonormal projective two-view motion of §2.3. This fixes the coordinate frame. Each other view is then modeled by the orthonormal representation  $(U, V, \sigma_1, \sigma_2, s) \in \mathbb{P}$  described above. Each  $O(3)$  matrix  $U, V$  can be updated using minimal parameters as e.g.  $U \leftarrow U \cdot R(x)$  whereas  $\sigma_1, \sigma_2$  and  $s$  are directly optimized. As in the two-view case, one can derive algorithms to ensure the boundaries on  $\sigma_1$  and  $\sigma_2$ . A closed-form solution for the Jacobian matrix of the residuals with respect to the motion parameters can be derived in a manner similar to the two-view case. Another advantage of this representation is that one can directly compute the inverse of  $H$ , the  $(3 \times 3)$  leading part of  $P$ , from its orthonormal representation. This may be useful for e.g. projecting 3D lines or estimating 2D homographies using a symmetric cost function.

### 5 Experimental Results Using Simulated Data

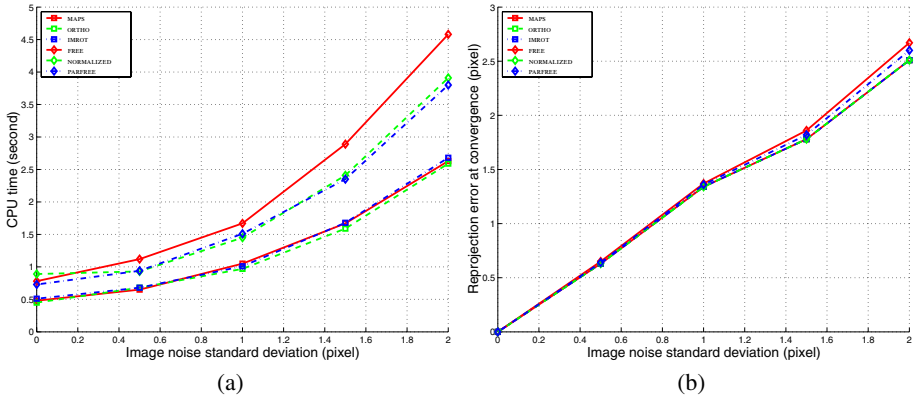
In this section, I compare the algorithm using the orthonormal motion representation (see table 1 and §§2.4 and 4.4) to existing ones using simulated data. I use the Levenberg-Marquardt method to perform optimization. Points are minimally parameterized as in [8], p.579. The test bench consists of 50 points lying inside a sphere with a radius of 1 meter observed by cameras with a focal length of 1000 (expressed in number of pixels). Each of these cameras is looking at the center of the sphere and is situated at a distance of 10 meters from it. The baseline between two consecutive cameras is 1 meter.

Points are generated in 3D space, projected onto the images and corrupted by an additive centered gaussian noise with varying standard deviation.

I measure the two quantities characteristic of a bundle adjustment process, computational cost, i.e. CPU time to convergence and the error at convergence, versus the standard deviation of added image noise. I also measure the error of the current estimate as a function of time through the optimization processes. The plots correspond to median values over 300 trials. The bundle adjustments are initialized by the values obtained using the

8 point algorithm [4] and the triangulation method described in [5] for two views. Each other view is then registered in turn by linearly computing its camera matrix. I compare the following algorithms:

- ORTHO: uses the minimal methods given in table 1 and §§2.4 and 4.4;
- MAPS and IMROT (Image Rotation): are other minimal methods given in [1], the associated research report. These methods are equivalent to those described in [18, 19] in the sense that the number of unknowns is minimal;
- FREE: uses an overparameterization with free gauges, namely all the entries of the camera matrices are optimized;
- NORMALIZED: uses an overparameterization plus hallucinated measurements to prevent the gauge to drift [12];
- PARFREE (Partially Free): uses a partially free gauge by completely parameterizing all camera matrices except the first one [7];



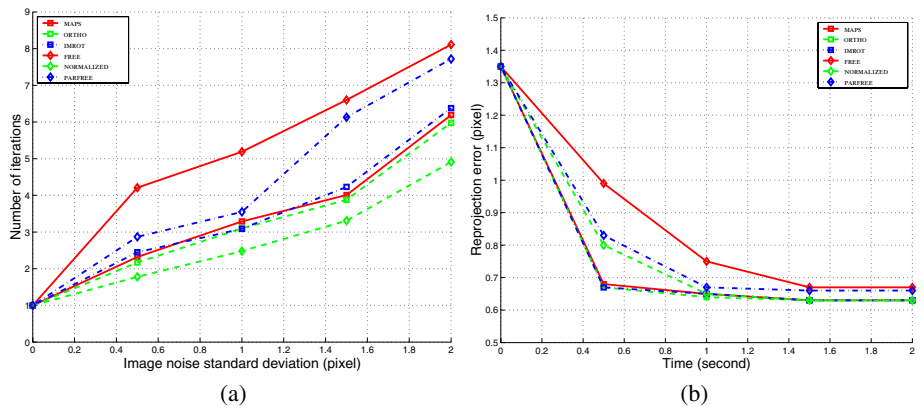
**Fig. 1.** Comparison of (a): the CPU time to convergence and (b): value of the error at convergence versus varying image noise for different methods.

I conduct a serie of experiments using two cameras. One can observe on figure 1 (b) that, roughly speaking, all methods converge to the same minimum of the cost function. Methods that have a slightly less reliable convergence than the others are FREE and PARFREE.

Figure 1 (a) shows that, for roughly the same convergence properties, there are quite big discrepancies between the computational cost of each method. The method that has the highest computational cost is FREE, followed by NORMALIZED and PARFREE. This can be explained by the fact that these methods have more unknowns to estimate than the minimal ones. This requires more computational time for each iteration to be performed. Finally, methods using the minimal number of parameters, MAPS, ORTHO and IMROT have the lowest computational cost, roughly the same.



In the light of these results, it is clear that methods using minimal parameters should be preferred for both computational cost and convergence properties. The method ORTHO, relying on the orthonormal representation given in this paper has the advantage of simplicity. However, in order to understand and explain the behaviour of the different methods, I have measured the number of iterations and the computational cost of these iterations. These results are shown on figure 2.



**Fig. 2.** Comparison of (a): the number of iterations to convergence versus varying image noise and (b): the evolution of reprojection errors.

**Table 2.** CPU time per iteration (second) for each method.

MAPS	ORTHO	IMROT	FREE	NORMALIZED	PARFREE
0.3841	0.3838	0.3864	0.4741	0.7043	0.4717

In more detail, I have found that methods FREE or PARFREE, leaving the gauge drift freely have very bad convergence properties, performing more iterations, roughly twice more, than the others, see figure 2 (a). Method NORMALIZED performs a number of iterations smaller than all the other methods but involves solving a much more costly linear system at each iteration, see table 2. Methods using the minimal number of parameters are trade-offs between the number of iterations and their computational cost: each iteration has a low computational cost and the number of iterations needed is in-between those of free gauge methods and NORMALIZED. This explains why these methods achieve the lowest total computational cost.

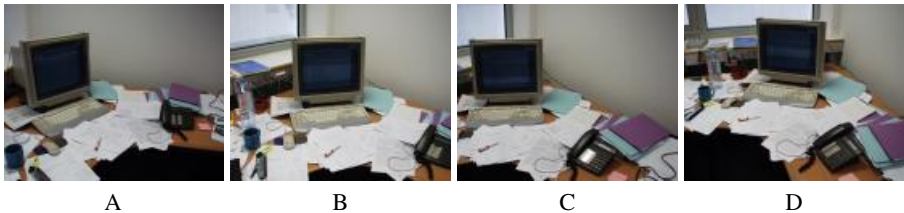
Figure 2 (b) shows the evolution of reprojection error for the different optimization processes. This experiment is useful in the sense that the time to convergence previously measured is highly dependent on how convergence is determined, e.g. by thresholding

two consecutive errors, and does not account for the ability of the algorithms to quickly, or not, reach almost the value of convergence. This experiment has been conducted using the same test bench as previously with a noise level on image point positions of 0.5 pixel. One can see on this figure that methods based on a minimal parameterization reach their value of convergence before the others. The `NORMALIZED` and `PARFREE` methods take roughly twice the same time, while the `FREE` method takes three times more.

Finally, I conduct experiments using 10 views. I observe that the differences between the algorithms observed in the two-view case are decreased while those requiring the lowest computation time are the same, i.e. `MAPS`, `ORTHO` and `IMROT`. Other experiments on the convergence properties of the algorithms in the multiple-view case yield conclusions similar to the two-view case.

## 6 Experiments on Real Images

In this section, I validate my algorithms using real images. I first consider the case of two images. In order to cover all possibilities for the epipoles to be close to the images or at infinity, I use pairs of the images shown on figure 3. Initial values for structure and motion are computed as in the case of simulated data.



**Fig. 3.** Real images used to validate the algorithms.

Results are shown in table 3. For each combination of image pair and each algorithm, I estimate the CPU time to convergence  $\mathcal{T}$  and the error at convergence  $\mathcal{E}$ . The last row of the table show the mean values  $\bar{\mathcal{T}}$  and  $\bar{\mathcal{E}}$  of  $\mathcal{T}$  and  $\mathcal{E}$  for each algorithm over the set of image pairs. These results confirmed those obtained using simulated data.

I have also tested the algorithms on all four images of figure 3. Initial values have been obtained by registering each view to an initial guess of structure and motion obtained from the two first ones. The results are the followings: all algorithms converge with a final error of 0.73 pixels and their relative performances in terms of computation times to convergence were equivalent to those obtained in the case of two views.

## 7 Conclusions

I studied the problem of optimizing projective motion over minimal sets of parameters. I proposed the orthonormal representation of projective two-view motion. I showed

**Table 3.** Error at convergence  $\mathcal{E}$  and time to convergence  $\mathcal{T}$  obtained when combining pairs of images from figure 3.

epipoles		views	MAPS		ORTHO		IMROT		FREE		NORMALIZED		PARFREE	
e	e'		$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$
$\infty$	$\infty$	A, B	0.63	2.45	0.63	2.39	0.63	2.47	0.68	3.98	0.63	2.99	0.68	3.02
		A, C	0.71	2.38	0.71	2.41	0.71	2.40	0.77	4.01	0.71	3.56	0.71	3.71
$\infty$	$\infty$	A, D	0.45	2.03	0.45	1.76	0.45	2.19	0.57	3.13	0.45	3.09	0.45	2.93
$\infty$	$\infty$	B, C	0.88	3.53	0.88	3.39	0.88	3.55	1.23	6.70	0.88	5.12	0.88	4.63
$\infty$	$\infty$	B, D	0.59	2.33	0.59	2.10	0.59	2.81	0.59	3.99	0.59	3.41	0.59	3.56
		C, B	0.51	1.91	0.51	1.92	0.51	2.02	0.51	3.39	0.51	2.79	0.51	3.04
average $\mathcal{E}$ and $\mathcal{T}$			0.628	2.430	0.628	2.328	0.628	2.573	0.725	4.200	0.628	3.493	0.637	3.482

how this can be used to locally update projective two-view motion using a minimal set of 7 parameters. The canonic projection matrices can be directly extracted from the orthonormal representation. I extent this representation to projective multiple-view motion. As a byproduct, I derive the orthonormal representation of 2D homographies. The method can be plugged into most of the (possibly sparse) non-linear optimizers such as Levenberg-Marquardt. I gave a closed-form expression for the Jacobian matrix of the residuals with respect to the motion parameters, necessary for Newton-type optimization techniques.

The introduced orthonormal representation seems to be a powerful tool for minimal optimization of homogeneous entities in particular.

I conducted experiments on simulated and real data. My conclusions are that methods based on minimal parameter sets perform better than the others, in terms of computational cost while achieving equivalent results in terms of convergence properties. The most interesting results are obtained in the two-view case. Existing algorithms that do not constrain the gauge by any means perform worse than the others.

I will make an implementation of the method available on my home-page.

In future work, I plan to investigate the use of the orthonormal representation introduced in this paper to model other algebraic entities and devise minimal estimation techniques for them.

## References

1. A. Bartoli and P. Sturm. Three new algorithms for projective bundle adjustment with minimum parameters. Research Report 4236, INRIA, Grenoble, France, August 2001.
2. A. Bartoli, P. Sturm, and R. Horaud. Projective structure and motion from two views of a piecewise planar scene. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 593–598, July 2001.
3. P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 683–695. Springer-Verlag, April 1996.

4. R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1064–1070, June 1995.
5. R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
6. R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceeding of the DARPA-ESPRIT workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 187–202, October 1993.
7. R.I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, October 1994.
8. R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.
9. M. Irani and P. Anadan. Parallax geometry of pairs of points for 3d scene analysis. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, pages 17–30. Springer-Verlag, 1996.
10. Q.T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.
11. Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.
12. P. F. McLauchlan. Gauge invariance in projective 3D reconstruction. In *Proceedings of the Multi-View Workshop, Fort Collins, Colorado, USA*, 1999.
13. J. Oliensis. The error surface for structure and motion. Technical report, NEC, 2001.
14. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.
15. C.C. Slama, editor. *Manual of Photogrammetry, Fourth Edition*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.
16. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 709–720. Springer-Verlag, April 1996.
17. B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.
18. Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.
19. Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *Computer Vision and Image Understanding*, 82(2):174–180, May 2001.

# Structure from Many Perspective Images with Occlusions

Daniel Martinec and Tomáš Pajdla\*

Center for Machine Perception  
Department of Cybernetics  
Czech Technical University in Prague  
Karlovo nám. 13, 121 35 Praha, Czech Republic  
{martid1, pajdla}@cmp.felk.cvut.cz

**Abstract.** This paper proposes a method for recovery of projective shape and motion from multiple images by factorization of a matrix containing the images of all scene points. Compared to previous methods, this method can handle perspective views and occlusions jointly. The projective depths of image points are estimated by the method of Sturm & Triggs [11] using epipolar geometry. Occlusions are solved by the extension of the method by Jacobs [8] for filling of missing data. This extension can exploit the geometry of perspective camera so that both points with known and unknown projective depths are used. Many ways of combining the two methods exist, and therefore several of them have been examined and the one with the best results is presented. The new method gives accurate results in practical situations, as demonstrated here with a series of experiments on laboratory and outdoor image sets. It becomes clear that the method is particularly suited for wide base-line multiple view stereo.

**Keywords:** projective reconstruction, structure from motion, wide base-line stereo, factorization

## 1 Introduction

In the past geometric and algebraic relations among uncalibrated views up to four in number have been described [5]. Various algorithms for scene reconstruction with both orthographic and perspective camera have been proposed [5,12,8,11,6,9,13,3,10]. The reconstruction problem from orthographic camera is conceptually satisfactorily solved but this could not be claimed for the case of a perspective camera. The biggest problem that remained to be solved was dealing consistently with scene occlusions.

---

\* This research was supported by the grants GACR 102/00/1679, MSMT KONTAKT 2001/09 and ME412, and MSM 212300013. Andrew Zisserman from the University of Oxford kindly provided the Dinosaur data, Marc Pollefeys from K.U.Leuven the Temple data, and Tomáš Werner from the University of Oxford provided the routine for the bundle adjustment.

**Table 1.** Comparison of some 3D reconstruction methods. Lexicographical ordering was used so that (i) the importance of a criterion decreases from the first to the last column and (ii) the quality of the method decreases from top to down

Algorithm	views	camera	occlusions	privileged data	depends on im. ordering
<b>the new algorithm</b>	<b>N</b>	<b>persp.</b>	<b>yes</b>	<b>no</b>	<b>no</b>
Fitzgibbon & Zisserman [3]	N	persp.	yes	no	yes
Avidan & Shashua [10]	N	persp.	yes	no	yes
Urban et al. [13]	N	persp.	yes	central view	no
Heyden [6]	N	persp.	no	no	no
Mahamud & Hebert [9]	N	persp.	no	weak persp.	no
Sturm & Triggs [11]	N	persp.	no	no	yes
Jacobs [8]	N	orthog.	yes	no	no
Tomasi & Kanade [12]	N	orthog.	yes	initial submatrix	no
Hartley & Zisserman [5]	2,3,4	orthog. persp.	no	no	no

This paper offers a linear method which extends and suitably combines previous methods so that the reconstruction in an entirely general situation, i.e. many images with perspective camera and occlusions, is possible. A review of previous works follows.

The situation is similar for two, three, and four uncalibrated images. 3D structure of a scene can be recovered up to an unknown projective transformation, where the camera geometry can be represented by the fundamental matrix, the trifocal, and the quadrifocal tensor respectively [5].

For any number of images, image coordinates of the projections of 3D points can be combined into a so called *measurement matrix*. Tomasi and Kanade [12] developed a factorization method of the measurement matrix for scene reconstruction with an orthographic camera and Sturm and Triggs [11] extended this method from affine to perspective projections. Heyden's method [6] uses a different approach. It relies only on subspace methods instead of multilinear constraints. Similarly, Mahamud & Hebert proposed a method [9], which computes projective depths iteratively but it can only be used for weak perspective or for full perspective with a good initial depth estimate.

Occlusions present a significant problem for reconstruction. The above mentioned Tomasi and Kanade's method solves this problem under the orthographic projection but the result depends on the choice of some initial submatrix of the measurement matrix. The method is iterative and errors may increase gradually with the number of iterations. Jacobs' method [8] improves the above approach so that no initial submatrix is needed. He combines constraints on the reconstruction derived from small submatrices of the full measurement matrix. It treats all data uniformly and is independent of image ordering.

Under perspective projection, the occlusion problem has not yet been generally solved. Method [13] by Urban et al. is dependent on the choice of a central image, which is combined with other images in a so called “cake” configuration. Only points whose projections are contained in the central image can be reconstructed. Method [3] by Fitzgibbon & Zisserman and [10] by Avidan & Shashua compute reconstruction from a sequence of images using trifocal tensors and fundamental matrices. Subsequent images are taken one after another and used to extend and improve actual reconstruction. Table 1 summarizes the differences among the mentioned methods.

Jacobs [8] solves reconstruction with occlusions for orthographic camera, Sturm & Triggs [11] solve reconstruction without occlusions for perspective camera. We present a novel method that builds on these two methods so that scene reconstruction from many perspective images with occlusions is obtained. Our method is independent of image ordering and treats all data uniformly up to a certain level of missing data. Above this level, the reconstruction process may prefer some data depending on the method of computing the projective depths.

The paper is organized as follows. The reconstruction problem is formulated in Section 2. In Section 3.1 and 3.2, algorithms [11] and [8] are reviewed, respectively. In 3.3, the new filling algorithm is presented. In 3.4, the new reconstruction method is proposed. Experiments with artificial and real data are presented in sections 5 and 6. Section 7 gives suggestions for future work.

## 2 Problem Formulation

Suppose a set of  $n$  3D points and that some of them are visible in  $m$  perspective images. The goal is to recover 3D structure (point locations) and motion (camera locations) from the image measurements. This recovery will be called *scene reconstruction*. No camera calibration or additional 3D information will be assumed, so it will be possible to reconstruct the scene up to a projective transformation of the 3D space.

Let  $\mathbf{X}_p$  be the unknown homogeneous coordinate vectors of the 3D points,  $\mathbf{P}^i$  the unknown  $3 \times 4$  projection matrices, and  $\mathbf{x}_p^i$  the measured homogeneous coordinate vectors of the image points, where  $i = 1, \dots, m$  labels images and  $p = 1, \dots, n$  labels points. Due to occlusions,  $\mathbf{x}_p^i$  are unknown for some  $i$  and  $p$ .

The basic image projection equation says that  $\mathbf{x}_p^i$  are the projections of  $\mathbf{X}_p$  up to unknown scale factors  $\lambda_p^i$ , which will be called (*projective*) *depths*:

$$\lambda_p^i \mathbf{x}_p^i = \mathbf{P}^i \mathbf{X}_p$$

The complete set of image projections can be gathered into a matrix equation:

$$\underbrace{\begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \dots & \lambda_n^1 \mathbf{x}_n^1 \\ \times & \lambda_2^2 \mathbf{x}_2^2 & \dots & \times \\ \vdots & & \ddots & \vdots \\ \lambda_1^m \mathbf{x}_1^m & \times & \dots & \lambda_n^m \mathbf{x}_n^m \end{bmatrix}}_{\mathbf{R}} = \underbrace{\begin{bmatrix} \mathbf{P}^1 \\ \vdots \\ \mathbf{P}^m \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_n \end{bmatrix}}_{\mathbf{X}}$$

where marks  $\times$  stand for unknown elements which could not be measured due to occlusions,  $\mathbf{X}$  and  $\mathbf{P}$  stand for structure and motion, respectively. The  $3m \times n$  matrix  $[\mathbf{x}_p^i]_{i=1..m, p=1..n}$  will be called the *measurement matrix* whereas  $\mathbf{R}$  will be called the *partially rescaled measurement matrix*, shortly PRMM, because  $\mathbf{R}$  will be used even with some unknown depths. Both measurement matrix and PRMM may have (and in most cases do have) some missing elements.

### 3 The Main Idea of the New Reconstruction Algorithm

A complete rescaled measurement matrix has rank four and therefore a projective reconstruction can be obtained by its factorization. However, from measurements in perspective images with occlusions, we can only compose a measurement matrix which is neither complete nor rescaled. When it is at all possible to compute projective depths of some known points in  $\mathbf{R}$ , e.g. via multi-view constraints, some missing elements of  $\mathbf{R}$  can often be filled using the knowledge that every five columns of complete rescaled  $\mathbf{R}$  are linearly dependent.

It would be ideal to first compute the projective depths of all known points in  $\mathbf{R}$  and then to fill all the missing elements of  $\mathbf{R}$  by finding a complete matrix of rank four that would be equal (or as close as possible) to the rescaled  $\mathbf{R}$  in all elements where  $\mathbf{R}$  is known. Such a two-step algorithm is almost the ideal linearized reconstruction algorithm, which uses all data and has a good statistical behavior. We have found that many image sets, in particular those resulting from wide base-line stereo, can be reconstructed in such two steps.

Of course, there are image sets, e.g. sets with the structure of missing data on the borderline of reconstructibility or long sequences with very factionalized tracks, which cannot be solved in the above two steps. Instead, the two steps have to be repeated while the measurement matrix  $\mathbf{R}$  is not complete. If the correspondences between the images are such that the measurement matrix is large and diagonally dominant, then it is possible to use another reconstruction technique, e.g. to fuse partial consecutive reconstructions [3,10]. However, if there is no clear sequence of images or central image like in [13], the proposed algorithm has a clear advantage. It can handle arbitrary scenes in pseudo-optimal manner without a priori preferring any particular image. It provides a unique solution and thus is suited for the initialization of bundle adjustment optimizations.

In what follows, we shall describe the two steps of the algorithm. Let us first review the two steps we build on and their respective extensions. Later we will describe how to combine the two steps.

#### 3.1 Estimating the Projective Depths

Many works dealt with estimating the projective depths. In this work, we used Sturm & Triggs' method [11] exploiting epipolar geometry but other methods, e.g. [6,9,5], can be applied also. The method [11] was proposed in two alternatives. The alternative with a central image is more appropriate for wide base-line stereo while the alternative with a sequence is more appropriate for video-sequences. The former will be denoted as  $\omega_{cent,c}$  where  $c$  denotes the number



of a central image while the latter will be denoted as  $\omega_{seq}$ . Thus, we have altogether the totality  $\Omega = \{\omega_{seq}, \omega_{cent,1} \dots \omega_{cent,m}\}$  of alternatives for computing the projective depths. Also, the method from [11] has to be furthermore slightly modified on account of missing data. The complete algorithm is summarized in Algorithm 1. The  $p$ -th track there denotes a subsequence of known points in sequence  $\mathbf{x}_p^1 \dots \mathbf{x}_p^m$ .

1. Set  $\lambda_p^j = 1$  for all  $p$  corresponding to known points  $\mathbf{x}_p^j$  in view  $j = \begin{cases} 1 : & \text{for } \omega_{seq} \\ c : & \text{for } \omega_{cent,c} \end{cases}$
2. For  $\begin{cases} j = 1 \dots m-1, i = j+1 : & \text{for } \omega_{seq} \\ j = c, i \neq j : & \text{for } \omega_{cent,c} \end{cases}$  do the following. If images  $i$  and  $j$  have enough points in common to compute a fundamental matrix uniquely<sup>a</sup> then compute their fundamental matrix  $\mathbf{F}^{ij}$ , epipole  $\mathbf{e}^{ij}$ , and depths  $\lambda_p^i$  according to

$$\lambda_p^i = \frac{(\mathbf{e}^{ij} \wedge \mathbf{x}_p^i) \cdot (\mathbf{F}^{ij} \mathbf{x}_p^j)}{\|\mathbf{e}^{ij} \wedge \mathbf{x}_p^i\|^2} \lambda_p^j$$

if the right side of the equation is defined, where  $\wedge$  stands for the cross-product.

For  $\omega_{seq}$ : if the  $p$ -th track ( $p = 1 \dots n$ ) is discontinuous, start with  $j = b(p)$  where  $b(p)$  denotes the initial image of the longest continuous subtrack of the  $p$ -th track.

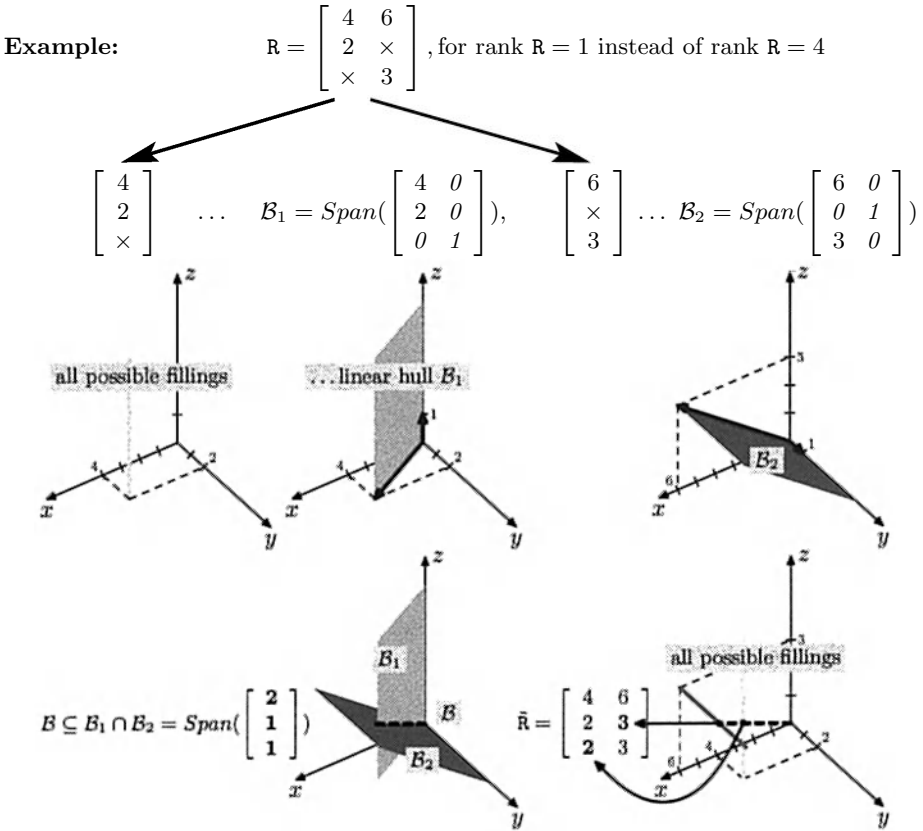
<sup>a</sup> See Section 3.4.

**Algorithm: 1.** Estimating the depths: alternatives  $\omega_{seq}$  and  $\omega_{cent,c}$

### 3.2 Filling of Missing Elements in $\mathbf{R}$

Filling of missing data was first realized by Tomasi & Kanade [12] for orthographic camera. D. Jacobs [8] improved their method and we use our extension of his method for the perspective case. Often, not all depths can be computed because of missing data. Therefore, we extend the method from [8] so that also points with unknown depths are exploited. Moreover, the extension is independent of how depths are estimated and thus any method for estimating the depths could be used. Before describing our modification for the perspective camera, the original Jacobs' algorithm for the orthographic case has to be explained.

D. Jacobs treated the problem of missing elements in a matrix as fitting an unknown matrix of a certain rank to an incomplete noisy matrix resulting from measurements in images. Assume noiseless measurements for a while to make the explanation simpler. Assuming perspective images, an unknown complete  $3m \times n$  matrix  $\tilde{\mathbf{R}}$  of rank 4 is fitted to PRMM  $\mathbf{R}$ . Technically, a basis of the linear vector space that is spanned by the columns of  $\tilde{\mathbf{R}}$  is searched for. Thus, when there are 4 complete linearly independent columns in  $\mathbf{R}$ , then they form the desired basis. When no such 4-tuple of columns exists, the basis has to be constructed from incomplete columns. Fortunately, some 4-tuples of incomplete columns provide constraints on the basis and a sufficient number of such constraints determine it.



**Fig. 1.** Forming constraints on the basis and filling the matrix. For  $R$  is of rank 1, constraints on  $\mathcal{B}$  are formed by single columns

Let us explain what we mean by saying that an incomplete column  $c$  of  $R$  spans (generates) a subspace. Every complete column of  $R$  generates a one-dimensional subspace of  $\mathbb{R}^{3m}$ . Thus, an incomplete  $c$  generates a subspace  $V$ , as the smallest linear space containing all one-dimensional subspaces generated by  $c$  after replacing unknown elements by some arbitrary real numbers. Linear subspaces form a complete lattice [2] and therefore such smallest linear space  $V$  exists. It is a subspace of  $\mathbb{R}^{3m}$  and equals the linear hull of all one-dimensional subspaces. The generators of  $V$  can be obtained by constructing the column containing the known elements of  $c$  and zeros instead of the unknown ones and augmenting it with the standard basis spanning the dimensions of the unknown elements (see Fig. 1 and the example in Section 3.3).

Let the space generated by the columns of  $\tilde{R}$  be denoted by  $\mathcal{B}$ . Let  $\mathcal{B}_t$  denotes the span of the  $t$ -th 4-tuple of columns of  $R$  which are linearly independent in coordinates known in all four columns.  $\mathcal{B}$  is included in each  $\mathcal{B}_t$  and thus also

in their intersection i.e.  $\mathcal{B} \subseteq \bigcap_{t \in T} \mathcal{B}_t$ , where  $T$  is some set of indices. When the intersection is 4D,  $\mathcal{B}$  is known exactly. If it is of a higher dimension, only an upper bound on  $\mathcal{B}$  is known and more constraints from 4-tuples must be added. Any column in  $\tilde{\mathbf{R}}$  is a linear combination of vectors of a basis of  $\tilde{\mathbf{R}}$ . Thus, having a basis  $\mathbf{B}$  of  $\tilde{\mathbf{R}}$ , any<sup>1</sup> incomplete column  $c$  in  $\mathbf{R}$  can be completed by finding the vector  $\tilde{c}$  generated by  $\mathbf{B}$  which equals  $c$  in the elements where  $c$  was known in  $\mathbf{R}$  (see Fig. 1).

Linear independency of the 4-tuple of columns is crucial to obtain a valid constraint on the basis. Consider, e.g., a 4-tuple consisting of four equal columns, thus spanning only a 1D space. Even if three coordinates in one of its columns are made unknown, and thus a 4D space is spanned,  $\mathcal{B}$  does not have to be included in the span. A row with some missing coordinates can be ignored because the entire corresponding dimension is spanned and the constraint on  $\mathcal{B}$  is always satisfied in the dimension, meaning such a row contains no information. This is the reason to use just the 4-tuples of columns linearly independent in coordinates known in all four columns.

Because of noise in real data, the intersection  $\bigcap_{t \in T} \mathcal{B}_t$  quickly becomes empty. This is why  $\mathcal{B}$  is searched for as the closest 4D space to spaces  $\mathcal{B}_t$  in the sense of the minimal sum of square differences of known elements. Denoting complement of a linear vector space by  $\perp$ ,  $\bigcap_{t \in T} \mathcal{B}_t$  can be expressed according to the well known De Morgan rule as  $(\text{Span}_{t \in T} \mathcal{B}_t^\perp)^\perp$ . The generators of  $\mathcal{B}_t^\perp$  can be found as  $\mathbf{B}_t^\perp = \mathbf{u}(:, d+1 : \text{end})$ , where  $[\mathbf{u}, \mathbf{s}, \mathbf{v}] = \text{svd}(\mathbf{B}_t)$  and  $d$  is the dimension of  $\mathcal{B}_t$ .  $\text{Span}_{t \in T} \mathcal{B}_t^\perp$ , where  $T$  is of cardinality  $z$ , is generated by  $[\mathbf{B}_1^\perp \mathbf{B}_2^\perp \dots \mathbf{B}_z^\perp]$ .  $(\text{Span}_{t \in T} \mathcal{B}_t^\perp)^\perp$  is generated by  $\mathbf{u}(:, \text{end} - 3 : \text{end})$ , where  $[\mathbf{u}, \mathbf{s}, \mathbf{v}] = \text{svd}([\mathbf{B}_1^\perp \mathbf{B}_2^\perp \dots \mathbf{B}_z^\perp])$ .

### 3.3 Filling of Missing Elements for Perspective Cameras

Jacobs' method [8] cannot use image points with unknown depths. But, PRMM constructed from measurements in perspective images often has many such points where the corresponding depths cannot be computed. Therefore, we extended the method to exploit also points with unknown depths. It brings two advantages: (i) because the actual iteration of the two-step algorithm exploits more information, the number of iterations may decrease and consequently more accurate results may be obtained; (ii) it is possible to reconstruct more scene configurations. See Section 8 in [1] for more details about this. It is important that the proposed extension is still a linear method as was the Jacobs' method [8].

Let us first explain the extension for two images. Suppose that  $\lambda_p^i$  and  $\mathbf{x}_p^i$  are known for  $i = 1, 2$ , and for  $p = 1 \dots 4$  except  $\lambda_4^2$ . Then, consider the first four columns of  $\mathbf{R}$  to be the  $t$ -th 4-tuple of columns,  $\mathbf{A}_t$ . A new matrix  $\mathbf{B}_t$ , whose span will be denoted by  $\mathcal{B}_t$ , can be defined using known elements of  $\mathbf{A}_t$  as

$$\mathbf{A}_t = \begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_4^1 \mathbf{x}_4^1 \\ \lambda_1^2 \mathbf{x}_1^2 & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & ? \mathbf{x}_4^2 \end{bmatrix} \longrightarrow \mathbf{B}_t = \begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_4^1 \mathbf{x}_4^1 & 0 \\ \lambda_1^2 \mathbf{x}_1^2 & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & 0 & \mathbf{x}_4^2 \end{bmatrix}$$

<sup>1</sup> containing at least four known elements, which in practice means six elements resulting from two known points

It can be proved (see Corollary 1 in Appendix A in [1]) that if  $B_t$  is of full rank (i.e. five here) then  $\mathcal{B} \subseteq \text{Span}(B_t)$ , which is exactly the constraint on  $\mathcal{B}$ .

In a general situation there are also some missing elements in  $R$ . Then, the matrix  $B_t$  is constructed from the  $t$ -th 4-tuple  $A_t$  of columns of  $R$  as follows:

1. Set  $B_t$  to  $A_t$ .
2. Replace all unknown points and points with unknown depth by zero in  $B_t$ .
3. For each unknown depth  $\lambda_p^i$  in  $A_t$ , add a column with  $\mathbf{x}_p^i$  and zeros everywhere else to  $B_t$ .
4. For each triple of rows in  $A_t$  containing some unknown point, add to  $B_t$  the standard basis spanning the dimensions of the unknown point.

The following example demonstrates the construction of  $B_t$  from a 4-tuple  $A_t$ :

$$\begin{aligned}
 A_t &= \begin{bmatrix} ? & \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_4^1 \mathbf{x}_4^1 \\ \lambda_1^2 \mathbf{x}_1^2 & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & \lambda_4^2 \mathbf{x}_4^2 \\ \lambda_1^3 \mathbf{x}_1^3 & \lambda_2^3 \mathbf{x}_2^3 & \lambda_3^3 \mathbf{x}_3^3 & \times \end{bmatrix} \xrightarrow{2} \begin{bmatrix} \mathbf{0} & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_4^1 \mathbf{x}_4^1 \\ \lambda_1^2 \mathbf{x}_1^2 & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & \lambda_4^2 \mathbf{x}_4^2 \\ \lambda_1^3 \mathbf{x}_1^3 & \lambda_2^3 \mathbf{x}_2^3 & \lambda_3^3 \mathbf{x}_3^3 & \mathbf{0} \end{bmatrix} \\
 &\quad \downarrow 3 \\
 B_t &= \begin{bmatrix} \mathbf{0} & \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_2^1 \mathbf{x}_2^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \lambda_1^2 \mathbf{x}_1^2 & \mathbf{0} & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & \lambda_4^2 \mathbf{x}_4^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \lambda_1^3 \mathbf{x}_1^3 & \mathbf{0} & \lambda_2^3 \mathbf{x}_2^3 & \lambda_3^3 \mathbf{x}_3^3 & \mathbf{0} & \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix} \xleftarrow{4} \begin{bmatrix} \mathbf{0} & \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \lambda_3^1 \mathbf{x}_3^1 & \lambda_4^1 \mathbf{x}_4^1 \\ \lambda_1^2 \mathbf{x}_1^2 & \mathbf{0} & \lambda_2^2 \mathbf{x}_2^2 & \lambda_3^2 \mathbf{x}_3^2 & \lambda_4^2 \mathbf{x}_4^2 \\ \lambda_1^3 \mathbf{x}_1^3 & \mathbf{0} & \lambda_2^3 \mathbf{x}_2^3 & \lambda_3^3 \mathbf{x}_3^3 & \mathbf{0} \end{bmatrix}
 \end{aligned}$$

If  $B_t$  is of full rank, its span  $\mathcal{B}_t$  includes  $\mathcal{B}$  (this can be proved by induction from Corollary 1 in Appendix A in [1]). By including also image points with unknown projective depths the spaces  $\mathcal{B}_t$  spanned by 4-tuples of columns become smaller, thus solving the complete problem becomes more efficient.

It can be seen that the concept of generating constraints on the basis for the orthographic case is only a special case of generating constraints for the perspective case. The former is equivalent to the latter having all depths set to the same number thus corresponding to the perspective camera with the projection center at infinity and looking at a finite scene.

### 3.4 Combining the Filling Method with Estimating the Depths

Due to occlusions, the computation of projective depths can be carried out in various ways depending on which depths are computed first and if and how those already computed are used to compute the others. One way of depth estimation will be called a *strategy*. Depending on the chosen strategy, different subsets of depths are computed and different submatrices of PRMM are filled. It may happen when some strategy exploiting e.g. epipolar geometry of some pair of images is used that the fundamental matrix cannot be computed due to occlusions. Consequently, depths needed to form a constraint on the basis of PRMM in one of the images cannot be estimated, thus the missing data in the image cannot be filled and the two steps of depth estimation and filling has to be repeated.

For accurate data, all strategies should be equivalent. It is not so if the data is noisy. In such case, the task is to choose the strategy which results in the smallest

error. It would be unrealistically costly to compute all possibilities (although there is “only” a finite number of them) and to choose the best one. Fortunately, we do not have to compute all of them in order to find some good one. From the structure of missing data, it is possible to predict a good strategy for depth estimation that results in a good reconstruction. Some criterion deciding which strategy is good is needed. For scenes reconstructible in more steps, such criterion also determines which subset of depths is better to be computed first.

The following two observations have been made. First, the more iterations are performed, the less accurate results are obtained because the error from the former iteration spreads in subsequent iterations as was also mentioned in [8]. Secondly, unknown elements should not be computed from fewer data when they can be computed from more data, and thus more accurately due to the law of big numbers and supposition of random noise. Both these observations support the following.

**Principle 1** *The more image points that are filled in one step, the smaller the expected error.*

This principle leads to a pseudo-optimal number of iterations that need to be performed.<sup>2</sup> Practically, however, it is not crucial problem that such obtained strategy is only pseudo-optimal because, as will be seen later, it is possible to realize Principle 1 so that, for many scenes, only one iteration is performed. The following proposition holds.

**Proposition 1** *The more depths known before the filling, the smaller the expected error.*

Proof of Proposition 1 inheres in our extension of Jacob’s method (see Appendix B in [1]). Usage of Principle 1 and Proposition 1 in order of their designation proved to be a good criterion. We choose the set of strategies which fill the most points, and from this set, we choose those which scale the most points. From the resulting set, an arbitrary strategy can be used.

The criterion will now be described formally. Let  $\omega$  denote some strategy for estimating the depths and  $\Omega$  denote some set of strategies. Let  $\mathcal{F}(\omega)$  denote the predicted number of newly filled unknown image points during one iteration when  $\omega$  is used. The strategy, for which  $\mathcal{F}(\omega)$  is maximal, is the best strategy according to Principle 1. More such strategies often exist. Let  $\mathcal{S}(\omega)$  denote the predicted number of estimated depths when  $\omega$  is used. According to Proposition 1,  $\mathcal{S}(\omega)$  is maximal for the best strategy. The complete new method for scene reconstruction is summarized in Algorithm 2.

The usefulness of the concept of predictor functions  $\mathcal{F}, \mathcal{S} : \Omega \rightarrow 0..mn$  consists in their ability to be evaluated without neither estimating the depths

<sup>2</sup> An optimal strategy would have to be searched for as the shortest branch in the tree graph of all partial solutions. Partial solutions can be ordered into a tree graph. Edges in this graph correspond to chosen strategies and vertices correspond to the partial solutions obtained after one iteration. The root of the tree corresponds to the initial PRMM.

1. Estimate depths using an arbitrary strategy  $\omega^* \in \Omega^*$  where

$$\Omega_{\mathcal{F}} = \left\{ \omega \in \Omega \mid \mathcal{F}(\omega) = \max_{\tau \in \Omega} \mathcal{F}(\tau) \right\}$$

$$\Omega^* = \left\{ \omega \in \Omega_{\mathcal{F}} \mid \mathcal{S}(\omega) = \max_{\tau \in \Omega_{\mathcal{F}}} \mathcal{S}(\tau) \right\}$$

2. Fill the missing data.

Repeat steps 1. and 2. until  $\mathbf{R}$  is complete or no data can be filled in. Then factorize a maximal complete submatrix of  $\mathbf{R}$ .

**Algorithm: 2.** Estimating the depths: alternatives  $\omega_{seq}$  and  $\omega_{cent,c}$

nor data filling. The knowledge of which image points are known or unknown is the only information for the evaluation of  $\mathcal{F}$  and  $\mathcal{S}$ . It is very simple (and fast) but it cannot detect degenerate configurations of points because, in fact, the multi-view tensors are not computed. If it then, when the tensor is computed, turns out that the configuration is degenerate, the second best strategy is used, etc.

To define  $\mathcal{F}$  and  $\mathcal{S}$ , a few symbols have to be introduced. Let logical variable  $x_p^i$  be true if and only if the image point  $\mathbf{x}_p^i$  is known. Let  $i$  and  $j$  be as in step 2 of Algorithm 1. Let  $\mathcal{I}^{ij}$  be true if and only if the data of image  $i$  can be used by the filling method consistently with other images [11]. It is only possible if  $i = j$  or if images  $i$  and  $j$  have enough (at least seven) points in common, which are necessary to compute a fundamental matrix uniquely, thus

$$\mathcal{I}^{ij} \equiv |\{p \mid x_p^i \wedge x_p^j\}| \geq 7 \quad \vee \quad i = j \quad (1)$$

The uniqueness is demanded for the depths consistency with other images. All available points are used for the fundamental matrix estimation. (i) If there are only 7 points, the 7-point algorithm [5] is performed. If it provides three real solutions, the fundamental matrix is not unique. (ii) If there are 8 points or more, the 8-point algorithm [5] is performed. In this case, degenerate configurations can easily be detected.

The predictor functions depend on the way how projective depths are computed. Let us first define the predictor functions for the alternative  $\omega_{cent,c}$  when the depths are computed using a central image  $c$ . Let  $\mathcal{P}_p^c$  be true if and only if the  $p$ -th 3D point can be filled in by the filling method when depths were estimated using strategy  $\omega_{cent,c}$ . To recover a 3D point uniquely from known basis of PRMM, at least two its images are needed. Moreover, it can be proved (see Theorem 4 in Appendix A in [1]) that at least two known depths in each image are needed for the constraints on  $\mathcal{B}$ . It means that  $\mathcal{P}_p^c$  is true if and only if the  $p$ -th 3D point is seen in at least 2 images and the corresponding fundamental matrices, which are needed for estimating at least some two depths in the images, can be computed:

$$\mathcal{P}_p^c \equiv |\{i \mid \mathcal{I}^{ic} \wedge x_p^i\}| \geq 2 \quad (2)$$

Now, predictor functions  $\mathcal{F}$  and  $\mathcal{S}$  can be defined as follows

$$\begin{aligned}\mathcal{F}(\omega_{cent,c}) &= |\{ \langle i, p \rangle \mid \mathcal{I}^{ic} \wedge \mathcal{P}_p^c \wedge \neg x_p^i \}| \\ \mathcal{S}(\omega_{cent,c}) &= |\{ \langle i, p \rangle \mid \mathcal{I}^{ic} \wedge \mathcal{P}_p^c \wedge x_p^i \wedge x_p^c \}| \end{aligned}$$

Term  $\mathcal{I}^{ic} \wedge \mathcal{P}_p^c$  says whether point  $\mathbf{x}_p^i$  can be reconstructed.

Similarly, the predictor functions for alternative  $\omega_{seq}$  when the depths are computed for a sequence are defined as

$$\begin{aligned}\mathcal{P}_p &\equiv |\{i \mid x_p^i\}| \geq 2 \\ \mathcal{F}(\omega_{seq}) &= |\{ \langle i, p \rangle \mid \mathcal{P}_p \wedge \neg x_p^i \}| \\ \mathcal{S}(\omega_{seq}) &= \sum_{p \in 1..n} \max_{k \in b(p)..m} \arg \bigwedge_{i \in b(p)..k} x_p^i \end{aligned} \quad (3)$$

Eq. (3) simply says that the points in the longest continuous subtracks have known depths (See Algorithm 1).

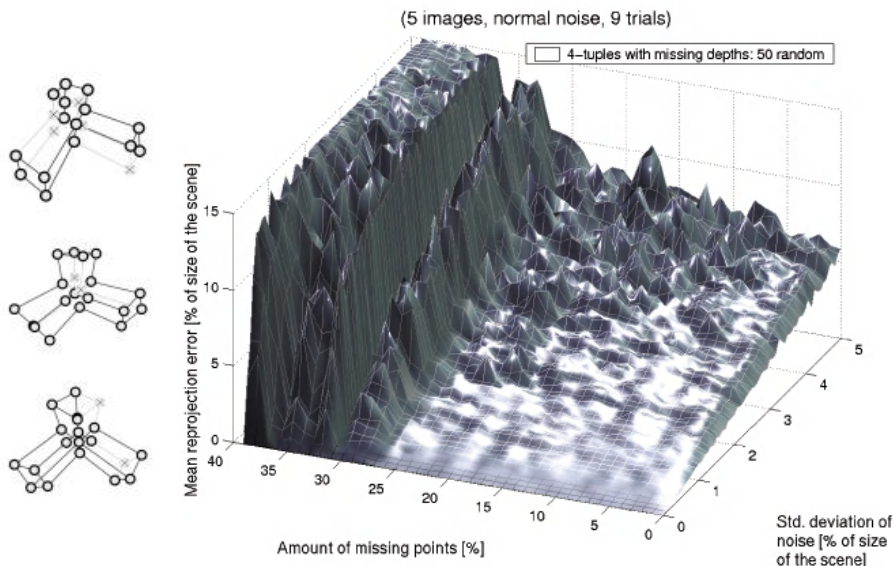
## 4 Implementation Details

On account of good numerical conditioning, several normalizations of the data and balancing similar to those in [11] need to be performed. Choosing of 4-tuples of columns is implemented so that almost each chosen 4-tuple gives the constraint on the basis of PRMM. This is aimed so that columns are chosen one after another. The columns, which cannot provide the constraint with already chosen ones, are temporarily removed from PRMM until the next 4-tuple is chosen. By this way, a good efficiency is achieved.

## 5 Experiments with Artificial Scenes

For experiments with artificial scenes, a simulated scene with cubes was used. The scene models a real scene, hence it represents a generic situation. Twenty points in space were projected by perspective cameras into several images from different locations and directions. Some image points were made unknown to simulate scene occlusions, see the left-hand side of Experiment 1.

Points were taken out from the scene randomly but in a uniform fashion so that, first, the numbers of missing points in each image differed maximally by one, and secondly, the numbers of images of each point differed maximally by one. Points were only removed as long as the whole scene could still be reconstructed. The necessary condition for a complete reconstruction is that each image contains at least 7 points and each point has at least 2 images (see (1) and (2)). The more data available, the higher the percentage of missing data permissible. For this specific experiment, i.e. 20 points in 5 images, 65 % of missing data is the upper bound allowable to get a complete reconstruction. But because of randomly spread holes in data, the actual level of the maximum amount of missing data



Experiment 1: Dependency of reprojection error on noise and missing data

for the complete reconstruction is lower. Experiment 1 shows the dependency of the reprojection error of the reconstruction using Alg. 2 on noise and missing data. Along the left horizontal axis, the amount of the missing data grows while along the right horizontal axis, standard deviation of Gaussian noise of zero mean value added to image points increases. The standard deviation of the added noise as well as the reprojection error is displayed in percentage of the scene size.

If no noise is present, the reconstruction is precise. The reprojection error grows linearly with noise with slope approximately equal one and is almost constant in the direction of missing points up to the level of missing data above which the reconstruction fails. To conclude, the new algorithm is accurate and robust with respect to noise as well as missing data.

## 6 Experiments with Real Scenes

For each experiment, one image, an error table, and the structure of PRMM are provided. The correspondences across the images have been detected either manually or by the Harris interest operator [4]. Besides the scene name and point detection, the table includes the chosen strategy for estimating the depths, the amount of missing data, the number of images used, image sizes, the number of known points in each image, and reprojection errors for our method Algorithm 2 and bundle adjustment initialized by the output of our method. The structure of PRMM shows the exploitation of image points with known ("•") and unknown ("o") projective depths. Empty places stand for unknown points. All scenes have been reconstructed in one iteration of Algorithm 2.



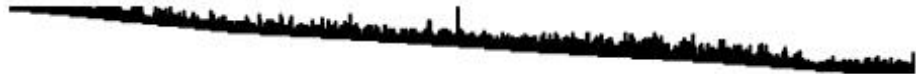
Method	LM = linear method, BA = bundle adj.									
	Scene name	<i>House</i>								
	Point detection	manual								
	Depth estimation	$\omega_{cent,1}$								
	Amount of missing data	<b>47.83 %</b>								
LM	Mean error per image point [pxl]	<b>3.91</b>								
LM + BA		<b>1.44</b>								
	Image No. [2952×2003]	1	2	3	4	5	6	7	8	85 10
	Number of corresp.	116	112	857	112	851	785	130	126	101 855
LM	Maximal error [pxl]	11.0	36.6	12.1	85.3	25.8	15.5	13.6	8.9	14.7 13.4
LM + BA		4.3	6.6	4.5	4.4	5.8	8.3	7.5	6.3	10.7 10.1
LM	Mean error	2.3	6.8	3.2	2.3	8.1	5.0	2.5	2.3	3.3 4.8
LM + BA		1.1	1.8	1.5	1.2	1.5	1.6	1.2	1.4	1.5 1.8



size = 10 × 203, " " missing (47.83 %), "•" scaled (75.7 %), "o" not scaled (24.3 %)


Experiment 2: House

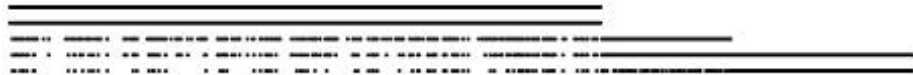
Method	LM = linear method, BA = bundle adj.													
	Scene name	<i>Dinosaur (Oxford)</i>												
	Point detection	Harris' operator												
	Depth estimation	$\omega_{seq}$												
	Amount of missing data	<b>90.84 %</b>												
LM	Mean error per image point [pxl]	<b>1.76</b>												
LM + BA		<b>0.64</b>												
	Image No. [720×576]	1	5	9	13	17	21	25	29	33	36			
	Number of corresp.	257	318	322	516	535	568	602	459	464	381			
LM	Maximal error [pxl]	18.4	16.3	29.5	56.4	46.9	73.9	44.1	28.5	19.4	33.9			
LM + BA		10.9	12.7	7.8	41.5	25.7	13.1	13.4	17.3	17.9	21.4			
LM	Mean error	0.6	0.7	2.3	2.0	3.8	1.7	1.4	1.6	1.3	1.0			
LM + BA		0.3	0.5	0.6	1.0	1.0	0.4	0.3	0.5	0.9	0.7			



size = 36 × 4983, " " missing (90.84 %), "•" scaled (100.0 %)

Experiment 3: Dinosaur (Oxford)

Method	LM = linear method, BA = bundle adj.					
	Scene name	<i>Temple (Leuven)</i>				
	Point detection	Harris' operator				
	Depth estimation	$\omega_{seq}$				
Amount of missing data					<b>46.32 %</b>	
LM	Mean error per image point [pxl]				<b>0.49</b>	
LM + BA					<b>0.23</b>	
	Image No. [867×591]	1	2	3	4	5
	Number of corresp.	456	456	297	374	285
LM	Maximal error [pxl]	3.0	2.3	2.8	2.5	3.0
LM + BA		2.5	1.5	2.4	1.8	2.5
LM	Mean error	0.4	0.5	0.6	0.5	0.5
LM + BA		0.3	0.2	0.2	0.2	0.2



size = 5 × 696, " " missing (46.32 %), "●" scaled (100.0 %)

Experiment 4: Temple (Leuven)

The “House” scene (see Experiment 2) was captured on 10 images at high resolution. Approximately 100 points were manually detected in each image. Although 47.83 % data was missing, the reprojection error, given in pixels, is low considering the image sizes. It can be seen that our algorithm could have exploited all known data including 24.3 % unscaled points.

The “Dinosaur” scene (see Experiment 3) was captured on 36 images. Points were detected automatically by the Harris operator. Although the amount of missing data is high (90.84 %), the mean error per image point was lower because of more precise point detection and since 100 % of points were scaled.

The data in Experiment 4 contained outliers, that were removed one after another in the following manner. The scene was first reconstructed with all the data including outliers. Then, the column of PRMM, which contained the point with the highest reprojection error, was discarded. Afterwards, the scene was again reconstructed, another column discarded etc. These two steps were repeated till the highest reprojection error was significant. For the “Temple” scene in Experiment 4, the threshold was set to 4 pixels which lead to discarding 23 out of 719 columns.

To conclude, the new algorithm is enough accurate on real scenes to provide a good initial solution for bundle adjustment.

## 7 Summary and Conclusions

A new linear method for scene reconstruction has been proposed and tested on artificial and real scenes. The method extends and suitably combines previous

methods so that the reconstruction in an entirely general situation, i.e. many images with perspective camera and occlusions, is possible.

A new way of exploiting points with unknown depth was developed. Correctness of this way was proved as well as its abilities and limitations were studied in [1]. Its theoretical asset is the ability to reconstruct linearly some very small scene configurations, which can be reconstructed by other methods only non-linearly (see Theorem 3 in [1]), cannot be reconstructed at all (see Theorem 2 in [1]), or cannot exploit all known data (see Theorem 1 in [1]). Moreover, it gives good results in practical situations as presented here.

The proposed method was intended to deal with several problems in 3D reconstruction. These were the perspective projection, many images, and occlusion. However, one problem was not taken into account explicitly and that is the problem of outliers in correspondences. Although the method was not intended to deal with outliers, it was observed that it can deal with them if they are few compared to the number of inliers (see Experiment 4). To deal well with a bigger amount of outliers, extension [7] of factorization handling outliers can be added.

## References

1. D. Martinec and T. Pajdla. Structure from Many Perspective Images with Occlusions. Research Report CTU-CMP-2001-20, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, July 2001. <ftp://cmp.felk.cvut.cz/pub/cmp/articles/martinec/Martinec-TR-2001-20.pdf>.
2. M. K. Bennett. *Affine and Projective Geometry*. John Wiley and Sons, New York, USA, 1995.
3. A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, pages 311–326. Springer-Verlag, June 1998.
4. C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of Alvey Vision Conference*, pages 147–151, 1988.
5. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
6. A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Proc. 10th SCIA*, pages 963–968, June 1997.
7. D. Q. Huynh and A. Heyden. Outlier Detection in Video Sequences under Affine Projection. In *Proc. of CVPR*, 2001.
8. D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *CVPR*, pages 206–212, 1997.
9. S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *CVPR*, 2000.
10. S. Avidan and A. Shashua. Threading Fundamental Matrices. In *IEEE Trans. on PAMI*, Vol. 23(1), pp. 73–77, 2001.
11. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV96(II)*, pages 709–720, 1996.
12. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In *IJCV(9)*, No. 2, pages 137–154, November 1992.
13. M. Urban, T. Pajdla, and V. Hlaváč. Projective reconstruction from N views having one view in common. In *Vision Algorithms: Theory & Practice*. Springer LNCS 1883, pages 116–131, September 1999.

# Sequence-to-Sequence Self Calibration

Lior Wolf and Assaf Zomet

School of Computer Science and Engineering,  
The Hebrew University,  
Jerusalem 91904, Israel  
`{liwolf,zomet}@cs.huji.ac.il`

**Abstract.** We present a linear method for self-calibration of a moving rig when no correspondences are available between the cameras. Such a scenario occurs, for example, when the cameras have different viewing angles, different zoom factors or different spectral ranges. It is assumed that during the motion of the rig, the relative viewing angle between the cameras remains fixed and is known. Except for the fixed relative viewing angle, any of the internal parameters and any of the other external parameters of the cameras may vary freely. The calibration is done by linearly computing multilinear invariants, expressing the relations between the optical axes of the cameras during the motion. A solution is then extracted from these invariants. Given the affine calibration, the metric calibration is known to be achieved linearly (e.g. by assuming zero skew). Thus an automatic solution is presented for self calibration of a class of moving rigs with varying internal parameters. This solution is achieved without using any correspondences between the cameras, and requires only solving linear equations.

**Keywords:** Self-Calibration, Multi-View Invariants.

## 1 Introduction

The projective framework of Structure from Motion (SFM) is supported by a relatively large body of literature on the techniques for taking matching image features (points and lines) across multiple views and producing a projective representation of the three-dimensional (3D) positions of the features in space.

For many tasks such as computer graphics, projective representation is not sufficient, and an Affine/Metric representation is required. In recent years there has been much progress in the theory and algorithms of Self/Auto-Calibration, the upgrading of a projective structure to an Affine/Metric one, without using knowledge of the viewed scene. These algorithms must make assumptions about the parameters of the camera to achieve a unique solution (For a review see [6]).

There are roughly two approaches to self-calibration: The non-stratified approach solves directly for metric upgrade. This can be done for example by solving Kruppa's equations [4], or by solving for the absolute quadric [11]. In both cases the solution of a set of non-linear equations is required. The stratified approach first upgrades the projective representation to an affine representation by solving for the homographies through the plane at infinity and then upgrades

the affine representation to a metric one [3,8]. It was pointed by several authors (e.g. [6,7]) that the first stage, the affine calibration, is the most challenging in stratified methods.

By using a stereo rig, a stable solution for both affine and metric calibration can be computed linearly from two or more images from each camera ([7], following [12]). This method assumes that the rig structure and the Zoom/Focus are fixed. Another approach for rig calibration achieves a linear solution by assuming known relative orientations of the cameras, but allowing the Zoom/Focus and the relative displacements to vary [13]. Both methods require correspondences between the cameras.

In this paper we focus on a self-calibration scenario which is of interest on both practical and theoretical fronts. We consider a rig of two or more video cameras, each capturing an image sequence, where there is no or little spatial overlap between the fields of view of the cameras (such as when the cameras have different zoom settings or pointing to largely different directions). While in motion, each camera may change its internal parameters and zoom factor.

Applying conventional self calibration algorithms for each camera separately is always challenging with varying internal parameters, especially the affine calibration stage. Applying one of the existing rig algorithms [7,13], on the other hand, is not possible since no correspondences are available between the cameras.

Therefore, we seek to solve for the calibration of the cameras, while exploiting the *rigidity constraints between them*. We refer to this problem as "Sequence to Sequence Self-Calibration", since every sequence requires a different calibration, and only the relations between the sequences can be exploited to recover these calibrations. A similar scenario, of exploiting the rigidity between multiple cameras, was presented recently in the context of image alignment [1].

A previous approach for handling self calibration without correspondences between the cameras [2] assumed highly constrained conditions: In addition to knowing one degree of freedom about the rotation between the cameras, that solution also assumed fixed internal parameters, where only two of which were unknown. It also required the solution of highly non-linear equations over a large set of variables. The work presented in this paper solves for affine calibration linearly, without using any assumption but one known d.o.f. of the relative rotation between the cameras.

In order to solve for the calibration, we use assumptions about the relative orientations of the cameras in the rig. We divide the problem to three different cases:

- Two cameras with parallel image planes. The cameras may view the same direction, or the opposite directions (Back to back).
- Two cameras with non-parallel image planes, with the special case of orthogonal image planes.
- Cameras with varying orientations, each rotating about its  $Y$  axis. Their  $Y$  axes are assumed to be parallel.

In the following sections we present a linear solution for affine self-calibration for each of the cases above, assuming the cameras are synchronized. Later in

Section. 5 we show how such synchronization can be achieved automatically. The calibration is done by computing multilinear tensorial invariants, expressing the relations between the axes of the cameras during the motion. A solution is then extracted from these tensors. Having solved for the challenging stage of affine calibration, the metric calibration can be achieved linearly by imposing further constraints, e.g. zero skew (see [6]). In case the cameras view the same/opposite directions, they share the same affine ambiguity. Thus metric constraints on both cameras can be simultaneously used for calibrating both of the cameras.

### 1.1 Formal Statement of the Problem

A pinhole camera projects a point  $P$  in 3-D projective space  $\mathcal{P}^3$  to a point  $p$  in the 2-D projective plane  $\mathcal{P}^2$ . The projection can be written as a  $3 \times 4$  homogeneous matrix  $M$ :

$$p \cong MP$$

where  $\cong$  marks equality up to a scale factor. When the camera is calibrated, it can be factored (by QR decomposition):

$$M = K[R; T]$$

where  $R$  and  $T$  are the rotation and translation of the camera respectively, and  $K$  is a  $3 \times 3$  upper diagonal matrix containing the internal parameters of the camera. The most general form of the internal parameters matrix  $K$  is:

$$K = \begin{bmatrix} f & \gamma & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $f$  is the *focal length*,  $\alpha$  is the *aspect ratio*,  $(u_0, v_0)$  is the *principle point* and  $\gamma$  is the *skew*. It is practical to model  $K$  by a reduced set of internal parameters, for example assume zero skew.

Generally, given projections of  $m$  3-D points  $\{P_j\}_{j=1}^m$  to  $n$  images, it is possible to estimate the location of the 3-D points and the camera matrices  $\{M\}_{i=1}^n$  up to a projective transformation (collineation) represented by a  $4 \times 4$  matrix  $H$ :

$$p \cong MH^{-1}HP \quad (2)$$

The matrices  $\{C_i = M_i H^{-1}\}_{i=1}^n$  are called the projective camera matrices. The points  $HP$  are the points in the projective coordinate system. We refer to the matrix  $H^{-1}$  as the projective to Euclidean matrix.

When the internal parameters of the cameras  $\{K_i\}_{i=1}^n$  are known, then  $H$  can be recovered up to a 3-D similarity transformation. The goal of (internal) calibration is to recover  $\{K_i\}_{i=1}^n$ , or equivalently recover the  $4 \times 4$  collineation  $H$  up to a similarity transformation.

In all following sections it is assumed that the projective camera matrices  $\{C\}_{i=1}^n$  were already computed for each one of the sequences using for example point correspondences between the different frames. This can be done in various ways (for a review see [6]).

## 2 Parallel Optical Axes

We first analyze the case of two cameras attached in a rig, viewing the same direction, or the opposite directions (back-to-back). In addition, each camera may rotate freely about its  $Z$  axis. Note that there are no constraining assumption about neither the internal parameters of the cameras, nor the relative displacements of the centers of projections.

The first Euclidean camera matrix is given by:

$$M_i^1 = K_i^1[R_i; t_i^1]$$

and the second Euclidean camera is given by:

$$M_i^2 = K_i^2[R_0 * R_i; t_i^2]$$

Where  $R_0$  is the relative rotation between the cameras, and the third row of  $R_0$  equals to  $[0, 0, 1]$  up to sign.

Given the projective cameras  $C_i^1, C_i^2$ , we seek for two projective transformations  $H_1, H_2$  mapping the projective cameras to the Euclidean ones (up to similarity transformation).

Let  $r_i$  be the third row of  $R_i$ , and let  $c_i^j$  be the third row of  $C_i^j$ . Let  $\hat{H}_j, j = 1..2$  be the  $4 \times 3$  matrices composed from the first 3 columns of the matrices  $H_j$ .

The internal parameters matrices  $K_i^j$  are upper triangular matrices and so  $\hat{H}_1^T c_i^1 \cong r_i$ . Having also the special structure of the third row of  $R_0 : \hat{H}_2^T c_i^2 \cong r_i$ . Thus for every  $i$ :

$$\hat{H}_1^T c_i^1 \cong \hat{H}_2^T c_i^2 \cong r_i \quad (3)$$

The constraint above holds also for  $D\hat{H}_1, D\hat{H}_2$  for every  $3 \times 3$  matrix  $D$ . This results in an ambiguity that would later on express itself as an affine ambiguity on the cameras, the same affine ambiguity for both cameras (up to similarity).

The above Eqns. 3 have a form we call “equivalence after projection”. In the next section we present the solution for such equations.

Given the solution to these equations, the homographies through the plane at infinity can be recovered for each sequence and between the sequences. A point  $[a^T, 0]^T$  on the plane at infinity is projected to  $C_i^j \hat{H}_j a$  in the  $i$ th frame of the  $j$ th camera. Hence the homography at infinity  $H_{ijkl}^\infty$  between frame  $i$  of camera  $j$  and frame  $k$  of camera  $l$  is given by

$$H_{ijkl}^\infty = C_k^l \hat{H}_l (C_i^j \hat{H}_j)^{-1}$$

The solutions to Eqn. 3 are up to a common  $3 \times 3$  matrix  $D$  which cancels out, and so:

$$H_{ijkl}^\infty = C_k^l \hat{H}_l (C_i^j \hat{H}_j)^{-1} = C_k^l \hat{H}_l D (C_i^j \hat{H}_j D)^{-1}$$

Similarly, the coordinates of the plane at infinity  $L_j$  in the projective representations of camera  $j$  can be recovered. Since  $L_j$  satisfies  $L_j^\top \hat{H}_j a = 0$  for every  $a$ , then  $L_j$  is the null space of  $\hat{H}_j^\top$ , which is also the null space of  $D\hat{H}_j^\top$ .

## 2.1 The “Equivalence after Projection” Problem

We define the equivalence after projection problem as follows: Given two sets of points in  $P^3$   $\{P_i\}_{i=1}^n, \{Q_i\}_{i=1}^n$ , determine whether there exist two  $3 \times 4$  projection matrices  $A, B$  such that for every  $i$ :

$$AP_i \cong BQ_i$$

If such  $A$  and  $B$  exist solve for them up to a multiplication on the left by a  $3 \times 3$  matrix. We next show a solution for this problem which requires only solving linear systems of equations.

Let  $C_B$  be the center of the projection matrix  $B$ , i.e  $C_B$  is the null space of  $B$ . Let  $B^+$  be a pseudo-inverse of  $B$ , i.e. a mapping from the image plane of  $B$  to a plane in  $P^3$  such that  $BB^+p \cong p$ .

The points  $C_B, B^+AP_i$  are incident with the line of sight of the projection matrix  $B$  associated with the image point  $AP_i$ . Since  $AP_i \cong BQ_i$ ,  $Q_i$  is also on this line. This defines a constraint on the projection of these points by an arbitrary projection matrix  $O$ : Let  $e_O = OC_B$ . Then the point  $OQ_i$  resides on the line  $e_O \times OB^+AP_i$ . Therefore:

$$(OQ_i)^T([e_O]_x OB^+A)P_i = 0 \quad (4)$$

By choosing an arbitrary camera matrix  $O$ , the  $3 \times 4$  matrix  $F_O = ([e_O]_x OB^+A)$  can be recovered linearly from the pairs of points  $\{OQ_i, P_i\}_{i=1}^n$ . The introduction of  $O$  enables to extract a unique bi-linear invariant with a minimal number of parameters. There exists a set of invariants described by  $4 \times 4$  matrices  $G$ , satisfying:  $(Q_i)^T(G)P_i = 0$ . These matrices span a linear space of dimension 4. By using the matrix  $O$ , we get a unique invariant with 12 elements.

For each  $O$ , the image  $e_O$  of  $C_B$  can be recovered as the null space of  $F_O^T$ . Using two such projection matrices  $O$ , the 3D position of  $C_B$  can be recovered by triangulation.

$A, B$  can be recovered up to some  $3 \times 3$  matrix  $D$ . All cameras of the form  $DB$  share the same center of projection. Thus  $B$  can be chosen as any  $3 \times 4$  matrix whose left null space is the recovered  $C_B$ . In order to solve for  $A$ ,  $\{BQ_i\}_{i=1}^n$  are first computed. Since for all  $i$ ,  $AP_i \cong BQ_i$  and  $BQ_i, P_i$  are known,  $A$  can be recovered. This can be done by any method for recovering a camera from the projections of known 3D points [6].

## 3 Non-parallel Optical Axes

In the previous section, we studied the case of two cameras viewing the same or opposite directions. In this section, it is assumed that the two cameras have a known constant angle between their optical axes.

The directions of the optical axes of the cameras are given by  $c_i^j T \hat{H}_j$ , and the angle between the axes of the two cameras  $\alpha$  satisfies:

$$\cos^2 \alpha = \frac{(c_i^{1T} \hat{H}_1 \hat{H}_2^T c_i^2)^2}{(c_i^{1T} \hat{H}_1 \hat{H}_1^T c_i^1)(c_i^{2T} \hat{H}_2 \hat{H}_2^T c_i^2)} \quad (5)$$



This defines a nonlinear metric constraint on  $\hat{H}_1, \hat{H}_2$ .

We solve for the case of orthogonal optical axes, where  $\cos^2 \alpha = 0$ , i.e. by Eqn. 5:

$$c_i^{1T} \hat{H}_1 \hat{H}_2^T c_i^2 = 0$$

Each image pair provides one constraint on the matrix  $F = \hat{H}_1 \hat{H}_2^T$ . From the matrix  $F$ , one can extract using SVD the matrices  $\hat{H}_1 D$  and  $\hat{H}_2 D^{-T}$  for some unknown  $3 \times 3$  matrix  $D$ .

The planes at infinity of the two projective reconstructions are the null spaces of  $\hat{H}_1^T, \hat{H}_2^T$ , as in the previous section. The homography through the plane at infinity between frames  $i, k$  of the same sequence  $j$  can be computed similarly to previous case by  $C_k^j \hat{H}_j (C_i^j \hat{H}_j)^{-1}$ . Since the two matrices  $\hat{H}_1, \hat{H}_2$  do not have the same ambiguity, the homography through the plane at infinity between frames of different sequences cannot be computed.

## 4 Non-fixed Parallel Rotation Axes

In previous sections we analyzed self calibration for rigs in which the angle between the optical axes of the cameras remains fixed. In some cases it is useful to enable the cameras to rotate. For example, they may need to be reoriented such that the object of interest appears in the image.

In this section we explore another type of constraints for self calibration of rigs without correspondences between the cameras. It is assumed that the cameras in the rig may rotate arbitrarily about their  $Y$  axis (or similarly about its  $X$  axis). It is further assumed that their  $Y$  axes are parallel. The rig as a whole can rotate and translate freely in space, and the internal parameters of the cameras may vary freely. We describe two cases. In the first case three cameras are used and the internal cameras are not constrained. In the second case two cameras are used, but we assume their skew is 0.

### 4.1 Three Cameras

Let  $\sigma_i^j$  be plane X-Y of camera  $j$  in time instance  $i$ . Since the three cameras rotate about their  $Y$  axes, and the  $Y$  axes are parallel, then in every time instance  $i$ , all planes  $\{\sigma_i^j\}_{j=1,2,3}$  intersect in a point on the plane at infinity. We next show how we express this constraint as a multilinear equation in the projective camera matrices.

Let  $H_j^{-1}$  be the transformation mapping the projective coordinate system of camera  $j$  to a common affine coordinate system in which the plane at infinity is given by  $L = [0 \ 0 \ 0 \ 1]$ . Let  $\{\pi^j\}_{j=1,2,3}$  be three planes, where  $\pi^j$  is given in the coordinate system of the  $j$ -th camera. Then if these planes intersect in a point at infinity, the determinant of the following matrix vanishes:

$$\begin{bmatrix} (\pi^1)^\top H_1 \\ (\pi^2)^\top H_2 \\ (\pi^3)^\top H_3 \\ L \end{bmatrix} \quad (6)$$

This determinant can be written using a  $4 \times 4 \times 4$  tensor  $J^{abc}$ , as

$$\pi_a^1 \pi_b^2 \pi_c^3 J^{abc} = 0$$

The reader is assumed to be familiar with tensor notations. See for example in [9]

The tensor  $J^{abc}$  can now be used to express the calibration constraints. In each time  $i$  all planes  $\{\sigma_i^j\}_{j=1,2,3}$  intersect in a point on the plane at infinity. As  $\sigma_i^j$  is given by  $[0, 0, 1]C_i^j$ , this is expressed by:

$$([0, 0, 1]C_i^1)_a ([0, 0, 1]C_i^2)_b ([0, 0, 1]C_i^3)_c J^{abc} = 0$$

Given the projective camera matrices  $C_i^j$ , every time instance provides a linear constraint on  $J^{abc}$ . Thus  $J^{abc}$  can be computed linearly from the projection matrices.

In order to extract the homographies through the plane at infinity between different images, We identify points at infinity in each coordinate frame. Let  $Y_i^3$  be a double contraction of the form  $Y_i^3 = ([0, 0, 1]C_i^1)_a ([0, 0, 1]^\top C_i^2)_b J^{abc}$ . Let  $N$  be any plane in the third coordinate system intersecting  $\sigma_i^1, \sigma_i^2$  in a point on the plane at infinity. Then:

$$([0, 0, 1]C_i^1)_a ([0, 0, 1]^\top C_i^2)_b N_c J^{abc} = N^\top Y_i^3 = 0$$

Hence  $Y_i^3$  is the point of intersection of  $\{\sigma_i^j\}_{j=1,2,3}$  in the coordinate frame of the third camera. Similarly this point can be extracted in the two other coordinate systems:  $Y_i^1, Y_i^2$ .

The set of points  $\{Y_i^j\}_{i=1..n}$  are sufficient to determine the plane at infinity at the  $j$ th coordinate system. We can use these matching points on the plane at infinity to compute the homography at infinity between all frames: The homography at infinity  $H_{stuv}^\infty$  between frames  $s$  and  $u$  of cameras  $t$  and  $v$  can be computed using the pairs of matching points  $\{(C_s^t Y_i^t, C_u^v Y_i^v)\}_{i=1..n}$ .

## 4.2 Two Cameras with Zero Skew

In this section we show that when the cameras have zero skew, two cameras in the above settings are sufficient for linear recovery of the affine calibration. Then a Metric upgrade can be achieved linearly using standard methods [6].

For every  $i, j$  the projective camera matrix  $C_i^j$  satisfies:

$$C_i^j \hat{H}_j \cong K_i^j R_j^i R_i$$

where  $R_j^i$  is the rotation of the  $j$ -th camera with respect to the first camera in time instance  $i$ , and  $R_i$  is the rotation of the rig in the same time. Again we base the derivation on the structure of a rotation matrix  $R_j^i$  about the  $Y$  axis. Since the first and third rows of  $R_j^i$  have a vanishing second coordinate, then so is their linear combination. Assuming zero skew, the first and last rows of  $C_i^j \hat{H}_j$  are linear combinations of the first and third rows of  $R_i$

Let  $a_i^j$  be the first row of the projective camera matrix  $C_i^j$ . Then the following determinant vanishes for every  $i$ :

$$\begin{bmatrix} (a_i^1)^\top \hat{H}_1 \\ (c_i^1)^\top \hat{H}_1 \\ (c_i^2)^\top \hat{H}_2 \end{bmatrix}$$

This constraint can be expressed by a trilinear tensor:

$$([1, 0, 0]C_i^1)_a([0, 0, 1]C_i^1)_b([0, 0, 1]C_i^2)_c K^{abc} = 0$$

The tensor  $K^{abc}$  can be solved linearly from the projection matrices. As in the previous section, this constraint can be interpreted geometrically as the intersection of three planes with the plane at infinity. Hence the plane at infinity of the coordinate systems of the two cameras can be extracted by a method similar to the one in the previous section.

Note that similar tensors can be derived by choosing other three-combinations of  $\hat{H}_1^\top a_i^1, H_1^\top c_i^1, \hat{H}_2^\top a_i, \hat{H}_2^\top c_i$

## 5 Sequences Synchronization

In the previous sections it was assumed that the sequences are synchronized. This enabled to compute the multilinear constraints:  $F_O$  of section 2.1,  $F$  of section 3 and  $J^{abc}$  of Section 4. However the existence of these constraints may be used to establish the synchronization between the sequences.

Consider for example the case of the orthogonal optical axes presented in section 3. The existence of a matrix  $F$  such that for every frame  $i$ :  $c_i^{1\top} F c_i^2 = 0$  is not guaranteed if the two sequences are not temporally aligned. Let  $A$  be the estimation matrix of  $F$ , i.e  $A$  is the matrix whose  $i$ th row is composed from the Kronecker product of  $c_i^1$  and  $c_i^2$ . If such an  $F$  exists then the rank of the matrix  $A$  is not more than 8 and the vector composed from the elements of  $F$  lies in the null space of  $A$ . In practice due to noise  $A$  is always of full rank, and we use a least squares solution, choosing  $F$  to be composed out of the elements of the eigenvector of  $AA^\top$  with the smallest eigenvalue. Let  $f$  be this eigenvector. We define the magnitude of  $Af$  as the algebraic error of the estimation of  $F$ . When the sequences are synchronized this magnitude is expected to be small.

In order to synchronize the sequences this algebraic error is measured for each temporal shift between the sequences. The shift which produces the minimal algebraic error is chosen as the solution. This measure is not optimal for a number of reasons. First it has no real geometrical meaning. Second it depends on the number of the frames which can bias in short sequences toward the ends of the sequence. Third, changing the coordinates of  $c_i^j$  changes the measure. However in practice we find minimizing the algebraic error to work well on our sequences.

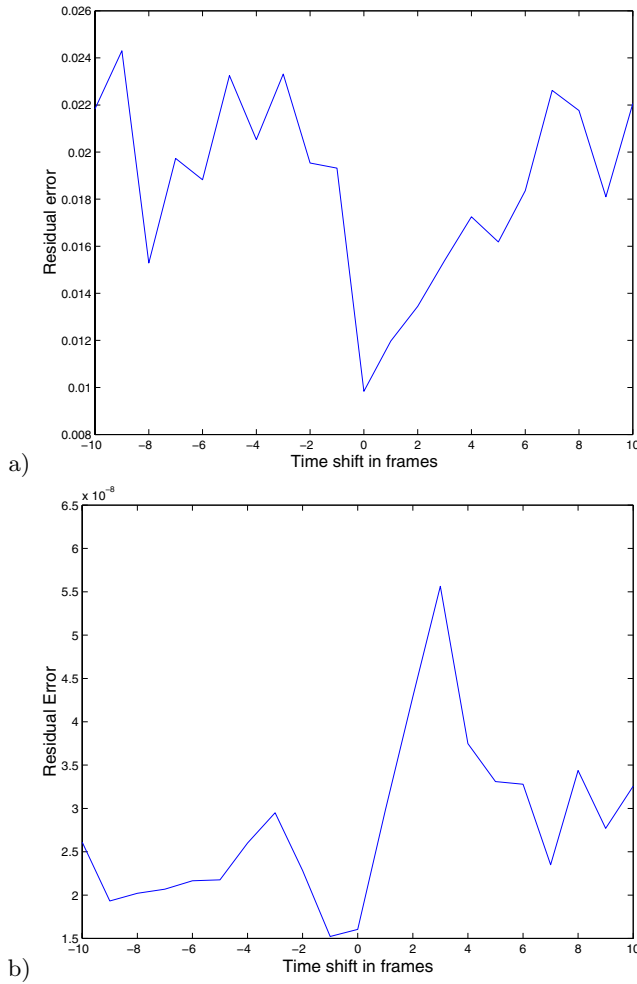
## 6 Experiments

We have conducted several experiments for testing the quality of the affine self-calibration and the sequence synchronization. In all our experiments we have used an object modeling framework: The cameras were static, and an object to be modeled was rotating in the scene. Points on this object were tracked, and the virtual camera motion with respect to the object was estimated. Since no calibration was available, the cameras and points had a projective ambiguity to be recovered in the experiments. Then the proposed algorithm was used, estimating an Affine representation of the structure and motion.

In the first experiment we tested the sequence synchronization application. The algorithm was applied for several temporal shifts, and the sequences were temporally aligned by finding the shift with the minimal residual error, as described in Section. 5. Figures 1-a,b present the results of this experiment for opposite and orthogonal directions respectively. The true shift was estimated using hand waving in front of the camera. Note that this is an integer estimation for the non-integer temporal shift of the sequences, and thus it is accurate up to 0.5 a frame. In the experiments we tested 10 frames shifts in each temporal direction on sequences containing 40 frames. Indeed a shift close to the estimated by no more than 1 frame yielded the minimal estimation error.

One way to verify the accuracy of affine calibration is to test the accuracy of the homographies through the plane at infinity. In the second experiment we tested the accuracy of the homography through the plane at infinity by mapping vanishing points between the images. We have marked points lying on parallel lines, and tracked them along the sequences. The vanishing points at each frame were computed as the intersections of the parallel lines defined by the tracked points. We then mapped the vanishing points from the input images to a common coordinate system. The quality of the homographies was measured by the proximity of the mapped vanishing points. We have conducted this test for two scenarios: One for cameras viewing opposite directions (Section. 2), and one for cameras viewing orthogonal directions (Section. 3). Figure 2 shows the results of these experiments. Errors in these results are combined from errors in the Affine calibration, errors in the projective camera matrices (the input to our algorithm), and errors in the estimations of the vanishing points in the images due to drifts in the points tracking. However it is visible that the algorithm does indeed align the vanishing points.

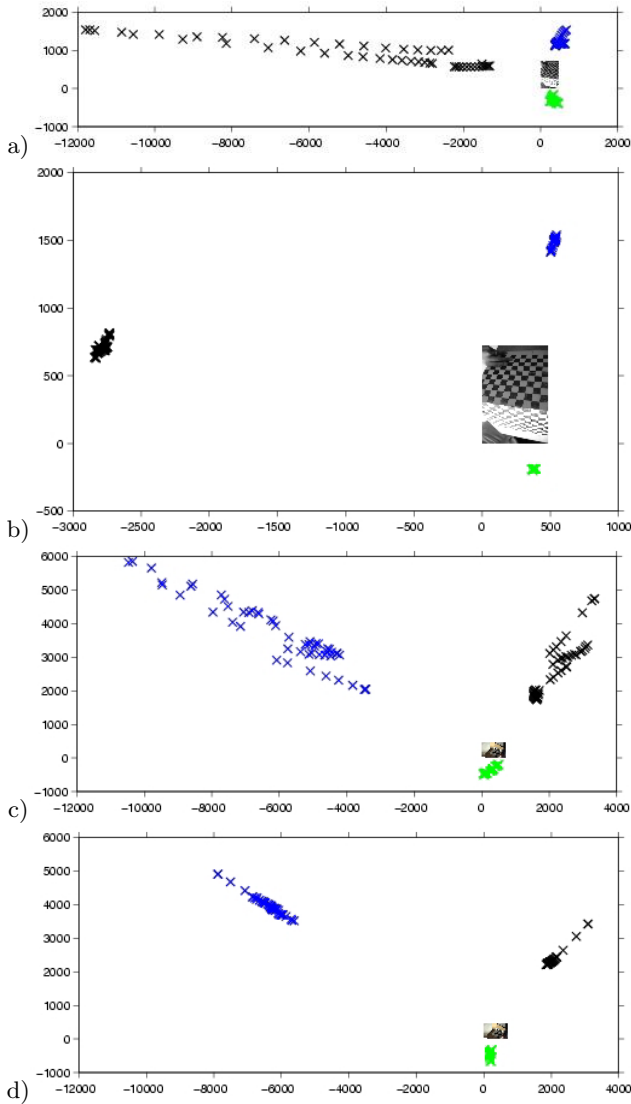
Finally, the quality of the homographies can also be tested visually. Warping a sequence of images of a moving camera to a common reference by the homographies through the plane at infinity cancels the rotations of the cameras, as well as the variations in the internal parameters. The result is a sequence in which the camera virtually moves in pure translation and constant internal parameters. Figure. 3 shows the results of applying the computed homographies through the plane at infinity to several images. Note that while the original motion included a rotation, the homographies through the plane at infinity canceled the rotations, leaving only a translational component.



**Fig. 1.** Sequences synchronization for cameras viewing opposite directions (Fig. 1-a) and for cameras viewing orthogonal directions (Fig. 1-b). The residual error of the estimation matrix is presented as a function of the temporal shift between the sequences, where 0 shift corresponds to the manual integer estimation of the real shift. The residual error for the orthogonal case is averaged over four random choices of the camera matrix  $O$ .

## 7 Summary and Future Work

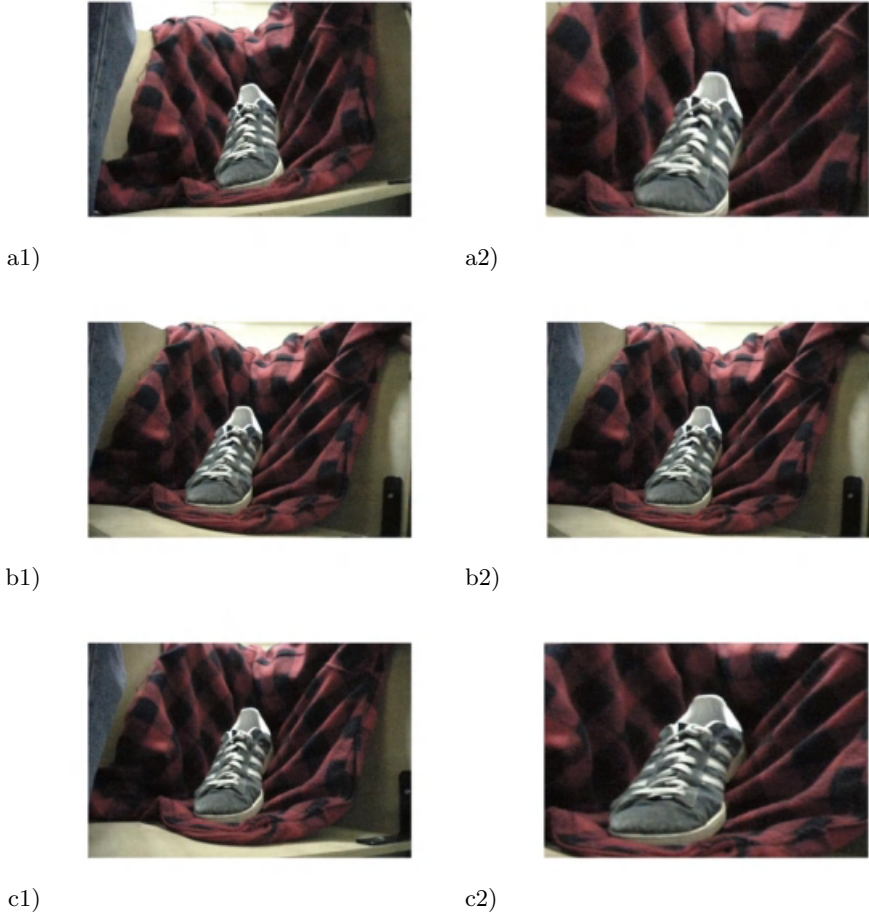
We have analyzed self calibration of rigs with varying internal parameters, when no correspondences are available between the cameras. The only assumption used for the affine calibration, was that the angle between the viewing directions of the cameras is known.



**Fig. 2.** Mapping vanishing points by the homographies through the plane at infinity extracted by the proposed algorithm. a) The original vanishing points, opposite viewing directions. b) The mapped vanishing points, opposite viewing directions. c) The original vanishing points, orthogonal viewing directions. d) The mapped vanishing points, orthogonal viewing directions. The size of the images in illustrations a,b and c,d visualize the scale differences between the coordinate axes.

We presented constraints for general angles, and solved specific cases of parallel, opposite and orthogonal angles by extracting affine invariants.

Future work can be solving similar cases by deriving metric invariants, e.g.:



**Fig. 3.** Using the computed homographies through the planes at infinity to generate a virtual motion of pure translation. Figures b1,b2 contain the same image, to which the input images were warped. Figures a1,c1 to the left are the original input images. Figures a2,c2 to the right are the results of warping a1,c1 by the respective homographies through the plane at infinity.

- Solving the case of arbitrary angles, with or without knowing the angle.
- Assuming that the distances between the centers of projections of the cameras remain fixed during the motion.

We hope that even if such invariants are not compact, they may find use in some application such as cameras synchronization.

We plan to implement such algorithms for self calibration on “domes” containing cameras with fixed orientations and varying Zoom/Focus.

## References

1. Y. Caspi, and M. Irani Alignment of Non-Overlapping Sequences In *Proceedings of the International Conference on Computer Vision*, Vol, II, pages 76-83, Vancouver, Canada, July 2001.
2. F. Dornaika, F. Self-Calibration of a Stereo Rig Using Monocular Epipolar Geometry. In *Proceedings of the International Conference on Computer Vision*, Vol, II, pages 467-472, Vancouver, Canada, July 2001.
3. O.D. Faugeras. Stratification of three-dimensional vision: projective, affine and metric representations. *Journal of the Optical Society of America*, 12(3):465-484, 1995.
4. O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self calibration: Theory and experiments. In *Proceedings of the European Conference on Computer Vision*, pages 321-334, Santa Margherita Ligure, Italy, June 1992.
5. R. Hartley. Self calibration from multiple views with a rotating camera. In *Proceedings of the European Conference on Computer Vision*, pages 471-478, Stockholm, Sweden, May 1994.
6. R.I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.
7. R. Horaud, G. Csurka, and D. Demirdijian Stereo Calibration from Rigid Motions *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(22):1446-1452, December 2000.
8. M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown camera parameters. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998.
9. A. Shashua and L. Wolf. Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Planar Points. in Proc. of the European Conference on Computer Vision (ECCV), June 2000, Dublin, Ireland.
10. A. Shashua Omni-Rig Sensors: What Can be Done With a Non-Rigid Vision Platform?. In *Proc. of the Workshop on Applications of Computer Vision (WACV)*, Princeton, Oct. 1998.
11. B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 609-614, Puerto Rico, June 1997. IEEE Computer Society Press.
12. A. Zisserman, P.A. Beardsley, and I.D. Reid. Metric calibration of a stereo rig. In *Proc. of IEEE International Conference in Computer Vision*, Vol. I. pages 135-141, Vancouver, Canada, July 2001.
13. A. Zomet, L. Wolf and A. Shashua. Omni-Rig: Linear Self-Recalibration of a Rig with Varying Internal and External Parameters In *Proceedings of the International Conference on Computer Vision*, Vol. I, pages 135-141, Vancouver, Canada, July 2001.



# Structure from Planar Motions with Small Baselines

René Vidal<sup>1</sup> and John Oliensis<sup>2</sup>

<sup>1</sup> Department of EECS, 301 Cory Hall, Berkeley CA 94710  
rvidal@eecs.berkeley.edu

<sup>2</sup> NEC Research Institute, 4 Independence Way, Princeton NJ 08540  
oliensis@research.nj.nec.com

**Abstract.** We study the multi-frame structure from motion problem when the camera translates on a plane with small baselines and arbitrary rotations. This case shows up in many practical applications, for example, in ground robot navigation. We consider the framework for small baselines presented in [8], in which a factorization method is used to compute the structure and motion parameters accurately, efficiently and with guaranteed convergence. When the camera translates on a plane, the algorithm in [8] cannot be applied because the estimation matrix drops rank, causing the equations to be no longer linear. In this paper, we show how to linearly solve those equations, while preserving the accuracy, speed and convergence properties of the non-planar algorithm. We evaluate the proposed algorithms on synthetic and real image sequences, and compare our results with those of the optimal algorithm. The proposed algorithms are very fast and accurate, have less than 0.3% outliers and work well for small-to-medium baselines and non-planar as well as planar motions.

## 1 Introduction

The structure from motion problem has been studied extensively over the past decades and many algorithms have been proposed for general camera motion (see [12] for batch methods, [5,10] for recursive methods, [11,13,14] for factorization methods and [2] for projective methods).

However, most of these algorithms are not designed to give accurate answers when the baselines are small, which is the most complex case because the signal-to-noise ratio is small. Since the small baseline case shows up in most vision applications in control and robotics, we believe it is fundamental to design algorithms that work well in this case.

In [8], an algorithm explicitly designed for small baselines was presented, showing the possibility of computing structure and motion parameters accurately, efficiently and with guaranteed convergence. The translation is taken as zero initially, and then the algorithm repeatedly updates the rotation, translation, and depth estimates until they converge (convergence depends explicitly on the small baseline assumption). The translation and depth estimates are

obtained linearly from an approximate rank-3 factorization of a matrix which depends on the image displacements and the current rotation estimates.

The above algorithm works only for non-planar motion, *i.e.*, when the camera positions do not all lie in a plane or line. When the camera motion is planar or linear (as happens for example in ground robot navigation), the matrix used in [8] to compute the translations and depths has an approximate rank of 2 or 1, rather than 3, and thus the algorithm of [8] is not applicable.

In [7,9] (see also [3]), an algorithm is proposed which is designed to give accurate results for small baselines and linear motion. Though it also works for planar or fully non-planar motions, it gives less accurate results for these cases.

In this paper, we study the *planar motion* case in more detail. We show that, even though the equations relating depth and translation are no longer linear (as they were in the non-planar or linear motion cases), it is possible to cancel the non-linearities to obtain linear solutions. The proposed algorithms are very fast and accurate, have less than 0.3% outliers and work well for small-to-medium baselines and non-planar as well as planar motions.

## 2 Non-planar Motion Algorithm

We consider an image sequence containing  $N_P$  points in  $N_F$  frames. We use  $\mathbf{x}_p^i = (x_p^i, y_p^i, 1)^T$ ,  $p = 1 \cdots N_P$ ,  $i = 0 \cdots N_F - 1$ , to denote the image coordinates of the  $p^{th}$  point in the  $i^{th}$  image and choose the zeroth image as the reference frame. By convention  $\mathbf{x}_p = (x_p, y_p, 1)^T = (x_p^0, y_p^0, 1)^T$ . The motion of the  $i^{th}$  camera frame with respect to the zeroth is described by a rotation matrix  $R^i \in SO(3)$  and a translation vector  $T^i = (T_x^i, T_y^i, T_z^i)^T \in \mathbb{R}^3$ . Let  $Z_p$  be the depth of the  $p^{th}$  point in 3D with respect to the zeroth camera frame. The image points in the  $i^{th}$  frame are related to those of the zeroth by:

$$\lambda \mathbf{x}_p^i = R^i(\mathbf{x}_p - T^i/Z_p), \quad \lambda = [R^i(\mathbf{x}_p - T^i/Z_p)]_z. \quad (1)$$

Let  $\tau$  be the ratio between the largest translation and the smallest depth, *i.e.*,  $\tau = T_{\max}/Z_{\min}$ . We say that the *baselines are small* if  $\tau \ll 1$ . Under this assumption, one can initialize all the translations to be zero and then solve linearly for the rotations from (1). It is shown in [6] that the errors  $\|\Omega^i\|$  between these rotation estimates  $R_{est}^i$  and the true rotations  $R_{true}^i$  are approximately proportional to  $\tau$ , where  $\Omega^i \in \mathbb{R}^3$  is such that  $R_{true}^i R_{est}^{iT} = \exp([\Omega^i]_{\times}) \in SO(3)$ . (Here,  $[u]_{\times} \in so(3)$  represents the skew-symmetric matrix generating the cross product, *i.e.*, for all  $u, v \in \mathbb{R}^3$  we have  $u \times v = [u]_{\times} v$ ).

The initial rotation estimates  $R_{est}^i$  are used to warp the image points  $\mathbf{x}_p^i$  from the  $i^{th}$  to the reference frame. From the warped image points, we define a vector of *displacements*  $\mathbf{d}_p^i$  with respect to the reference frame as:

$$\mathbf{d}_p^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left( \frac{R_{est}^{iT} \mathbf{x}_p^i}{(R_{est}^{iT} \mathbf{x}_p^i)_z} - \mathbf{x}_p \right),$$

from which we form a *displacement* matrix  $D \in \mathbb{R}^{2N_P \times (N_F-1)}$ . It is shown in [8] that the displacement matrix satisfies:

$$D \approx \Phi(Z^{-1}) T + \Psi \Omega, \tag{2}$$

where

$$\begin{aligned} \Phi(Z^{-1}) &= \begin{bmatrix} -\{Z^{-1}\} & 0 & \{xZ^{-1}\} \\ 0 & -\{Z^{-1}\} & \{yZ^{-1}\} \end{bmatrix} \in \mathbb{R}^{2N_P \times 3} \\ T &= [T^1 \ \dots \ T^{N_F-1}] \in \mathbb{R}^{3 \times (N_F-1)} \\ \Psi &= \begin{bmatrix} -\{xy\} & \{1+x^2\} & -\{y\} \\ -\{1+y^2\} & \{xy\} & \{x\} \end{bmatrix} \in \mathbb{R}^{2N_P \times 3} \\ \Omega &= [\Omega^1 \ \dots \ \Omega^{N_F-1}] \in \mathbb{R}^{3 \times (N_F-1)}. \end{aligned}$$

In the above equations,  $\{xZ^{-1}\}$  (for example) denotes a vector whose  $p^{th}$  component is  $x_p Z_p^{-1}$ . Notice that  $\Psi$  can be interpreted as the matrix of rotational flows and  $\Omega$  as the matrix of (unknown) residual rotational velocities.

Equation (2) depends on  $\Omega$  which is of order  $\tau$  and thus approximately proportional to the size of the translations. Therefore, if we solved for  $T$  from (2) after neglecting the unknown  $\Omega$ , we would obtain translation estimates with an error of order  $\tau$ , *i.e.*, of the order of the translations themselves. In order to obtain translation estimates with a small error of order  $\tau^2$ , we define a matrix  $H \in \mathbb{R}^{(2N_P-3) \times 2N_P}$  annihilating the rotational flows, *i.e.*, such that  $H\Psi = 0$ . Multiplying (2) by  $H$  gives:

$$HD \approx H\Phi(\{Z^{-1}\}) T. \tag{3}$$

We conclude that  $HD$  (which can be computed from the given image points and the initial rotations) has a rank that is approximately equal to the rank of  $T$ , which is either 1, 2 or 3, depending on whether the motion is linear, planar or non-planar, respectively. In the non-planar motion case, one can use the singular value decomposition to factorize  $HD$  into its structure  $S \in \mathbb{R}^{(2N_P-3) \times 3}$  and motion  $M \in \mathbb{R}^{(N_F-1) \times 3}$  components as:

$$HD = SM^T = SUU^{-1}M^T \approx H\Phi(\{Z^{-1}\}) T, \tag{4}$$

where  $U \in \mathbb{R}^{3 \times 3}$  is an arbitrary nonsingular matrix. Given this factorization, one can solve linearly for  $\{Z^{-1}\}$  and  $U$  from the equation  $H\Phi(\{Z^{-1}\}) = SU$  and obtain the translation vectors from  $T = U^{-1}M^T$ .

Given these new estimates for the translations and depths, the algorithm in [8] improves the initial estimates for the rotation linearly from (1). Then, the algorithm repeatedly updates the rotation, translation, and depth estimates until they converge. Under the small baseline assumption, one can show that the incremental change in the unknowns between the  $k^{th}$  and  $(k+1)^{th}$  iteration is approximately proportional to  $\tau^k$ , and hence the algorithm has good convergence properties. (See [8] for details).

We summarize the algorithm for non-planar motion as follows<sup>1</sup>:

<sup>1</sup> The algorithm in [8] differs from this description by including an iteration that corrects the small-baseline approximation in (2).

**Algorithm 1 (Non-planar Motion Algorithm)** *Given a set of  $N_P$  corresponding image points  $\{\mathbf{x}_p^i\}$ ,  $p = 1, \dots, N_P$ ,  $i = 0, \dots, N_F - 1$ , with respect to  $N_F$  camera frames, compute the motion  $(R, T)$  and the depth  $\{Z\}$  as follows:*

1. Initialize  $T = 0$ .
2. Solve for  $R$  linearly from (1), given  $T$  and  $\{Z\}$ .  $\{Z\}$  is unnecessary if  $T = 0$ .
3. Given  $R$  compute  $D$ . Then compute  $S$  and  $M$  from the SVD of  $HD$ .
  - a) Solve for  $\{Z\}$  and  $U$  linearly from  $H\Phi(\{Z^{-1}\}) = SU$ , given  $S$ .
  - b) Solve for  $T = U^{-1}M^T$ , given  $U$  and  $M$ .
4. Goto 2. until  $(R, T, \{Z\})$  converge.

### 3 Planar Motion Algorithms

In the non-planar motion case, translation and depth parameters are estimated linearly from the equation:

$$H\Phi(\{Z^{-1}\})V = SU \quad (5)$$

where the columns of  $V \in \mathbb{R}^{3 \times 3}$  are a basis for the translation vectors and one can choose  $V = I_3$  without loss of generality.

Assume now that all translation vectors lie in a plane, so that  $\text{rank}(T) = 2$ . The data matrix  $HD$  now factors into rank-2 matrices  $S \in \mathbb{R}^{(2N_P-3) \times 2}$  and  $M \in \mathbb{R}^{(N_F-1) \times 2}$ . In (5), we now have  $U \in \mathbb{R}^{2 \times 2}$  and  $V \in \mathbb{R}^{3 \times 2}$ . The matrix  $V$  is defined by its orthogonality to the normal  $\pi \in \mathbb{S}^2$  to the true plane of motion, and thus has two degrees of freedom. Since there is no global parameterization for  $\mathbb{S}^2$ , hence for  $V$ , in order to solve (5) we will need to choose a set of local parameterizations for  $V$ . Further, notice that since (5) is bilinear in  $\{Z^{-1}\}$  and  $V$ , we cannot proceed as in step 3(a) of the non-planar motion algorithm.

In the following subsections, we show how to cancel the nonlinearity in (5) due to  $V$  and hence obtain a linear solution for  $\{Z\}$  and  $T$ .

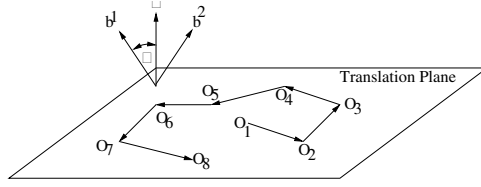
#### 3.1 Multiple $b$ Algorithm

Let  $\pi$  be the normal to the plane of motion. We consider vectors  $b^i \in \mathbb{R}^3$ ,  $i = 1, \dots, m$  that are not perpendicular to  $\pi$ , i.e., vectors that do not lie in the true plane of motion, and parameterize  $V$  depending on the directions specified by these  $b^i$ 's (see Fig. 1). We start with the simplest case of a single  $b$ .

$b = [0, 0, 1]^T$ . Assume that the normal  $\pi$  to the true plane of motion does not lie in the  $X$ - $Y$  plane. Then (5) can be written as:

$$H \begin{bmatrix} -\{Z^{-1}\} & \{0\} & \{xZ^{-1}\} \\ \{0\} & -\{Z^{-1}\} & \{yZ^{-1}\} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ v_1 & v_2 \end{bmatrix} = SU, \quad (6)$$

where  $v_1 \in \mathbb{R}$  and  $v_2 \in \mathbb{R}$  are unknowns chosen to make the columns of  $V$  perpendicular to  $\pi$ . Define the following matrices in  $\mathbb{R}^{(2N_P-3) \times N_P}$ :  $H_x = H(:, 1:N_P)$ ,



**Fig. 1.** Multiple b algorithm for planar motions.

$H_y = H(:, N_P + 1 : 2N_P)$  and  $H_z = H_x \text{diag}(\{x\}) + H_y \text{diag}(\{y\})$ . Then (6) can be rewritten as:

$$\begin{aligned} (-H_x + v_1 H_z) \{Z^{-1}\} &= SU_1, \\ (-H_y + v_2 H_z) \{Z^{-1}\} &= SU_2, \end{aligned} \quad (7)$$

where  $U = [U_1, U_2] \in \mathbb{R}^{2 \times 2}$ . Define a  $(N_P - 3) \times (2N_P - 3)$  matrix  $N_z$  to annihilate the columns of  $H_z$ .  $N_z$  can be computed quickly using Householder matrices. We obtain:

$$\begin{bmatrix} N_z H_x & N_z S & 0 \\ N_z H_y & 0 & N_z S \end{bmatrix} \begin{bmatrix} \{Z^{-1}\} \\ U_1 \\ U_2 \end{bmatrix} \equiv G \begin{bmatrix} \{Z^{-1}\} \\ U_1 \\ U_2 \end{bmatrix} = 0. \quad (8)$$

Multiplying by  $N_z$  introduces an additional solution  $[\{Z\}; U_1; U_2] = [\{1\}; 0; 0]$  to the system of equations in (6). In order to show that this is a solution, we just need to prove that  $N_z H_x \{1\} = N_z H_y \{1\} = 0$ . Since  $H\Psi = 0$ , we have  $H_x \{1 + x^2\} + H_y \{xy\} = 0$ . Therefore  $-N_z H_x \{1\} = N_z (H_x \{x^2\} + H_y \{xy\}) = N_z H_z \{x\} = 0$ . Similarly, one can show that  $-N_z H_y \{1\} = N_z H_z \{y\} = 0$ .

In the presence of noise, the solution  $[\{1\}; 0; 0] \in \mathbb{R}^{N_P+4}$  will still correspond to the zero singular value of  $G$ , while the solution we are looking for will be a linear combination of the two smallest singular vectors of  $G$ . That is, the first and second singular vectors of  $G$  equal  $[\{1\}; 0; 0]$  and  $[\zeta; U_1; U_2]$ , respectively, where  $\zeta \in \mathbb{R}^{N_P}$  represents the first  $N_P$  entries of the second singular vector. Therefore,  $U_1$  and  $U_2$  can be obtained directly from the second singular vector of  $G$ , while the inverse depths equal  $\zeta$  up to an unknown additive mixture of the constant term  $\{1\}$ , i.e.,  $\{Z^{-1}\} = \lambda \{1\} + \zeta$  for some  $\lambda \in \mathbb{R}$ .

Replacing the above expression for  $\{Z^{-1}\}$  in (7) we obtain:

$$\begin{aligned} -H_x \{1\} \lambda + H_z \zeta v_1 + H_z \{1\} \lambda v_1 &= H_x \zeta + SU_1 \\ -H_y \{1\} \lambda + H_z \zeta v_2 + H_z \{1\} \lambda v_2 &= H_y \zeta + SU_2. \end{aligned}$$

The bilinear terms  $\lambda v_1$  and  $\lambda v_2$  can be canceled by multiplying these equations by  $\tilde{N}_z \in \mathbb{R}^{(2N_P-4) \times (2N_P-3)}$  such that  $\tilde{N}_z H_z \{1\} = 0$ . We obtain:

$$\begin{bmatrix} -\tilde{N}_z H_x \{1\} & \tilde{N}_z H_z \zeta & 0 \\ -\tilde{N}_z H_y \{1\} & 0 & \tilde{N}_z H_z \zeta \end{bmatrix} \begin{bmatrix} \lambda \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \tilde{N}_z (H_x \zeta + SU_1) \\ \tilde{N}_z (H_y \zeta + SU_2) \end{bmatrix},$$

from which one can solve linearly for  $\lambda$ ,  $v_1$  and  $v_2$ . Note that  $\{Z^{-1}\}$  is now known since  $\zeta$  and  $\lambda$  are. One can improve these initial estimates for  $\{Z^{-1}\}$  and  $U$  by solving (7) given  $v_1$  and  $v_2$ . The plane normal is given by  $\pi = (-v_1; -v_2; 1)$  and the translations are  $T = VU^{-1}M^T$ , where  $M$  is the motion factor of  $HD$ .

**Arbitrary single  $b$ .** Assume that  $b \in \mathbb{R}^3$  is a given vector with a nonzero component along the true plane normal, *i.e.*,  $b^T \pi \neq 0$ . Let  $a_1$ ,  $a_2$  and  $b$  form an orthonormal basis of  $\mathbb{R}^3$ . Then a basis for the translation plane is given by  $V = [a_1 + v_1 b \quad a_2 + v_2 b]$  for some unknown  $v_1 \in \mathbb{R}$  and  $v_2 \in \mathbb{R}$ . Equation (5) can be written as:

$$\begin{aligned} (H_{a_1} + v_1 H_b) \{Z^{-1}\} &= SU_1, \\ (H_{a_2} + v_2 H_b) \{Z^{-1}\} &= SU_2, \end{aligned} \quad (9)$$

where  $H_{a_1} = -a_{1x}H_x - a_{1y}H_y + a_{1z}H_z$ ,  $H_{a_2} = -a_{2x}H_x - a_{2y}H_y + a_{2z}H_z$  and  $H_b = -b_xH_x - b_yH_y + b_zH_z$ .

As before, we define  $N_b \in \mathbb{R}^{(N_P-3) \times (2N_P-3)}$  to annihilate the columns of  $H_b$  and obtain the system of equations:

$$\begin{bmatrix} -N_b H_{a_1} & N_b S & 0 \\ -N_b H_{a_2} & 0 & N_b S \end{bmatrix} \begin{bmatrix} \{Z^{-1}\} \\ U_1 \\ U_2 \end{bmatrix} \equiv G \begin{bmatrix} \{Z^{-1}\} \\ U_1 \\ U_2 \end{bmatrix} = 0.$$

The two smallest singular vectors of  $G$  are now  $[\zeta_1; 0; 0]$  and  $[\zeta_2; U_1; U_2]$ , with  $\zeta_1 = b_x\{x\} + b_y\{y\} + b_z\{1\}$ . The inverse depths are  $\{Z^{-1}\} = \lambda\zeta_1 + \zeta_2$ , where  $\lambda$  is obtained by solving:

$$\begin{bmatrix} \tilde{N}_b H_{a_1} \zeta_1 & \tilde{N}_b H_b \zeta_2 & 0 \\ \tilde{N}_b H_{a_2} \zeta_1 & 0 & \tilde{N}_b H_b \zeta_2 \end{bmatrix} \begin{bmatrix} \lambda \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \tilde{N}_b (SU_1 - H_{a_1} \zeta_2) \\ \tilde{N}_b (SU_2 - H_{a_2} \zeta_2) \end{bmatrix},$$

with  $\tilde{N}_b \in \mathbb{R}^{(2N_P-4) \times (2N_P-3)}$  defined to annihilate  $H_b \zeta_1$ .

Finally, given  $v_1$  and  $v_2$  one can re-solve for  $\{Z^{-1}\}$  and  $U$  from (9) to improve the estimates. The plane normal is given by  $\pi = (a_1 + v_1 b) \times (a_2 + v_2 b)$  and the translations are obtained as  $T = VU^{-1}M^T$ .

**Multiple  $b$  algorithm.** We conclude from the previous section that using a single  $b$  to parameterize  $V$  has the disadvantage of introducing an additional solution to the system of equations in (9). Although this is not a problem from an algebraic point of view since we have shown how to resolve the ambiguity, in the presence of noise it can lead to solutions which are not robust.

For a single  $b$ , the additional solution has the form  $[b_x\{x\} + b_y\{y\} + b_z\{1\}; 0; 0]$ . Therefore, it can be eliminated by choosing more than one  $b$ . Let  $a_1^i$ ,  $a_2^i$ ,  $b^i$ ,  $i = 1, \dots, m$  be a set of orthonormal bases for  $\mathbb{R}^3$  and assume none of the  $b^i$ 's lies in the translation plane. One can solve for  $\{Z^{-1}\}$ ,  $U_1^i$  and  $U_2^i$  uniquely from

the set of equations:

$$\begin{bmatrix} -N_{b^1}H_{a_1^1} & N_{b^1}S & 0 & \cdots & 0 & 0 \\ -N_{b^1}H_{a_2^1} & 0 & N_{b^1}S & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ -N_{b^m}H_{a_1^m} & 0 & 0 & & N_{b^m}S & 0 \\ -N_{b^m}H_{a_2^m} & 0 & 0 & \cdots & 0 & N_{b^m}S \end{bmatrix} \begin{bmatrix} \{Z^{-1}\} \\ U_1^1 \\ U_2^1 \\ \vdots \\ U_1^m \\ U_2^m \end{bmatrix} = 0. \quad (10)$$

Given  $\{Z^{-1}\}$  and  $U^i$ , one can solve for  $v_j^i$ ,  $i = 1, \dots, m$  and  $j = 1, 2$  from:

$$H_{b^i}\{Z^{-1}\}v_j^i = SU_j^i - H_{a_j^i}\{Z^{-1}\}. \quad (11)$$

Given  $v_j^i$  one can improve the estimates for  $\{Z^{-1}\}$  and  $U_j^i$  by re-solving (11). Finally, let  $\pi^i = (a_1^i + v_1^ib^i) \times (a_2^i + v_2^ib^i)$  and  $\Pi = [\pi^1 \dots \pi^m] \in \mathbb{R}^{3 \times m}$ . The plane normal is obtained as the leading left singular vector of  $\Pi$  and the translations are obtained as:

$$T = \frac{1}{m} \left( \sum_{i=1}^m V^i U^{i-1} \right) M^T. \quad (12)$$

**Choosing the  $b$ 's.** Since the  $b^i$ 's cannot be perpendicular to the true plane normal, some estimate of  $\pi$  is needed in order to choose the  $b^i$ 's. If  $\pi = b$ , we have  $v_1 = v_2 = 0$ . Therefore, we can obtain an initial estimate of  $\pi$  as follows:

1. Obtain three estimates of  $(\pi, v_1, v_2)$  from the single  $b$  algorithm with  $b$  equal to the  $X$ ,  $Y$  and  $Z$  axes.
2. Choose the  $\pi$  that minimizes  $v_1^2 + v_2^2$ .

Given an initial estimate for  $\pi$ , one has to choose  $b^i$ 's so that they are not perpendicular to  $\pi$  and not close to each other. We choose all the  $b$ 's to be randomly distributed on a cone with angle  $\alpha$  from  $\pi$ . Experimentally, the best performance is obtained with  $\alpha \approx 37^\circ$  and three different  $b$ 's. Using more  $b$ 's has the disadvantage of over-parameterizing  $U$ , while one can show theoretically that using two  $b$ 's does not exploit all the image data at some image points.

**Algorithm 2 (Multiple  $b$  algorithm for planar motions)** *Given a set of corresponding image points  $\{\mathbf{x}_p^i\}$ ,  $p = 1, \dots, N_P$ ,  $i = 0, \dots, N_F - 1$ , compute the motion  $(R, T)$  and the depth  $\{Z\}$  as follows:*

1. *Initialization*
  - a) *Solve for  $R$  linearly from (1), given  $T = 0$ .*
  - b) *Given  $R$  compute  $D$ . Then compute  $S$  and  $M$  from the SVD of  $HD$ .*
  - c) *Compute  $(\{Z\}, T, \pi)$  from “best” single  $b$  algorithm along  $X$ ,  $Y$  or  $Z$ .*
  - d) *Use  $\pi$  to generate multiple  $b$ 's in a cone with angle  $\alpha$  along  $\pi$ .*
2. *Solve for  $R$  linearly from (1), given  $T$  and  $\{Z\}$ .*

3. Given  $R$  compute  $D$ . Then compute  $S$  and  $M$  from the SVD of  $HD$ .
  - a) Solve for  $\{Z\}$  and  $U^1 \dots U^m$  from (10), given the current  $b$ 's.
  - b) Solve for  $\pi$  from SVD of  $\Pi = [\pi^1 \dots \pi^m]$  and for  $T$  from (12).
  - c) Use  $\pi$  to generate multiple  $b$ 's in a cone with angle  $\alpha$  along  $\pi$ .
4. Goto 2. until  $(R, T, \{Z\})$  converge.

### 3.2 The Intersection Algorithm

Let the columns of  $V = [V_1, V_2] \in \mathbb{R}^{3 \times 2}$  be a basis for the translation plane and let  $N_s \in \mathbb{R}^{(2N_P-5) \times (2N_P-3)}$  be a matrix annihilating  $S$ . From (5), we obtain:

$$\begin{aligned} (-H_x V_{jx} - H_y V_{jy} + H_z V_{jz}) \{Z^{-1}\} &= S U_j \\ N_s (-H_x V_{jx} - H_y V_{jy} + H_z V_{jz}) \{Z^{-1}\} &= 0 \\ [N_s H_x \{Z^{-1}\} \quad N_s H_y \{Z^{-1}\} \quad -N_s H_z \{Z^{-1}\}] V_j &= 0, \end{aligned}$$

for  $j = 1$  and  $2$ . We conclude that the matrix

$$\mathcal{I} = [N_s H_x \{Z^{-1}\} \quad N_s H_y \{Z^{-1}\} \quad -N_s H_z \{Z^{-1}\}] \in \mathbb{R}^{(2N_P-5) \times 3}$$

has rank 1. Since  $\pi^T V = 0$  we further have

$$[N_s H_x \{Z^{-1}\} \quad N_s H_y \{Z^{-1}\} \quad -N_s H_z \{Z^{-1}\}] = B \pi^T \quad (13)$$

for some  $B \in \mathbb{R}^{2N_P-5}$  that belongs to the *intersection* of the subspaces generated by the columns of  $N_s H_x$ ,  $N_s H_y$  and  $N_s H_z$ .

After eliminating  $B$  from (13) we obtain:

$$\begin{bmatrix} N_s H_x & -N_s H_y & 0 \\ N_s H_x & 0 & N_s H_z \end{bmatrix} \begin{bmatrix} \{Z^{-1}\}/\pi_x \\ \{Z^{-1}\}/\pi_y \\ \{Z^{-1}\}/\pi_z \end{bmatrix} = 0. \quad (14)$$

Rather than solving (14) directly, we first eliminate  $N_s$  from the equations. One can show that this reduces the bias in solving (14), since, in the presence of noise, the matrix  $S$  is white. We define  $Y_w = \{Z^{-1}\}/\pi_w$ , for  $w = x, y, z$ . Then, since  $N_s^T N_s$  is a projection matrix, equation (14) is equivalent to:

$$\begin{bmatrix} H_x - H_y & 0 & S & 0 \\ H_x & 0 & H_z & 0 & S \end{bmatrix} \begin{bmatrix} Y_x \\ Y_y \\ Y_z \\ U_1 \\ U_2 \end{bmatrix} = 0. \quad (15)$$

If we disregard the fact that  $Y_x, Y_y, Y_z$  are dependent, we can solve equation (15) linearly. When  $\pi_x \neq 0$ ,  $\pi_y \neq 0$  and  $\pi_z \neq 0$ , unique solutions for  $Y_x, Y_y, Y_z$  are obtained, from which  $\{Z^{-1}\}$  can be uniquely recovered as the leading left singular vector of  $[Y_x, Y_y, Y_z] \in \mathbb{R}^{N_P \times 3}$  using SVD. When at least one of the  $\pi_w$  equals zero, there are two possible solutions for the linear system (15). For



example, if  $\pi_x = \pi_y = 0$ , the two solutions are  $[Y_x; Y_y; Y_z] = [\{Z^{-1}\}; \{Z^{-1}\}; 0]$  or  $[\{Z^{-1}\}; -\{Z^{-1}\}; 0]$ . One can still recover  $\{Z^{-1}\}$  from these solutions.

We improve the above estimate for  $\{Z^{-1}\}$  by first recovering  $B$  from the SVD of  $\mathcal{I}$ , using the previously recovered  $\{Z^{-1}\}$ , and then solving linearly for  $\{Z^{-1}\}$  and  $\pi$  from (13), given  $B$ . Given  $\{Z^{-1}\}$  and  $\pi$  one can improve the initial estimate of  $U$  by solving (5), with  $V \in \mathbb{R}^{3 \times 2}$  obtained from the equation  $\pi^T V = 0$ . Finally, the translations are given by  $T = VU^{-1}M^T$  as usual.

**Algorithm 3 (Intersection algorithm for planar motions)** *Given a set of corresponding image points  $\{\mathbf{x}_p^i\}$ ,  $p = 1, \dots, N_P$ ,  $i = 0, \dots, N_F - 1$ , compute the motion  $(R, T)$  and the depth  $\{Z\}$  as follows:*

1. Initialize  $T = 0$
2. Solve for  $R$  linearly from (1), given  $T$  and  $\{Z\}$ .  $\{Z\}$  is unnecessary if  $T = 0$ .
3. Given  $R$  compute  $D$ . Then compute  $S$  and  $M$  from the SVD of  $HD$ .
  - a) Solve for  $Y_x, Y_y, Y_z$  and  $U$  from (15).
  - b) Solve for  $\{Z\}$  as the leading left singular vector of  $[Y_x Y_y Y_z]$  using SVD.
  - c) Solve for  $B$  from  $\mathcal{I}$  and then solve for  $\pi$  and  $\{Z\}$  from (13).
  - d) Solve for  $U$  and  $V$  from (5), and let  $T = VU^{-1}M^T$ .
4. Goto 2. until  $(R, T, \{Z\})$  converge.

### 3.3 Hybrid Algorithm

Both the multiple  $b$  and the intersection algorithms solve for depth, translation and plane normal in two main stages:

- (a) First depth is obtained from either (10) or (15).
- (b) Then the plane normal  $\pi$  is obtained from the SVD of  $\Pi$  or from (13).

In the first stage, the multiple  $b$  algorithm solves for a unique  $\{Z\}$  and multiple copies of  $U$ , while the intersection algorithm solves for multiple copies of  $\{Z\}$  and a unique  $U$ . In the presence of noise, this causes the multiple  $b$  algorithm to give a very accurate estimate for  $\{Z\}$  and a less accurate estimate for  $U$ , and vice-versa for the intersection algorithm.

In the second stage, the multiple  $b$  algorithm solves for the plane normal and translations based on the estimates of  $U$  and  $\{Z\}$ , while the intersection algorithm solves for the plane normal from the estimates of  $\{Z\}$  only.

This suggests to combine the best part of both algorithms in a hybrid manner as follows: first solve for  $\{Z\}$  using the multiple  $b$  algorithm and then solve for  $\pi$  and  $T$  using the intersection algorithm.

### 3.4 General Properties of the Planar Algorithms

**Convergence.** Since the only modification of the non-planar algorithm is to replace step 3(a) by a method that remains linear, all the planar algorithms that we propose inherit the convergence guarantees of the non-planar algorithm.

**Detection of planar motion.** We detect when the motion is roughly planar, and thus when to apply our new algorithms, from the singular values  $s_1$ ,  $s_2$  and  $s_3$  of  $HD$ : if  $s_3/s_2 \geq \epsilon$ , where  $\epsilon$  is a pre-fixed threshold, we use the non-planar motion algorithm; if  $s_1/s_2 < \epsilon$  we use the algorithm for linear motion in [7].

**Non-planar motion.** Note that our planar algorithm works even on fully non-planar motions: in the non-planar case, restricting to two of the singular vectors of  $HD$  does not corrupt the reconstruction but simply omits the additional information from the third singular vector [9].

## 4 Experimental Results

In this section, we evaluate the proposed algorithms on synthetic and real images. We compare our results with those of the following two algorithms:

1. *Linear motion algorithm*: the algorithm of [7] first estimates the rotations linearly assuming that the translations are zero. It computes the SVD of a matrix that depends on the image displacements and the rotation estimates. For each of the first three singular values of this matrix that is above a pre-fixed noise threshold, it recovers a translation direction from the corresponding singular vector, using a linear algorithm similar to that of [3] for optical flow. It refines the estimates of the translation directions by minimizing appropriate error functions. It linearly computes the depths from the recovered translation directions and the leading singular vectors, and it finally computes the translation magnitudes. This algorithm has been shown in [9] to give better results than the Sturm/Triggs algorithm [11] for arbitrary small motions. In our experiments, we always apply this algorithm under the assumption of planar motion, *i.e.*, with the fixed threshold set so that just the two largest singular values are used.
2. *Optimal re-projection error*: motion parameters  $(R, T)$  and 3D structure  $X$  are obtained by minimizing the function:

$$F(R, T, X) = \sum_{p=1}^{N_P} \sum_{i=0}^{N_F-1} \left\| \mathbf{x}_p^i - \frac{R^i(X_p - T^i)}{(R^i(X_p - T^i))_z} \right\|^2$$

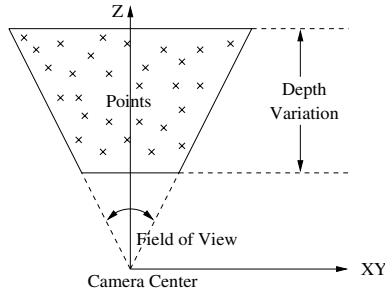
starting from the ground truth. The depth of each point is  $Z_p = (X_p)_z$ .

In our comparison, we use the following error measures for the different parameters, averaged over the number of trials (and frames if appropriate):

$$\begin{aligned} \text{Rotation error} &= \text{acos}((\text{trace}(R_{true} R_{est}^T) - 1)/2) \\ \text{Translation error} &= \text{acos}(T_{true}^T T_{est}) / (\|T_{true}\| \|T_{est}\|) \\ \text{Depth error} &= \text{acos}(Z_{true}^T Z_{est}) / (\|Z_{true}\| \|Z_{est}\|) \\ \text{Normal error} &= \text{acos}(\pi_{true}^T \pi_{est}) / (\|\pi_{true}\| \|\pi_{est}\|). \end{aligned}$$

**Table 1.** Simulation parameters.

Parameter	Unit	Value
Number of trials		1000
Number of points		20
Number of frames		8
Field of view	degrees	90
Depth variation	u.f.l.	100 - 400
Image size	pixels	$500 \times 500$
$\tau = T_{\max}/Z_{\min}$		0.1-0.6

**Fig. 2.** Truncated pyramid used to generate the structure.

#### 4.1 Experiments on Synthetic Images

In our simulations, we varied the motion, plane normal, structure and image noise randomly at each trial. We generated the structure by randomly picking each coordinate of the 3D point according to a uniform distribution taken from a truncated pyramid specified by the depth variation and the field of view, as shown in Fig. 2. Simulation parameters are shown in Table 1<sup>2</sup>.

**Error vs. noise.** Figures 3 and 4 compare the performance of the different algorithms for different levels of noise<sup>3</sup>. For small  $\tau$ , we observe that the new algorithms significantly outperform the *linear motion algorithm*. The best algorithm for rotation is the *multiple b algorithm*, and the best one for translation, depth and normal is the *hybrid algorithm*. The new algorithms give very accurate estimates for rotation, translation and normal to the plane of motion (almost indistinguishable from the optimal). Depth estimates are suboptimal due to the bas-relief ambiguity<sup>4</sup>. As  $\tau$  increases, so does the error in the noise free case, which decreases the accuracy of our algorithms with respect to the optimal. This is expected, since the approximation  $T_{z,\max}/Z_{\min} \approx 0$  is no longer valid. Notice that the slope of the error decreases with  $\tau$ .

Table 2 includes the number of outliers for each algorithm. We define a trial to be an outlier if the error for that trial in any of the motion or structure param-

<sup>2</sup> In the table u.f.l. stands for units of focal length.

<sup>3</sup> Mean errors do not include outliers.

<sup>4</sup> [8] describes methods for repairing a similar problem in depth recovery observed for the non-planar motion algorithm, which could also be applied here.

eters exceeds the mean error by 8 times the standard deviation. The *intersection algorithm* has between 1% and 3% outliers, while the *multiple b algorithm* has less than 0.4%. As explained in Section 3.3, this is because the initial estimation of depth from (15) is not as robust as that of the *multiple b algorithm* from (10). The *hybrid algorithm* has less than 0.3% outliers.

Overall, the *hybrid algorithm* is the one with the best performance: it is more accurate and has fewer outliers.

**Error vs.  $\tau$ .** Figure 5 compares the performance of the proposed algorithms for different values of  $\tau$ , for a noise level of 1 pixel. We observe that the proposed algorithms have very good performance in the range  $0.1 \leq \tau \leq 0.6$ , with the best performance for  $\tau \approx 0.3$ . When  $\tau < 0.1$ , the signal-to-noise ratio is too

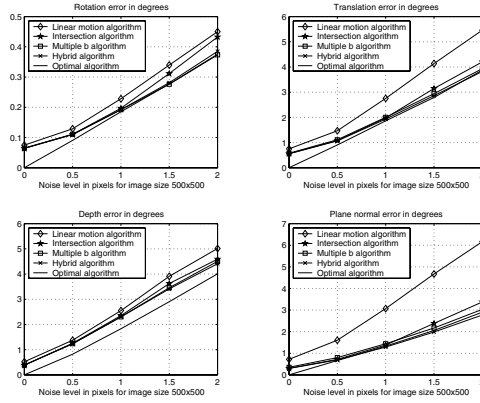


Fig. 3. Error vs. noise for  $\tau = T_{\max}/Z_{\min} \in (0.1, 0.2)$ .

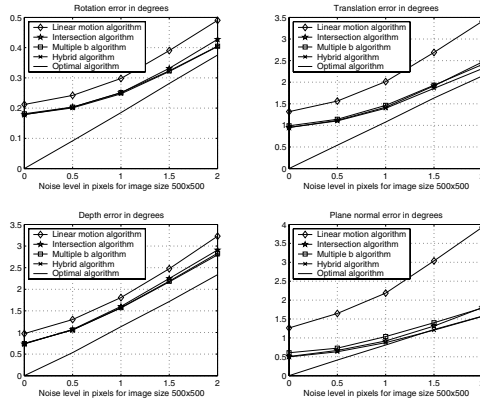


Fig. 4. Error vs. noise for  $\tau = T_{\max}/Z_{\min} \in (0.2, 0.3)$ .

**Table 2.** Number of outliers for each algorithm as a function of noise.

Algorithm	$\tau$	Number of outliers				
		Noise level in pixels				
		0.0	0.5	1.0	1.5	2.0
Linear Motion	0.1–0.2	0	1	2	2	9
Intersection	0.1–0.2	0	11	18	20	28
Multiple $b$	0.1–0.2	0	0	0	1	4
Hybrid	0.1–0.2	0	0	0	1	3
Linear Motion	0.2–0.3	0	1	0	1	4
Intersection	0.2–0.3	7	6	5	19	18
Multiple $b$	0.2–0.3	0	0	0	0	1
Hybrid	0.2–0.3	0	0	0	1	1
Linear Motion	0.3–0.4	0	1	0	0	3
Intersection	0.3–0.4	7	4	9	17	19
Multiple $b$	0.3–0.4	0	0	1	0	1
Hybrid	0.3–0.4	0	1	0	1	1

small, causing an increase of both the error and the number of outliers<sup>5</sup>. When  $\tau > 0.6$  the small baseline assumption is violated, and hence the mean error increases. Notice that we still get good results for  $\tau = 0.6$ , which corresponds to a relatively large translational motion.

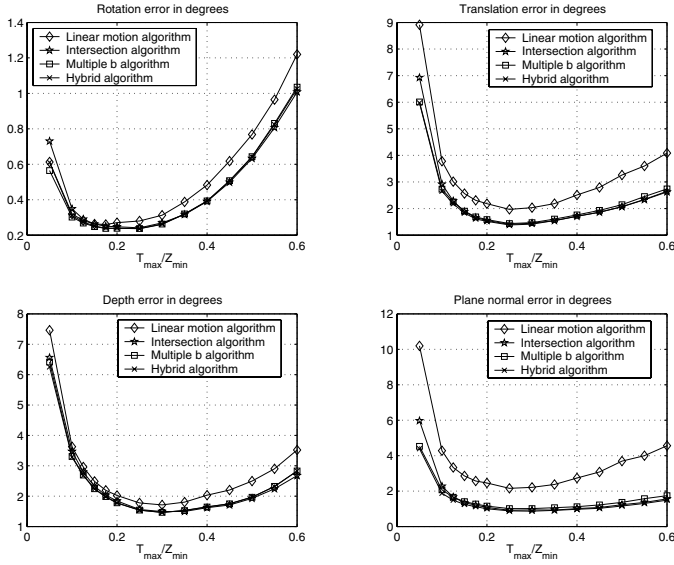
**Efficiency.** Table 3 shows the average execution time on a Pentium III 800 MHz for a MATLAB implementation of each algorithm. The average is taken over 1000 trials, 8 frames and 20 points. We can observe that the fastest of the new algorithms is the *hybrid algorithm*, which is approximately 130 times faster than the optimal algorithm minimizing the re-projection error. Since the optimal algorithm is initialized from the ground truth, the ratio could be higher in practice.

**Table 3.** Execution time of each algorithm for  $0.1 \leq \tau \leq 0.2$ .

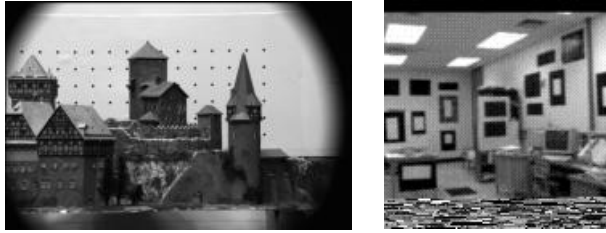
	Linear Motion	Intersection	Multiple $b$	Hybrid	Optimal
Time (sec)	0.11	0.21	0.44	0.20	25.75

---

<sup>5</sup> A generalization of the discussion in [8,9] implies that, when the signal-to-noise ratio is small and the motion is partly forward, one can get  $s_1 \gg s_2 \sim s_3$ , *i.e.*, the translational motion can be effectively linear rather than planar. In this situation, one may get better results by applying our algorithms under the assumption of linear rather than planar motion, as [9] verified experimentally for an analogous case.



**Fig. 5.** Error vs.  $\tau = T_{\max}/Z_{\min}$  for a noise level of 1 pixel.



**Fig. 6.** First frames of the castle and puma sequences.

## 4.2 Experiments on Real Images

We tested our algorithms on the *castle* [1] and *puma* [4] sequences. For the castle sequence we used the first 7 frames only so that the motion is approximately planar. The singular values of the matrix  $HD$  for the castle sequence are  $s_1 = 125.0480$ ,  $s_2 = 39.3361$  and  $s_3 = 0.0301$  and the largest baseline is  $\tau = 0.0443$ . For the puma sequence we considered the first 16 frames, for which the motion is approximately linear, and added some frames so that the motion is approximately planar. The resulting singular values are  $s_1 = 5.5359$ ,  $s_2 = 3.7313$  and  $s_3 = 0.0714$ , and the largest baseline is  $\tau = 0.1090$ . The first frames of the sequences are shown in Figure 6.

Figure 7 compares the performance of the proposed algorithms for the castle sequence. Again, we observe that the new algorithms outperform the *linear motion algorithm*. Further, motion and structure parameters are estimated very

accurately (within 0.5 degrees). The algorithm with best performance is the *hybrid algorithm*, except for translation for which the best algorithm is the *multiple b algorithm*. For the puma sequence, the *multiple b algorithm* is the best for rotation and depth, and the *intersection algorithm* is the best for translation and normal. This result is not surprising, since theoretically the best performance of the intersection algorithm is for  $\pi \approx [0; 0; 1]$ . Notice that all the algorithms worked well on this sequence though it is slightly non-planar.

## 5 Conclusions

We presented a set of linear algorithms for motion and structure estimation when the camera translates on a plane with small baselines and arbitrary rotations. Our algorithms are based on an approximate rank-2 factorization of a matrix which depends on the image displacements and current rotation estimates.

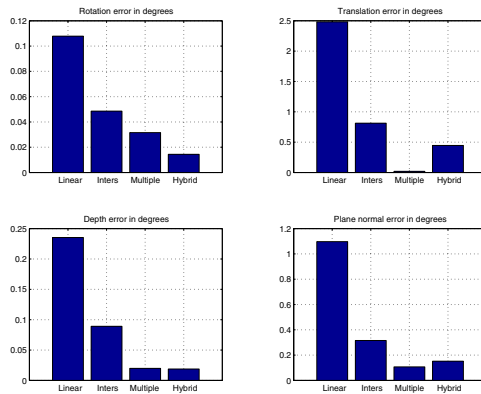


Fig. 7. Mean error of each algorithm for the castle sequence.

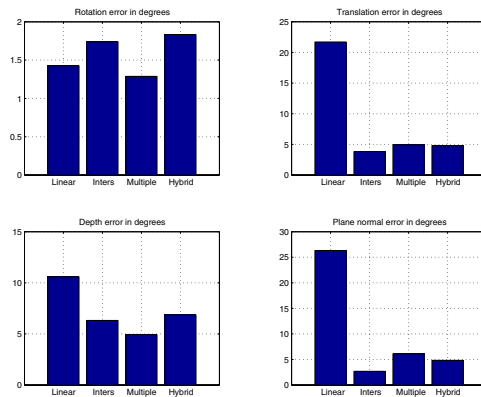


Fig. 8. Mean error of each algorithm for the puma sequence.

Translation and depth are obtained from the factorization after an appropriate parameterization of the plane of translation.

We tested our algorithms on both synthetic and real sequences. Experimental results show that the proposed algorithms are able to compute the structure and motion parameters accurately and efficiently for baselines in the range  $0.1 \leq \tau \leq 0.6$ . The proposed algorithms have good convergence properties and the best algorithm presents less than 0.3% outliers.

## References

1. CMU CIL-0001 castle image sequence. <http://www.cs.cmu.edu/~cil/cil-ster.html>.
2. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.
3. D. Heeger and A. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.
4. R. Kumar and A. Hanson. Sensitivity of the pose refinement problem to accurate estimation of camera parameters. In *IEEE International Conference on Computer Vision*, pages 365–369, 1990.
5. P. McLauchlan and D. Murray. A unifying framework for structure and motion recovery from image sequences. In *International Conference on Computer Vision and Pattern Recognition*, pages 314–20, 1995.
6. J. Oliensis. Rigorous bounds for two-frame structure from motion. In *European Conference on Computer Vision*, pages 184–195, 1996.
7. J. Oliensis. Structure from linear or planar motions. In *International Conference on Computer Vision and Pattern Recognition*, pages 335–42, 1996.
8. J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *International Journal of Computer Vision*, 34(2-3):163–192, 1999.
9. J. Oliensis and Y. Genc. Fast and accurate algorithms for projective multi-image structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):546–559, 2001.
10. S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996.
11. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision*, pages 709–720, 1996.
12. R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.
13. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992.
14. B. Triggs. Factorization methods for projective structure and motion. In *International Conference on Computer Vision and Pattern Recognition*, pages 845–51, 1996.



# Revisiting Single-View Shape Tensors: Theory and Applications

Anat Levin and Amnon Shashua

School of Computer Science and Engineering,  
The Hebrew University,  
Jerusalem 91904, Israel  
{shashua,alevin}@cs.huji.ac.il

**Abstract.** Given the projection of a sufficient number of points it is possible to algebraically eliminate the camera parameters and obtain view-invariant functions of image coordinates and space coordinates. These single view invariants have been introduced in the past, however, they are not as well understood as their dual multi-view tensors. In this paper we revisit the dual tensors (bilinear, trilinear and quadlinear), both the general and the reference-plane reduced version, and describe the complete set of synthetic constraints, properties of the tensor slices, reprojection equations, non-linear constraints and reconstruction formulas. We then apply some of the new results, such as the dual reprojection equations, for multi-view point tracking under occlusions.

## 1 Introduction

There is a large body of research on multi-view geometry of 3D scenes which has culminated to the point where the issues and solutions are well understood. The body of work on multi-view geometry centers around matching tensors of 2,3,4 views known as multi-view constraints (bifocal, trifocal and quadrifocal tensors) which are borne out of algebraic elimination of the scene geometry (shape) from the 3D-to-2D projection equations given a sufficient number of views [16,20,9,5,7,10,17,1]. The “dual” form of the elimination process is to eliminate the camera parameters (motion) given a sufficient number of points in a single view with the result of what is known as *single-view shape tensors* [2,21,3] and in a reduced setting where a reference plane is identified in advance is called *parallax geometry* [11,12,4,18].

Multi-view geometry has been put into practice in a variety of applications including 3D reconstruction, novel view synthesis, camera ego-motion, augmented reality and visual recognition by alignment. The multi-point geometry, on the other hand, has been hardly put into use although the topic makes a very appealing case for applications. In many instances, one would like to achieve a direct representation of 3D shape from images without the need to recover the camera geometry as an intermediate step. This includes indexing into a library of objects (cf. [13]), multi-body segmentation (collection of points belong to the same structure when the shape invariants hold), and even for tracking applications

(which traditionally use multi-view constraints) where features may get lost due to occlusions and later reappear. A direct shape constraint is advantageous due to the local image support needed to make it work.

In this paper we focus on open issues which remain with shape tensors, such as the number and nature of the *synthetic* constraints, properties of tensor slices, reprojection equations, a full account of the non-linear constraints and reconstruction formulas. We then apply some of the new results, such as the dual reprojection equations, for multi-view point tracking under occlusions.

We will start with a brief description of what is known about these tensors which will create the context for describing in more details our contributions to this topic.

### 1.1 What Is Known To-Date about Shape Tensors

The basic idea is that points and cameras can be switched (duality principle) and as a result one can obtain exactly the same multi-linear constraints as in the multi-view derivations, where instead of multiple views we have multiple points. Let  $P_i = (X_i, Y_i, Z_i, W_i)^\top \in \mathcal{P}^3$  denote points in 3D projective space and let  $M$  be a  $3 \times 4$  projection matrix, thus  $p_i \cong MP_i$  where  $p_i \in \mathcal{P}^2$  be the corresponding image points in the 2D projective plane. We wish to algebraically eliminate the camera parameters (matrix  $M$ ) by having a sufficient number of points. This could be done elegantly, and along the way obtain the duality principle, if we first change basis as follows. Let the first 4 points  $P_1, \dots, P_4$  be assigned  $(1, 0, 0, 0), \dots, (0, 0, 0, 1)$  and let the image undergo a projective change of coordinates such that the corresponding points  $p_1, \dots, p_4$  be assigned  $e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1), e_4 = (1, 1, 1)$ , respectively. Given this setup the camera matrix  $M$  contains only 4 non-vanishing entries:

$$M = \begin{bmatrix} \alpha & 0 & 0 & \delta \\ 0 & \beta & 0 & \delta \\ 0 & 0 & \gamma & \delta \end{bmatrix}$$

Let  $\hat{M} = (\alpha, \beta, \gamma, \delta) \in \mathcal{P}^3$  be a point (representing the camera) and let  $\hat{P}_i$  be the projection matrix:

$$\hat{P}_i = \begin{bmatrix} X_i & 0 & 0 & W_i \\ 0 & Y_i & 0 & W_i \\ 0 & 0 & Z_i & W_i \end{bmatrix}$$

And we have the duality  $p_i \cong MP_i = \hat{P}_i \hat{M}$  where the role of the motion (the camera) and shape have been switched. At this point we follow exactly the same steps one does with the multi-view tensors: let  $l_i, l'_i$  be two distinct lines passing through the image point  $p_i$ , i.e.,  $p_i^\top l_i = 0$  and  $p_i^\top l'_i = 0$ , and therefore we have  $l_i^\top \hat{P}_i \hat{M} = 0$  and  $l'^\top_i \hat{P}_i \hat{M} = 0$ . For  $i = 5, \dots, 8$  we have therefore  $E \hat{M} = 0$  where:

$$E = \begin{bmatrix} l_5^\top \hat{P}_5 \\ \cdot \\ l_8^\top \hat{P}_8 \\ l_5'^\top \hat{P}_5 \\ \cdot \\ l_8'^\top \hat{P}_8 \end{bmatrix} \quad (1)$$

Therefore the determinant of any 4 rows of  $E$  must vanish. The choice of the 4 rows can include 2 points, 3 points, or 4 points (on top of the 4 basis points  $P_1, \dots, P_4$ ) and each such choice determines a multilinear constraint whose coefficients are arranged in a tensor. For the case of (4+) 2 points, say points  $p_5, p_6$ , there is only one such tensor with the bilinear constraint  $p_6^\top \mathcal{F} p_5 = 0$  where the  $3 \times 3$  matrix  $\mathcal{F}$  contains the shape parameters of  $P_5, P_6$ . In the work of [2,21,3] the properties of  $\mathcal{F}$  were derived (there are 4 linear constraints and  $\text{rank}(\mathcal{F})=2$ ). The cases of 7 and 8 points were less understood. Clearly, in the case of 7 points, there are three tensors where we choose two rows of a “reference” point (say  $P_5$ ) and one row from the remaining two points ( $P_6, P_7$ ). The determinant expansion provides a trilinear constraint of the form  $p_5^i l_j^6 l_k^7 \mathcal{T}_i^{jk} = 0$  where  $p_5$  is the reference point,  $l^6, l^7$  are lines through the points  $p_6, p_7$  respectively, and the indices  $i, j, k$  follow the covariant-contravariant notations (upper index represents points, lower represent lines) and follow the summation convention (contraction)  $u^i v_i = u^1 v_1 + u^2 v_2 + \dots + u^n v_n$ . Note that since the tensor is contracted by a point ( $p_5$ ) and a *choice* of line through the remaining two points, then each view contributes 4 linear constraints on the 27 unknowns of the tensor.

At this point the literature becomes incomplete — clearly, we expect three views to be sufficient for recovering the tensor (because of duality with the multi-view trilinear tensor) thus the coefficients of the tensor must satisfy internal linear constraints. In [2,21,3] the way around this was to find out using Grobner basis with computer algebra tools that there are only 11 parameters (up to scale) which form 4 trilinear equations. The number of parameters is indeed 11 (as we will see later), but the tensor has been lost in all of this. In [8], in attempt to summarize the topic, have noticed that there are internal linear constraints, which they called “synthetic constraints” (which we will touch upon later). However, they did not provide the exact number of such constraints (which is indeed 16, leaving 11 parameters up to scale, as we shall see later). Moreover, the following issues remained open: (i) non-linear constraints on the tensor (there should be 4), (ii) tensor slices (from which we obtain “reprojection equation”, homography slices, dual epipoles, etc.), and (iii) reconstruction of shape from the tensor slices’ properties (dual epipole, dual homography).

The case of 8 points is open to a large extent. This case is dual to the quadrfocal multi-view tensor, thus by choosing one row from each point we obtain a vanishing determinant involving 4 points which provides 16 constraints (per view)  $l_i^5 l_j^6 l_k^7 l_t^8 \mathcal{Q}^{ijkl} = 0$  for the 81 coefficients of the tensor  $\mathcal{Q}^{ijkl}$ . Again, one expects two views to suffice, therefore the quadlinear tensor must contain many internal linear constraints. In [3] through the use of Grobner basis with computer algebra tools it was found that there are 22 quadlinear constraints (per view) with 41 coefficients, thus two views would suffice. It is unclear where this result comes from, in fact (as we will show later) the quadlinear tensor has 81 coefficients (just like its dual brother in the multi-view case) but there are 58 synthetic linear constraints — therefore we have 23 parameters up to scale. The first view provides 12 constraints, and the second view provides 11 constraints

(thus two views are sufficient). As for non-linear constraints, there are 13 of them.

Next, consider the case in which a plane has been identified in advance and has been “stabilized” across the sequence of views. This reduced setting has been coined “parallax geometry” and has the advantage of making a clear geometric picture of the basic building block of the dual geometry [11,12,4]. Existing work focus on the geometric interpretation (the dual epipole) and in [12] the trilinear constraint was derived geometrically (and has been shown to be bilinear in this setting).

We will show that, beyond the geometrical interpretation, the real advantages lies elsewhere. First, this setting corresponds to having the first 4 points  $P_1, \dots, P_4$  to have the coordinates  $(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (1, 1, 1, 0)$  which is appropriate when  $P_1, \dots, P_4$  are indeed coplanar. The tensors are the same, what changes is the number of synthetic constraints — for the bilinear tensor we have 5 constraints, for the trilinear tensor we have 21 constraints, and for the quadlinear tensor 72 constraints (the latter requires tools from representation theory). More importantly, these tensors do not have any non-linear constraints — thus making them appealing in practice. The lack of non-linear constraints is at the heart of the recent result in [18] showing that factorization is possible in this context.

Finally, we have conducted real imagery experiments which highlight the use of some of the new discoveries — such as the reprojection equations — and which covers a number of tracking applications and multi-body motion segmentation.

## 2 Synthetic Constraints in the General Case

The multi-point tensors are derived from the vanishing  $4 \times 4$  determinants of  $E$  (eqn. 1). Because of duality, we obtain exactly the same tensorial forms as in the multi-view case. The difference is that the projection matrices  $\hat{P}_i$  are sparse and as a result one obtains additional constraints, which following [8] we will call *synthetic* constraints, which we will now analyze.

Consider the camera projection matrix  $M_j$  be constructed such that the  $j$ 'th column is  $e_j$  (the  $j$ 'th standard basis vector) and the remaining entries vanish. We have then  $M_j P$  is either  $e_j$  or vanishes for all choices of  $P$ . Let  $l_i, l'_i$ ,  $i = 5, \dots, 8$ , be lines through  $e_j$ , therefore

$$\begin{aligned} l_i^\top M_j P &= l_i^\top \hat{P} \hat{M}_j = 0 \\ l'_i{}^\top M_j P &= l'_i{}^\top \hat{P} \hat{M}_j = 0 \end{aligned}$$

for all points  $P$ , and dually for all projection matrices  $\hat{P}$ . Therefore the  $4 \times 4$  determinants of  $E$  vanish regardless of  $\hat{P}_i$ . For example, in the case of 6 points (choose two rows from  $p_5$  and two rows from  $p_6$ ) we obtain  $e_j^\top \mathcal{F} e_j = 0$ ,  $j = 1, \dots, 4$ . Therefore, we have 4 synthetic constraints on  $\mathcal{F}$ , i.e., the 6-point tensor is represented by  $9 - 4 = 5$  parameters up to scale as already pointed out in [2, 21,3].

In the case of 7 points, say  $p_5$  is the reference point, thus we have the multilinear constraint  $p_5^i l_j^6 l_k^7 \mathcal{T}_i^{jk} = 0$  where  $l^6, l^7$  are lines through the points  $p_6, p_7$  respectively. Let  $l_j, l'_k$  be a line through  $e = e_j$ , then from the above we have that  $e^i l_j l'_k \mathcal{T}_i^{jk} = 0$  which provides 4 constraints (because there are two choices for lines  $l_j$  and two choices for lines  $l'_k$ ). Therefore we have 16 synthetic constraints (because  $e$  ranges over  $e_1 = (1, 0, 0), \dots, e_4 = (1, 1, 1)$ ). We have arrived to the result:

*Claim.* In the case of 7 points, each of the three  $3 \times 3 \times 3$  trilinear tensors contract on a point (the reference point) and two lines coincident with the remaining two points. The choice of the reference point determines the tensor in question. Each of these tensors has 16 internal linear constraints, thus leaving 11 parameters up to scale. Each view contributes 4 linear constraints on the tensor in question, thus 3 views are necessary for a linear solution.

The number of parameters a 7-point configuration carries is  $3+3 = 6$  (because  $P_5$  can be set arbitrarily, say  $P_5 = (1, 1, 1)$  and each additional point carries 3 parameters). *We therefore expect 4 non-linear constraints on each of the tensors.* We will return to this issue later after we study the tensor slices.

In the case of 8 points, we have a single  $3 \times 3 \times 3 \times 3$  tensor  $\mathcal{Q}^{ijkl}$  responsible for the 16 quadlinear constraints  $l_i^5 l_j^6 l_k^7 l_t^8 \mathcal{Q}^{ijkl} = 0$  (we have a choice of 2 lines for each point, thus 16 constraints). From the discussion above, if all the lines are coincident with  $e = e_j$  the constraint holds for all quadlinear tensors (i.e., apply to all space points). Therefore, for  $e = e_1$  we have 16 synthetic constraints. For  $e = e_2$  we will have 15 constraints because the line between  $e_1$  and  $e_2$  is already covered by the previous 16 constraints. Likewise, each additional point provides one less constraint, thus we have a total of  $16 + 15 + 14 + 13 = 58$  synthetic constraints. The first view will contribute 12 constraints (the lines through  $p_5, \dots, p_8$  passing through  $e_j$  are already spanned by the synthetic constraints), and the second view will contribute 11 constraints (because the lines through the four points in view 1 and the four points in view 2 are spanned by the 12 constraints from view 1). Therefore, we have  $58 + 12 + 11 = 81$  which provides sufficient constraints to solve for the quadlinear tensor. To summarize:

*Claim.* In the case of 8 points, there is a single quadlinear tensor of size  $3^4 = 81$ . The tensor has 58 linear constraints, thus is defined by 23 parameters up to scale. Each view contributes 16 linear constraints on the tensor of which 12 are independent for the first view and 11 are independent for the second view.

An 8-point configuration is determined by  $3 + 3 + 3 = 9$  parameters, thus we expect 13 non-linear constraints.

### 3 Synthetic Constraints with a Reference Plane

When a plane is identified in advance and stabilized we find a different set of synthetic constraints. The first 4 basis points  $P_1, \dots, P_4$  are assigned the coordinates

$(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (1, 1, 1, 0)$  which is appropriate when  $P_1, \dots, P_4$  are indeed coplanar. The resulting camera matrix becomes:

$$M = \begin{bmatrix} \delta & 0 & 0 & \alpha \\ 0 & \delta & 0 & \beta \\ 0 & 0 & \delta & \gamma \end{bmatrix}$$

and the resulting projection matrix  $\hat{P}_i$  becomes:

$$\hat{P} = \begin{bmatrix} W_i & 0 & 0 & X_i \\ 0 & W_i & 0 & Y_i \\ 0 & 0 & W_i & Z_i \end{bmatrix}$$

And as in the general case we have the duality  $p_i \cong MP_i = \hat{P}_i \hat{M}$  where the role of the motion (the camera) and shape have been switched. The matrix  $E$  is identical to eqn. 1 thus the tensors are exactly the same. What is different is the number of synthetic constraints (and also geometric interpretation which we will address later).

Since  $P_1, \dots, P_4$  are coplanar we have the constraint  $P_i^\top n = 0$ ,  $i = 1, \dots, 4$  and, due to our choice of coordinates,  $n = (0, 0, 0, 1)^\top$ . Consider the family of camera matrices  $M = un^\top$  for all choices of  $u = (u_1, u_2, u_3)$ . In other words, the 4'th column of  $M$  consists of the arbitrary vector  $u$  and all other entries vanish. Thus we have that  $MP$  either vanishes or is equal to  $u$  (up to scale) for all  $P$ . Let  $l_i, l'_i$  be lines through  $u$ , therefore

$$\begin{aligned} l_i^\top M_j P &= l_i^\top \hat{P} \hat{M}_j = 0 \\ l'_i{}^\top M_j P &= l'_i{}^\top \hat{P} \hat{M}_j = 0 \end{aligned}$$

for all points  $P$ , and dually for all projection matrices  $\hat{P}$ . Therefore the  $4 \times 4$  determinants of  $E$  vanish regardless of  $\hat{P}_i$ . For example, in the case of 6 points (choose two rows from  $p_5$  and two rows from  $p_6$ ) we obtain  $u^\top \mathcal{F} u = 0$  for all choices of  $u$ , thus the matrix  $\mathcal{F}$  is skew-symmetric and in turn is defined by 3 parameters (as already pointed out in [11,12,4]).

In the case of 7 points, say  $p_5$  is the reference point, thus we have the multi-linear constraint  $p_5^i l_6^j l_7^k \mathcal{T}_i^{jk} = 0$  where  $l_6, l_7$  are lines through the points  $p_6, p_7$  respectively. Let  $l_j, l'_k$  be a line through  $u$ , then from the above we have that  $u^i l_j l'_k \mathcal{T}_i^{jk} = 0$  for all choices of  $u$  where  $l^\top u = 0$  and  $l'^\top u = 0$ . The number of synthetic constraints is the dimension of  $\mathcal{T}_i^{jk}$ . The issue of dimension is dual to the issue of recovering the multi-view tensor from a coplanar scene. Let  $p, p', p''$  be matching points across 3 views and let  $A, B$  be the homography matrices  $p' \cong Ap$  and  $p'' \cong Bp$  (since the scene is coplanar). Let  $s, r$  be lines through  $p', p''$  respectively, thus the measurements for the multi-view trilinear tensor come from  $p^i s_j r_k \mathcal{T}_i^{jk} = 0$  for all choices of  $p$  while  $s^\top (Ap) = 0$  and  $r^\top (Bp) = 0$ . Since the choice of  $A, B$  does not affect the issue of dimension, we may as well set  $A = B = I$  and we have exactly the same situation as in the multi-point tensor described above. The issue of dimension for planar configuration for the multi-view trilinear tensor was resolved in [19] with the answer of 21. As a result, we can conclude that there are 21 synthetic constraints for the multi-point trilinear

tensor. Moreover, the remaining parameters  $27 - 21 - 1 = 5$  is exactly what is required to describe a configuration of 7 points, 4 of which are coplanar:  $P_6$  adds only 2 parameters (because the first 5 points do not provide a full projective basis because the 4'th point is spanned by the first three, thus carries only 2 parameters, therefore they provide only 14 degrees of freedom instead of 15), and  $P_7$  adds the usual 3 parameters. Note that since we are left with 6 parameters (up to scale) and each view contributes 4 linear equations on the tensor, only 2 views are necessary (instead of 3 which was required in the general case). Any of the two views adds only 3 constraints, instead of the expected 4, since it's possible to choose:  $l^6 = p_6 \times p_5$  and  $l^7 = p_7 \times p_5$ . Then  $p_5, l^6, l^7$  is a configuration of points and two lines through the point, those expressed already by the synthetic constraints. We summarize:

*Claim.* In the case of 7 points where the first 4 are coplanar, each of the three trilinear tensors has 21 internal linear constraints of the form  $u^i l_j l'_k \mathcal{T}_i^{jk} = 0$  for all choices of  $u$  where the lines  $l, l'$  are coincident with  $u$ . These constraints are all the constraints on the tensor, there are no other non-linear constraints. Finally, only two views are required in order to solve for the tensor.

In the case of 8 points, the dimension analysis is more subtle and requires different tools. The quadlinear tensor is defined exactly as in the general case: we have a single  $3 \times 3 \times 3 \times 3$  tensor  $\mathcal{Q}^{ijkl}$  responsible for the 16 quadlinear constraints  $l_i^5 l_j^6 l_k^7 l_t^8 \mathcal{Q}^{ijkl} = 0$  (we have a choice of 2 lines for each point, thus 16 constraints). From the discussion above, the four lines contracted by the tensor are all coincident with the arbitrary point  $u$ . Therefore, the question is what is the dimension of the set of constraints  $l_i^5 l_j^6 l_k^7 l_t^8 \mathcal{Q}^{ijkl} = 0$  where the lines are arbitrary but form a 2-dimensional subspace? Let

$$V = \{v_1 \otimes v_2 \otimes v_3 \otimes v_4 \mid \dim \text{Span}\{v_1, \dots, v_4\} \leq 2\}$$

where  $v_1, \dots, v_4$  are vectors in  $R^3$ . Our question regarding the number of synthetic constraints is equivalent to the question of *what is the dimensions of  $V$ ?* The answer is 72 and is derived as follows.

Let  $\lambda = (\lambda_1, \dots, \lambda_4)$ , be a partition of 4, i.e.,  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$  and  $\sum_i \lambda_i = 4$ . A diagram associated with  $\lambda$  has 4 rows of left-aligned boxes with  $\lambda_i$  boxes in row  $i$ . Let  $f_\lambda$  be the number of ways to fill the diagram of  $\lambda$  with the numbers from 1 to 4, such that all rows and columns are increasing. Let  $(i, j)$  denote the coordinates of the boxes of the diagram where  $i = 1, \dots, 4$  denotes the row number and  $j$  denotes the column number, i.e.,  $j = 1, \dots, \lambda_i$  in the  $i$ 'th row. The *hook length*  $h_{ij}$  of a box at position  $(i, j)$  in the diagram is the number of boxes directly below plus the number of boxes to the right plus 1. Then,

$$f_\lambda = \frac{m!}{\prod_{(i,j)} h_{ij}}$$

where the product of the hook-lengths is over all boxes of the diagram. Let  $d_\lambda$  denote the number of ways to fill the diagram with the numbers from 1 to 3,

such that all rows are non-decreasing and all columns are increasing. We have:

$$d_\lambda = \prod_{(i,j)} \frac{3-i+j}{h_{ij}}.$$

The definitions above can be found in [6]. With this in mind we have (proof is omitted) that:

$$\dim V = \sum_{\lambda_3=\lambda_4=0} f_\lambda d_\lambda.$$

We have therefore only three partitions:  $\lambda = (4), (2, 2), (3, 1)$  to consider. Thus,  $f_{(4)} = 1, d_{(4)} = 15, f_{(2,2)} = 2, d_{(2,2)} = 6, f_{(3,1)} = 3$  and  $d_{(3,1)} = 15$ . Therefore,  $\dim V = 15 + 12 + 45 = 72$ .

Note that there no more non-linear constraints because we are left with  $81 - 1 - 72 = 8$  parameters which is exactly the number required to represent a configuration of 8 points in which the first 4 are coplanar:  $P_6$  contributes 2 parameters,  $P_7, P_8$  contribute 3 parameters each. Finally, although we are left with  $81 - 72 - 1 = 8$  parameters, we still need two views: the first contributes 5 constraints and the second 4 constraints (proof omitted). We summarize:

*Claim.* In the case of 8 points where the first 4 are coplanar, the quadlinear tensor has 72 internal linear constraints. These constraints are all the constraints on the tensor, there are no other non-linear constraints.

## 4 Tensor Slices and Properties

The dual tensors are borne out of exactly the same multi-linear forms as the multi-view tensors — the differences lie in the fact that the projection matrices  $\hat{P}_i$  are sparse and thus additional constraints are imposed (like the synthetic constraints described above). Moreover, because the multilinear forms are the same we should expect to have a “dual” of each of the basic elements one encounters in multi-view analysis: image ray, epipolar line, epipoles and homography matrix. These duals exist both in the general case and in the special case of stabilized reference plane. The duals of the homography matrices are the most important because they are the key for obtaining the source of the non-linear constraints for the trilinear tensor (as pointed out in [1] for the multi-view trilinear tensor).

We will derive the dual elements, the reprojection equation from the trilinear tensor, the homography slices of the trilinear tensor, the non-linear constraints from the homography slices, the breakdown of the trilinear tensor into a epipole-homography structure, and reconstruction of space points. We will switch back and forth between the general case and the case of stabilized reference plane (which we will refer to as the “reduced case”).

We will assume, without proof due to lack of space, that indeed the family of all camera matrices projecting a fixed set of 4 coplanar points from  $\mathcal{P}^3$  to  $\mathcal{P}^2$ , have a common stabilized plane — thus they differ from one another only in the location of the center of projection.



Let  $\text{null}(M)$  be the projection center. Note that in the general case  $\text{null}(M) = (1/\alpha, 1/\beta, 1/\gamma, -1/\delta)^T$ , whereas in the reduced case  $\text{null}(M) = (\alpha, \beta, \gamma, -\delta)^T$ . This indicates something of importance: in the general case, when  $\hat{M} = (\alpha, \beta, \gamma, \delta)^T$  varies along a linear subspace (a line or a plane),  $\text{null}(M)$  varies along an algebraic *surface* (non-linear), whereas in the reduced case,  $\text{null}(M)$  varies along a linear subspace of the same dimension. This is the key for the simple geometric interpretation of the elements (like image ray and epipole) in the reduced case (as introduced in [11]). Nevertheless, all the elements exist and are well defined in the general case as well.

The dual epipoles. In the multi-view context the epipoles are  $M_i \text{null}(M_j)$  which is the projection of the  $j$ 'th camera center onto the  $i$ 'th image plane. Likewise, because of the duality  $MP = \hat{P}\hat{M}$ , the *dual epipole is defined by*  $\hat{P}_i \text{null}(\hat{P}_j)$ . In the case of 6 points bilinear tensor we should have two dual epipoles:  $e_{65} = \hat{P}_6 \text{null}(\hat{P}_5)$  and  $e_{56} = \hat{P}_5 \text{null}(\hat{P}_6)$ .

*Claim.*  $\hat{P}_6 \text{null}(\hat{P}_5)$  is the projection of  $P_6$  via the camera whose center is at  $P_5$ .

**Proof:** Recall that  $\text{null}(\hat{P}_5) = (1/X_5, 1/Y_5, 1/Z_5, -1/W_5)$ . From duality we have:

$$\hat{P}_6(\text{null}(\hat{P}_5)) = \begin{bmatrix} 1/X_5 & 0 & 0 & -1/W_5 \\ 0 & 1/Y_5 & 0 & -1/W_5 \\ 0 & 0 & 1/Z_5 & -1/W_5 \end{bmatrix} P_6$$

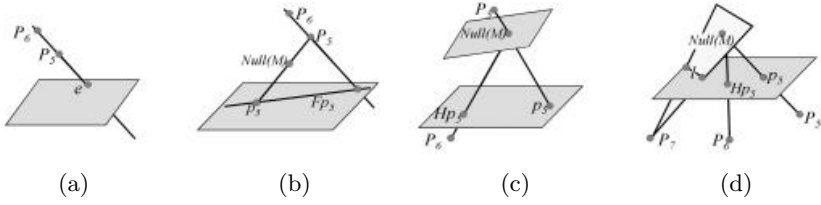
where the camera projection has its center equal to  $P_5$ .  $\square$

*Claim.*  $e_{65} = \text{null}(\mathcal{F})$  and  $e_{56} = \text{null}(\mathcal{F}^\top)$

**Proof:** Consider a camera matrix  $M$  whose null space is  $P_6$ . Such a camera maps  $P_6$  to 0. Therefore, for any point  $p_6 \in \mathcal{P}^2$ , the lines  $l_6, l'_6$  passing through  $p_6$  satisfies:  $l_6^\top MP_6 = 0$  and  $l'_6{}^\top MP_6 = 0$ . Therefore, if we take  $p_5$  to be the projection of  $P_5$  via  $M$ , then  $p_6^\top \mathcal{F} p_5 = 0$ , for all choices of  $p_6 \in \mathcal{P}^2$ . This of course implies that  $\mathcal{F} p_5 = 0$ .  $\square$

In the reduced case the two epipoles must coincide since  $\mathcal{F}$  is a skew-symmetric matrix (thus  $\text{null}(\mathcal{F}) = \text{null}(\mathcal{F}^\top)$ ). The epipole  $e = e_{65} = e_{56}$  was coined the “dual epipole” in [11]. Since in the reduced case the image plane is stabilized, the dual epipole is simply the intersection of the line  $\hat{P}_5 \hat{P}_6$  with the stabilized image plane (see Fig. 1a). Note however, that this is true only for the reduced case. In general there are two dual epipoles which we will denote as “left” and “right” dual epipoles.

The dual image ray. Let  $l, l'$  be two lines coincident with the image point  $p$ . Then,  $l^\top MP = 0$  and  $l'^\top MP = 0$ , therefore  $P$  has a 1-parameter degree of freedom, i.e., it is determined up to a line which is defined by the intersection of the two planes  $l^\top M$  and  $l'^\top M$ . This line is the image ray corresponding to  $p$ . The same applies in the dual: the camera vector  $\hat{M}$  is determined up to a line — the line passing through  $p$  and  $\text{null}(\hat{P})$  (defined by  $l^\top \hat{P} \hat{M} = 0$  and  $l'^\top \hat{P} \hat{M} = 0$ ). This is defined as the *dual image ray*. Note that in the general case this mean that the camera center varies along a 1-parameter curve (non-linear). In the reduced case, however,  $\text{null}(M)$  (the camera center) varies along a line — the line  $\bar{P}p$ .



**Fig. 1.** Stabilized reference plane. (a) the dual epipole. (b) The dual epipolar line  $\mathcal{F}p_5$ . (c) Homography duality. (d) Homography slice of the dual trilinear tensor.

*Claim.* In the reduced case,  $p \cong \hat{P}\hat{M}$  constrains  $null(M)$  to vary along the line  $\overline{Pp}$ , which is the line passing through  $p$  and  $P$ .

**Proof:** Note that for any line  $l$  that pass trough  $p$

$$0 = l^\top \hat{P}\hat{M} = l^\top \begin{bmatrix} -W & 0 & 0 & X \\ 0 & -W & 0 & Y \\ 0 & 0 & -W & Z \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ -\delta \end{pmatrix}$$

where the camera projection on the right-hand side has its center at  $P$ , and  $null(M) = (\alpha, \beta, \gamma, -\delta)^\top$ . Therefore,  $null(M)$  is on  $\overline{Pp}$ .  $\square$

*The dual epipolar line.* Recall that the dual fundamental matrix  $\mathcal{F}$  is a result of eliminating  $\hat{M}$  from the equations  $p_5 \cong \hat{P}_5\hat{M}$  and  $p_6 \cong \hat{P}_6\hat{M}$ . Recall also that  $p_5 \cong \hat{P}_5\hat{M}$  constrains  $\hat{M}$  to lie on the dual image ray of  $p_5$ . The projection of  $P_6$ ,  $MP_6 = \hat{P}_6\hat{M}$  is a line (lines are preserved under projection) in the image corresponding to all the vectors  $\hat{M}$  that vary along the dual image ray of  $p_5$ . This line,  $\mathcal{F}p_5$  is defined as the *dual epipolar line*. Note that in the reduced case, if  $\hat{M}$  varies along the dual image ray of  $p_5$ , then  $null(M)$  varies along the line  $\overline{P_5p_5}$ . Thus we obtain the simple geometric interpretation [11] drawn in Fig. 1b.

*Homography Duality.* The term “dual homography” is already used in classic projective geometry, thus we refer to the dual case of the homography matrix as “Homography duality”. Recall that in multi-view, the homography matrix  $H_\pi$  induced by a plane  $\pi$ , defines  $H_\pi p$  as the projection of  $P$  onto the second image plane, where  $P$  is the intersection point of the image ray corresponding to  $p$  with  $\pi$ . In the dual case,  $H_\pi p_5$  is the projection of  $P_6$  when the vector (point)  $\hat{M}$  is at the intersection of the dual image ray of  $p_5$  and the plane  $\pi$ . Let  $[n^\top, \lambda]$  be the normal to the plane  $\pi$ , and denote by  $\pi'$  the plane defined by:  $[n^\top, -\lambda]$ . In the reduced case, constraining  $\hat{M}$  to lie on  $\pi$  is equivalent to constraining  $null(M)$  to lie on  $\pi'$ . So  $null(M)$  is at the intersection of the line  $\overline{P_5p_5}$  with  $\pi'$ , thereby providing a simple geometric interpretation — see Fig. 1c.

The next three claims are provided without proofs due to lack of space:

*Claim.* Let  $p_6^\top \mathcal{F} p_5 = 0$ . Then  $\mathcal{F}^\top H_\pi$  is skew-symmetric for all choices of planes  $\pi$ .

Claim 4 is crucial for later on when we discuss the source of the non-linear constraints of the dual trilinear tensor.

*Claim.*  $[e_{65}]_\times H_\pi \cong \mathcal{F}$  for all  $\pi$ .

#### 4.1 Trilinear Tensor Properties

Given the elements introduced above we will describe the source of the non-linear constraints of the dual trilinear tensor, but first we will introduce a useful equation:

*Claim (dual reprojection equation).* Let  $l^6$  be a line coincident with the point  $p_6$ . Then,

$$p_5^i l_j^6 \mathcal{T}_i^{jk} \cong p_7 \quad (2)$$

**Proof:**  $p_5^i l_j^6 \mathcal{T}_i^{jk}$  is a point (contravariant vector)  $q$ . Since  $p_5^i l_j^6 l_k^7 \mathcal{T}_i^{jk} = 0$  for all lines  $l^7$  coincident with  $p_7$ , then  $l^7$  and  $q$  are coincident — hence,  $q = p_7$ .  $\square$

The reprojection equation maps the point  $p_5$  and any line through  $p_6$  onto the point  $p_7$ . It is dual to the multi-view reprojection where matching points in views 1,2 are mapped onto the matching point in view 3. The dual reprojection equation can be used for purposes of tracking (6 points predicting the position of the 7'th) and will be detailed further in the experimental section.

Homography slice. A single covariant contraction of the tensor produces a homography duality (just like in the multi-view tensor [16]).

*Claim.* The matrix  $\delta_k \mathcal{T}_i^{jk}$  is a homography duality induced by the plane  $\pi$  defined by  $null(\hat{P}_7)$  and the line  $\delta$  in the image.

**Proof:** Recall that  $p_5^i l_j^6 l_k^7 \mathcal{T}_i^{jk} = 0$  constrains  $\hat{M}$  to lie at the intersection of the dual image ray of  $p_5$ , the plane  $null(\hat{P}_6) \vee l^6$  and the plane  $null(\hat{P}_7) \vee l^7$  (since  $l^i \hat{P}_i \hat{M} = 0$ ). By a single contraction  $\delta_k \mathcal{T}_i^{jk}$  we therefore constrain  $\hat{M}$  to lie on the plane  $null(\hat{P}_7) \vee \delta$ .  $\square$

Note that in the reduced case, this also constrains  $null(M)$  to lie on the plane  $P_7 \vee \delta$  (see claim 4), thus we obtain the geometric interpretation shown in Fig. 1d.

Non-linear Constraints. We should expect 4 non-linear constraints on the trilinear tensor. Consider three homography slices:  $\mathcal{T}_i^{j1}, \mathcal{T}_i^{j2}, \mathcal{T}_i^{j3}$  and denote them as  $H_1, H_2, H_3$ . From Claim 4 we have that  $H_1^\top \mathcal{F}$  provides 6 linear constraints on  $\mathcal{F}$  and so do  $H_2^\top \mathcal{F}$  and  $H_3^\top \mathcal{F}$  — taken together 18 linear constraints. Choose 8 of these constraints, then the entries of  $\mathcal{F}$  are represented by  $8 \times 8$  determinant expansions from the  $8 \times 9$  estimation matrix constructed from the 8 constraints. Each of these determinant expansions is a polynomial in the entries

of  $\mathcal{T}_i^{jk}$ . The remaining 10 constraints are of rank 4 because only 4 constraints are required to specify  $\mathcal{F}$  ( $9 - 1 - 4 = 4$ ). Therefore by substituting the representation of the entries of  $\mathcal{F}$  as determinant expansions in the remaining 10 constraints (choose any 4 of them) we obtain 4 polynomials in the entries of  $\mathcal{T}_i^{jk}$ .

*Reconstruction.* All the information required for the 3D reconstruction task, is contained in the dual epipoles. In the case of the 6-point tensor, Let  $\mathcal{F}$  be recovered from image measurements (we need at least 4 views of the 6 points). The dual epipoles satisfy  $\mathcal{F}e_{65} = 0$  and  $\mathcal{F}^\top e_{56} = 0$ . Point  $P_5$  can be assigned  $(1, 1, 1, 1)$  (to complete the projective basis) and we are left with recovering point  $P_6$ . Recall:  $e_{65} \cong \hat{P}_6 \text{null}(\hat{P}_5) = (X_6 - W_6, Y_6 - W_6, Z_6 - W_6)^\top$  and  $e_{65} = \hat{P}_5 \text{null}(\hat{P}_6) = ((X_6 - W_6)/X_6, (Y_6 - W_6)/Y_6, (Z_6 - W_6)/Z_6)^\top$ .

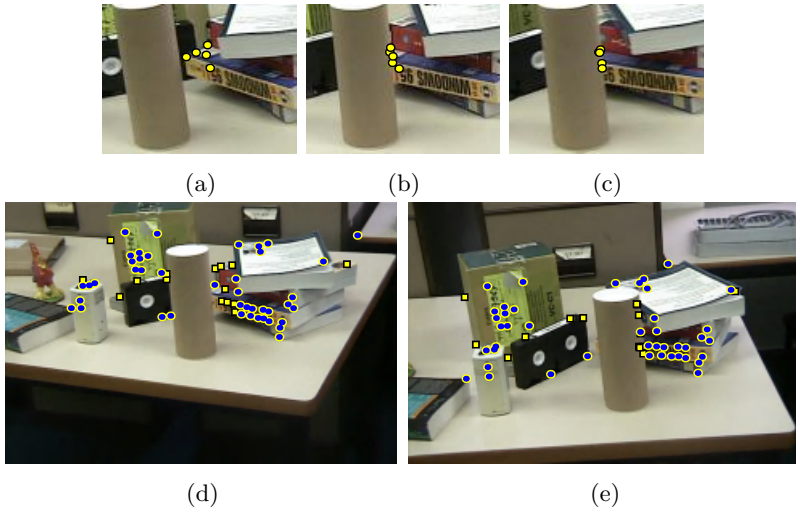
Therefore, taking the ratio (component-wise) of  $e_{56}/e_{65}$  would produce  $X_6, Y_6, Z_6$  up to a common scale, and  $W_6$  can be recovered by substitution in  $e_{65}$  thus obtaining a linear solution for  $P_6$  (up to scale). This reconstruction approach easily generalizes to the trilinear and quadlinear tensors. First one recovers the dual epipoles from the tensor and the reconstruction proceeds from there in a similar manner as with the 6-point tensor.

Regarding the reconstruction from the stabilized reference plane tensor, we first recall that  $P_6$  has only two parameters. On the other hand, in the reduced case the left and right epipoles coincide, so recovering the dual epipole from  $\mathcal{F}$  provides only two constraints on  $P_6$ , and this is the most one can expect.

In the trilinear tensor case, the points  $P_6, P_7$  contains 5 parameters. Therefore considering only the epipoles  $e_{56} = e_{65}$  and  $e_{57} = e_{75}$  provides only 4 constraints, which is not sufficient. Therefore, one should use the third epipole  $e_{76} = e_{67}$  as well. Note that in the reduced case, the epipole associated with the points  $P_i, P_j$  has the form:  $e_{ij} \cong (X_i - X_j, Y_i - Y_j, Z_i - Z_j)^\top$  (assuming that  $W_i = W_j = 1$ ). So the three epipoles provides us with the system:

$$\begin{aligned} e_{56}(3)(X_6 - 1) &= e_{56}(1)(Z_6 - 1) \\ e_{56}(3)(Y_6 - 1) &= e_{56}(2)(Z_6 - 1) \\ e_{57}(3)(X_7 - 1) &= e_{57}(1)(Z_7 - 1) \\ e_{57}(3)(Y_7 - 1) &= e_{57}(2)(Z_7 - 1) \\ e_{67}(3)(X_6 - X_7) &= e_{67}(1)(Z_6 - Z_7) \\ e_{67}(3)(Y_6 - Y_7) &= e_{67}(2)(Z_6 - Z_7) \end{aligned}$$

where  $e_{ij}(k)$  is the  $k$ 'th coordinate of the epipole  $e_{ij}$ . Though this system contains 6 equations, it has one trivial solution  $X_6 = Y_6 = Z_6 = X_7 = Y_7 = Z_7 = 1$ . If the system has a non-trivial solution, it must be of rank 5 most. Therefore the system enables the recovery of  $P_6, P_7$  up to one degree of freedom, as expected. Similarly, we can achieve 3D reconstruction from the quadlinear tensor, using 4 out of the 6 epipoles. Note, as pointed out in [18], that in the reduced case it possible to collect any number of images and perform factorization — however, points on the reference plane would not be reconstructed (and points near the reference plane would be subject to unstable reconstruction). Thus, the reconstruction formulas described above are of interest as they hold generally.



**Fig. 2.** Separating badly tracked points due to occlusions. (a,b) two images of a sequence of 100 — one of the problematic areas is zoomed in. (c) the 6-point tensor evaluated for all points. The badly tracked points are clustered to the right and (except one mistake) are well separated from the good points. (d,e) two views of the sequence with the labeling of good and bad points.

## 5 Experiments

We describe three applications. The first is occlusion detection in a point tracking experiment. Fig. 2(a,b) displays two images from a 100 frames sequence, and a set of points that were located and tracked by an automatic tracker (openCV’s [15] pyramid LK tracker). In the first image the tracker located some points on the edge of the books, but as the cylinder starts covering those edges the tracked points on the edges start moving as well and loose their accuracy. We use the 6-point tensor to evaluate configurations of 6 points in the following manner. The scene contained roughly 50 feature points which were automatically detected and tracked. For each of those points, 50 quintets of points were randomly selected. For each such sextet (the tested point + the quintet), we computed robustly (using LMeds [14]) the 6-points tensor over the entire sequence. For badly tracked points, we expect that for each sextet, the evaluation error (the contraction of the tensor with points  $p_5, p_5$ ) will be high. For a good point, we expect that out of the 50 random sets, there would be enough good sets, so that the sum of errors achieved this way will be significantly lower than the sum of errors achieved for a badly tracked point. Fig. 2(c) displays the error graph — one can observe a good separation between the badly tracked points (clustered on the right) and the good points (modulo one exception). Fig. 2(d-e) presents two images from the sequence with the labeling of good and bad points.



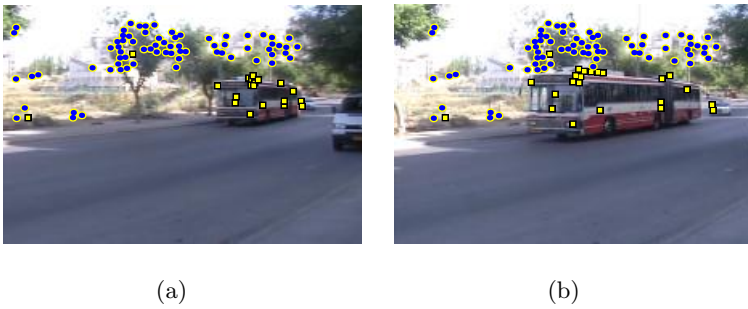
**Fig. 3.** Using the dual trilinear tensor to resume tracking of lost points. The images are part of a sequence of 160 images in which roughly 50 points were detected in frame 1 and tracked throughout the sequence. The points overlaid in (a) are those which were lost as some point due to the passing person (b) and recovered in (c).

The second application using the 7-point dual tensor reprojection equation (eqn. 2) describes a tracking recovery due to interference. In other words, situations in which the tracker loses a point due to low confidence yet in future frames the point reappears and one would like to resume tracking it. Fig. 3(a-c) shows a sample of 3 images, out of a sequence of 160. Roughly 50 points were automatically located and tracked in the beginning of the sequence. Some of those points were occluded due to the presence of a moving person and the goal was to recover the lost points once the occlusion was removed. For each lost point we wish to recover, we randomly chose a set of sextets of points out of the set of points that were successfully tracked all other the movie. For each such septet (the sextet+the point we wish to recover), we compute a 7-points dual trilinear tensor, using the images at the beginning of the movie, where we had all the 7 points. After the 7th point was lost, we used the computed shape tensor and the projections of the first 6 points in order to recover the 7th point. The recovered points that are shown in Fig. 3(c), are the average of the results that were achieved using the set of sextets we chose.

The third application is motion segmentation. Fig. 4 shows two images out of a sequence of 40 images of a moving bus (taken from a moving camera) where roughly 95 points were automatically located and tracked along the sequence. The segmentation technique uses the 6-point dual tensor in a manner similar to the first tracking application.

## 6 Summary

Single-view shape invariants were introduced in the past, however, much of the underlying constraints and forms remained incomplete. In this paper we have introduced a full account of the dual tensors which have exactly the same form as the multi-view tensors but with additional constraints borne out of the special structure of the dual projection matrices. The first difference lies with the internal constraints (synthetic constraints) — we have shown that the trilinear



**Fig. 4.** Two images out of a sequence of 40 images of a moving bus taken from a moving camera. The 6-point dual tensor was used for segmentation.

tensor has 16 of them and the quadlinear tensor has 58. The nature of these constraints change when a plane has been identified in advance and stabilized (the reduced case). There we have shown that the trilinear tensor has 21 constraints and the quadlinear tensor 72 constraints. We have introduced the dual of the tensor slices, the reprojection equation, the dual epipoles, the homography duality, the non-linear constraints of the trilinear tensor, and reconstruction from dual epipoles.

Given the understanding of the internal constraints and the introduction of the dual reprojection equation we made use of the shape tensors for two applications of tracking and an application of 2-body segmentation. The hope is that with a better understanding of the underlying internal structure of these shape invariants the applications (which have so far been a few) using these invariants would increase as well.

## References

1. N. Canterakis. A minimal set of constraints for the trifocal tensor. In *Proceedings of the European Conference on Computer Vision*, Dublin, Ireland, June 2000.
2. S. Carlsson. Duality of reconstruction and positioning from projective views. In *Proceedings of the workshop on Scene Representations*, Cambridge, MA., June 1995.
3. S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *International Journal of Computer Vision*, 27(3), 1998.
4. A. Criminisi, I. Reid, and A. Zisserman. Duality, rigidity and planar parallax. In *Proceedings of the European Conference on Computer Vision*, Friburg, Germany, 1998. Springer, LNCS 1407.
5. O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995.
6. W. Fulton and J. Harris. *Representation Theory*. Springer, 1991.

7. R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.
8. R.I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.
9. A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the International Conference on Computer Vision*, pages 1058–1063, Cambridge, MA, June 1995.
10. A. Heyden. A common framework for multiple view tensors. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, Freiburg, Germany, June 1998.
11. M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Proceedings of the European Conference on Computer Vision*, LNCS 1064, pages 17–30, Cambridge, UK, April 1996. Springer-Verlag.
12. M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multiview parallax geometry and applications. In *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany, 1998. Springer, LNCS 1407.
13. D.W. Jacobs. Space efficient 3D model indexing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 439–444, 1992.
14. P. Meer, D. Mintz, D. Kim and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision* 6(1), 1991.
15. Open source computer vision library  
<http://www.intel.com/research/mrl/research/cvlib/>
16. A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the International Conference on Computer Vision*, June 1995.
17. A. Shashua and Lior Wolf. On the structure and properties of the quadrifocal tensor. In *Proceedings of the European Conference on Computer Vision*, Dublin, Ireland, June 2000.
18. C. Rother and S. Carlsson. Linear Multi View Reconstruction and Camera Recovery. In *Proceedings of the International Conference on Computer Vision*, Vancouver, Canada, July 2001.
19. G. Stein and A. Shashua. On degeneracy of linear reconstruction from three views: Linear line complex and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):244–251, 1999.
20. B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, pages 338–343, Cambridge, MA, June 1995.
21. D. Weinshall, M. Werman, and A. Shashua. Duality of multi-point and multi-frame geometry: Fundamental shape matrices and tensors. In *Proceedings of the European Conference on Computer Vision*, LNCS 1065, pages 217–227, Cambridge, UK, April 1996. Springer-Verlag.



# Tracking and Rendering Using Dynamic Textures on Geometric Structure from Motion

Dana Cobzas and Martin Jagersand

Department of Computing Science, University of Alberta, Canada  
[http://www.cs.ualberta.ca/~ {dana,jag}](http://www.cs.ualberta.ca/~{dana,jag})

**Abstract.** Estimating geometric structure from uncalibrated images accurately enough for high quality rendering is difficult. We present a method where only coarse geometric structure is tracked and estimated from a moving camera. Instead a precise model of the intensity image variation is obtained by overlaying a dynamic, time varying texture on the structure. This captures small scale variations (e.g. non-planarity of the rendered surfaces, small camera geometry distortions and tracking errors). The dynamic texture is estimated and coded much like in movie compression, but parameterized in 6D pose instead of time, hence allowing the interpolation and extrapolation of new poses in the rendering and animation phase. We show experiments tracking and re-animating natural scenes as well as evaluating the geometric and image intensity accuracy on constructed special test scenes.

A problem of significant interest is how to capture and represent the image information from several sample views of a scene for the purpose of generating views from novel directions of that same scene. Such methods have applications in the confluence of vision and graphics where real scenes, too complex or tedious to model with conventional techniques, can instead be captured from photos or video and included in graphics renderings and animations. Image-based rendering techniques focus on the ray set and generate images by mapping pixels from an original set of sample views without having a precise geometric model of the scene. There are two main approaches to this problem. One is sampling the plenoptic function under some viewing constraints - limited camera motion inside a bounding box [6,11] or to only rotations [12]. Another approach is presented in [10], where a photoconsistent volumetric model is reconstructed from a set of input images by organizing the rays. These methods will theoretically generate correct renderings but are practically hard to apply for real scene and require calibrated cameras.

Another approach is to reconstruct a projective, affine or metric model from the input views using traditional structure from motion techniques. New renderings are generated by mapping the texture from the source images on the model. In most of the cases the metric structure is reconstructed based on planar surfaces [18] or lines [4] and the process is tedious and require a lot of human intervention. In order to have a valid texture reprojection, the model has to be decomposed in small planar patches. An advantage of metric reconstruction is

that the model can be correctly reprojected under perspective projection. Non-euclidean models are more easy to construct from a set of input images [5,21], but without additional metric information it is difficult to specify physically correct novel views.

Two significant practical challenges are: (1) Corresponding points are needed when acquiring models from images. In principle, accurate geometric structure can be obtained if the correspondences of all points in a scene between successive images can be determined. In practice, usually only a few points can be tracked reliably and accurately and object structure estimated at best coarsely. (2) In conventional texture-based rendering, the placement of triangles so that real edges on the object are aligned with edges in the model is critical. However, with a sparse model obtained from images of an otherwise unknown object this is difficult to ensure.

In this paper we present a two stage method which computes a coarse structure from motion, and then compensates for inaccuracies in the structure by modeling residual image variation as a dynamic, time varying texture on the coarse structure (see Figure 1). Both the structure and dynamic texture variation is parameterized in terms of object-camera pose, hence allowing reprojection and rendering in new positions.

In [1] a mixture model is introduced where intensity variation is modeled as a linear combination of iconic, form (motion), specular and illumination changes. We extend this by considering variation not on the image plane but on an object surface approximation. We also derive a connection between geometric plane homography warps and spatial image derivatives and show how image motion resulting from the warps can be expressed in the same way as iconic variation [13], i.e. using a set of specially tuned “eigen-filters” here optimized for capturing image motion instead of object appearance. Note that, as we show in [9], these eigen-filters parameterize motion in directly in image intensity space, unlike other approaches [2,1], where the optic flow field has to be computed.

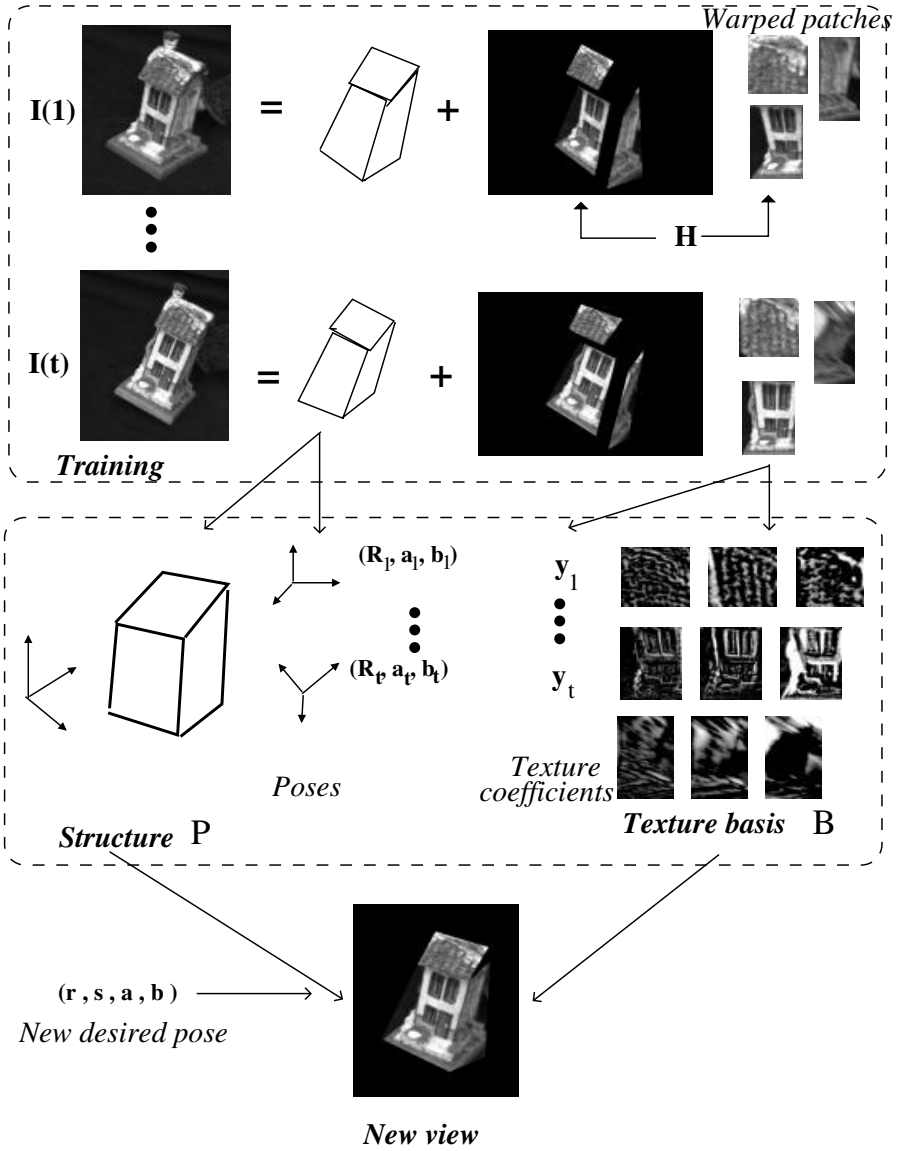
We present experimental results that compare the dynamic texturing with a regular static texture rendering and measuring errors in image intensities of the rendered images and geometric pixel errors of corresponding points.

## 1 Theory

The presented method generates new views by warping a *dynamic texture* on the projection of an estimated *structure* of the scene (see Figure 1). The input to the algorithm is a sequence of images  $I(t)$  and tracked fiducial points  $\mathbf{p}(t) = (\mathbf{u}(t), \mathbf{v}(t))$  grouped in quadrilateral regions. More precisely, each image  $I$  is composed from  $Q$  quadrilaterals  $I_q$ :

$$I = \sum_{q=1}^Q I_q \quad (1)$$

where each quadrilateral patch is obtained by warping its corresponding dynamic texture  $I_{wq}$  (Equation 3) from a standard shape (rectangle) to the desired posi-



**Fig. 1.** A sequence of training images  $I(1) \dots I(t)$  is decomposed into geometric shape information and dynamic texture for a set of quadrilateral patches. The scene structure  $P$  and motion  $(r, s, a, b)$  is determined from the projection of the structure using a factorization algorithm. The dynamic texture for each quadrilateral is decomposed into its projection  $y$  on an estimated basis  $B$ . For a given desired position, a novel image is generated by warping new texture synthesized from the basis  $B$  on the projected structure.

tion specified by the projection  $\mathbf{u}_q, \mathbf{v}_q$  of its corresponding fiducial points  $P$  in the affine structure (Equation 4) through a homography (Equation 2, Section 1.2).

$$I_q = I_{wq}(H(\mathbf{u}_q, \mathbf{v}_q)) \quad (2)$$

$$I_{wq} = B_q \mathbf{y}_q + \bar{I}_q \quad (3)$$

$$\begin{bmatrix} \mathbf{u}_q \\ \mathbf{v}_q \end{bmatrix} = RP + \begin{bmatrix} a \\ b \end{bmatrix} \quad (4)$$

Equation 3 represents the dynamic texture decomposition into its components on an estimated basis that capture the image variability caused by nonplanarities and illumination changes, algorithm described in Subsection 1.3. Equation 4 describes the reprojection of the estimated affine structure  $P$  in the image position specified by a scaled rotation  $R$  and the image components of the translation  $a, b$ . Subsection 1.1 describes the proposed factorization algorithm that is estimating the affine structure of the scene (fiducial points) and the motion parameters from the image sequence under weak perspective assumption.

### 1.1 Acquiring Geometric Structure from Motion

There are several techniques to build model structure from multiple uncalibrated images. In general they rely on obtaining corresponding points between images and solve for structure and motion using viewing geometry constraints. For a few images, under perspective projection assumption, this can be done using epipolar geometry (two images) or trilinear tensor (three images) [5,14].

In the case of video, *i.e.* long motion sequences, methods which utilize all image data in an uniform way are preferred. Such methods recover affine [21,22] or projective [19] structure using factorization approaches.

In our case of uncalibrated vision we have to pick a camera model and associated geometric framework. The methods above are formulated in either projective or affine geometry, corresponding to a perspective, weak perspective or orthographic camera assumption. In [3] we compare image based rendering using projective or affine geometry. Methods (e.g. [19,16]) using projective camera depend on the fundamental matrix,  $F$ , between pairs of images. With only a few tracked points we found it difficult to accurately estimate  $F$ .

On the other hand, a weak perspective model allow a direct linear formulation of the viewing geometry constraints between multiple images, and using factorization, an efficient method for decomposing the image data into object structure and pose. Using multiple images allow for stable solutions despite relatively few tracked points and typical tracking errors.

Here we develop an extension of the Tomasi-Kanade factorization algorithm[21] for weak perspective camera projection model inspired by [22]. First, the algorithm recovers affine structure from a sequence of uncalibrated images. Then, a relation between the affine structure and camera coordinates is established. This is used to transform the estimated scene structure to an

orthogonal coordinate frame. Finally, using similarity transforms expressed in metric rotations and translations, the structure can be reprojected into new, physically correct poses. Since we use only image information our metric unit of measure is pixel coordinates.

**Weak perspective projection - a factorization approach.** Under weak perspective projection, a point  $\mathbf{P}_i = (\mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i)_T$  is related to the corresponding point  $p_{ti} = (u_{ti}, v_{ti})_T$  in image frame  $I(t)$  by the following affine transformation:

$$\begin{aligned} u_{ti} &= s_t \mathbf{i}_t^T \mathbf{P}_i + a_t \\ v_{ti} &= s_t \mathbf{j}_t^T \mathbf{P}_i + b_t \end{aligned} \quad (5)$$

where  $\mathbf{i}_t$  and  $\mathbf{j}_t$  are the components along the camera rows and columns of the rotation  $R_t$ ,  $s_t$  is a scale factor and  $(a_t, b_t)$  are the first components  $\mathbf{t}\mathbf{1}_t$  of the translation  $\mathbf{t}_t$  ( $R_t$  and  $\mathbf{t}_t$  aligns the camera coordinate system with the world reference system).

Having  $N$  points tracked in  $M$  frames we can write

$$\begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & & \vdots \\ u_{M1} & \cdots & u_{MN} \\ v_{11} & \cdots & v_{1N} \\ \vdots & & \vdots \\ v_{M1} & \cdots & v_{MN} \end{bmatrix} = \begin{bmatrix} s_1 \mathbf{i}_1^T \\ \vdots \\ s_M \mathbf{i}_M^T \\ s_1 \mathbf{j}_1^T \\ \vdots \\ s_M \mathbf{j}_M^T \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_N \end{bmatrix}^T + \begin{bmatrix} a_1 \\ \vdots \\ a_M \\ b_1 \\ \vdots \\ b_M \end{bmatrix}$$

or

$$W = RP + \mathbf{t}\mathbf{1} \quad (6)$$

where  $W$  contains image measurements,  $R$  represents both scaling and rotation,  $P$  is the shape and  $\mathbf{t}\mathbf{1}$  is the translation in the image plane [22].

If the image points are registered with respect to their centroid in the image plane and the center of the world coordinate frame is the centroid of the shape points, the projection equation becomes:

$$\hat{W} = RP \text{ where } \hat{W} = W - \mathbf{t}\mathbf{1} \quad (7)$$

**Rank theorem.** Following [21], in the absence of noise  $\text{rank}(\hat{W}) = 3$ . Under most viewing conditions with a real camera the effective rank is 3. Assuming  $2M > N$ ,  $\hat{W}$  can be decomposed  $\hat{W} = O_1 \Sigma O_2$ , where  $O_1$  is an orthonormal  $2M \times N$  matrix,  $\Sigma$  is an  $N \times N$  diagonal matrix and  $O_2$  is an  $N \times N$  orthonormal matrix (SVD).

Defining

$$\begin{aligned} \hat{R} &= O_1' \\ \hat{P} &= \Sigma' O_2' \end{aligned} \quad (8)$$

we can write

$$\hat{W} = \hat{R} \hat{P} \quad (9)$$

$O'_1$  is formed from the first three columns of  $O_1$ ,  $\Sigma'$  is the first  $3 \times 3$  matrix of  $\Sigma$  and  $O'_2$  contains the first three rows of  $O_2$  (assuming the singular values are ordered in decreasing order).

**Metric constraints.** The matrices  $\hat{R}$  and  $\hat{P}$  are a linear transformation of the metric scaled rotation matrix  $R$  and the metric shape matrix  $P$ . More specifically there exist a  $3 \times 3$  matrix  $Q$  such that:

$$\begin{aligned} R &= \hat{R}Q \\ P &= Q^{-1}\hat{P} \end{aligned} \quad (10)$$

Normally, to align  $\hat{P}$  with an exocentric metric frame the world coordinates of at least four scene points are needed. In our case we assume no scene information, and we instead align  $\hat{P}$  with the pixel coordinate system of the camera row and column. This relates  $Q$  to the components of the scaled rotation  $R$ :

$$\begin{aligned} \hat{\mathbf{i}}_t^T Q Q^T \hat{\mathbf{i}}_t &= \hat{\mathbf{j}}_t^T Q Q^T \hat{\mathbf{j}}_t & (= s_t^2) \\ \hat{\mathbf{i}}_t^T Q Q^T \hat{\mathbf{j}}_t &= 0 \end{aligned} \quad (11)$$

where  $\hat{R} = [\hat{\mathbf{i}}_1 \cdots \hat{\mathbf{i}}_M \hat{\mathbf{j}}_1 \cdots \hat{\mathbf{j}}_M]^T$ . The first constraint assures that the corresponding rows  $s_t \hat{\mathbf{i}}_t^T$ ,  $s_t \hat{\mathbf{j}}_t^T$  of the scaled rotation  $R$  in Eq. 6 are unit vectors scaled by the factor  $s_t$  and the second equation constrain them to orthogonal vectors. This generalizes [21] from an orthographic to a weak perspective case. The resulting transformation is up to a scale and a rotation of the world coordinate system. To eliminate the ambiguity we align the axis of the reference coordinate system with the first frame and estimate only eight parameters in  $Q$  (fixing a scale). We solve this data fitting problem using Levenberg-Marquardt non-linear minimization algorithm [15].

To extract the motion parameters from each camera position, we first estimate the scale factor  $s_t$  and rotation components  $\hat{\mathbf{i}}_t$  and  $\hat{\mathbf{j}}_t$  by computing the norm of the rows in  $R$  that will represent the scale factors and then normalizing them. Considering that  $\hat{\mathbf{i}}_t$  and  $\hat{\mathbf{j}}_t$  can be interpreted as the orientation of the vertical and horizontal camera image axes in the object space, we compute the direction of the camera projection axis  $\mathbf{k}_t = \hat{\mathbf{i}}_t \times \hat{\mathbf{j}}_t$ . We now have a complete representation for the metric rotation that we parametrize with Euler angles  $r_t = [\psi_t, \theta_t, \varphi_t]$

$$[\hat{\mathbf{i}}_t, \hat{\mathbf{j}}_t, \mathbf{k}_t]^T = \mathcal{R}(r_t) =$$

$$\begin{bmatrix} \cos \psi_t \cos \varphi_t - \cos \theta_t \sin \psi_t \sin \varphi_t & \cos \psi_t \sin \varphi_t + \cos \theta_t \sin \psi_t \cos \varphi_t & \sin \theta_t \sin \psi_t \\ -\sin \psi_t \cos \varphi_t + \cos \theta_t \cos \psi_t \cos \varphi_t & -\sin \psi_t \sin \varphi_t + \cos \theta_t \cos \psi_t \cos \varphi_t & \sin \theta_t \cos \psi_t \\ \sin \theta_t \sin \varphi_t & -\sin \theta_t \cos \varphi_t & \cos \theta_t \end{bmatrix}$$

We recover the complete pose information  $X_t = [r_t, s_t, a_t, b_t]$  for each image  $I(t)$  in the initial sequence and the metric structure of the scene  $P$  (remember that the image components of the translation can be represented by the coordinates of the centroid  $a_t, b_t$ ).

**Reprojection property.** Given a set of position parameters at time  $t$ ,  $X_t = [r_t, s_t, a_t, b_t]$ , we can reproject the estimated object shape  $P$  using

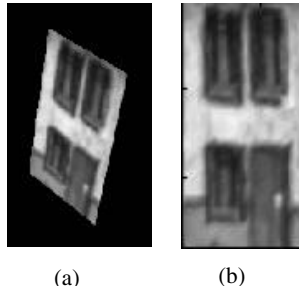
$$\mathbf{p}_t = \begin{bmatrix} u_{t1} \cdots u_{tN} \\ v_{t1} \cdots v_{tN} \end{bmatrix} = s_t \mathcal{R}(r_t) P + \begin{bmatrix} a_t \\ b_t \end{bmatrix} \quad (12)$$

## 1.2 Warping through a Homography

To apply the dynamic texture techniques described in Subsection 1.3, patches in the original images have to be warped to a standard shape. Considering that each patch is determined by four tracked points and represents a rigid part of an object, it can be approximated with a planar surface and mapped to a rectangular shape through a homography. It is well known [5,20] that, under perspective transformation, points in two planar scene views  $I(t_1)$  and  $I(t_2)$  are related by a 2D projective transformation (homography).

$$\begin{bmatrix} u_{1t_1} \cdots u_{Nt_1} \\ v_{1t_1} \cdots v_{Nt_1} \\ \xi_{1t_1} \cdots \xi_{Nt_1} \end{bmatrix} = H \begin{bmatrix} u_{1t_2} \cdots u_{Nt_2} \\ v_{1t_2} \cdots v_{Nt_2} \\ 1 \cdots 1 \end{bmatrix} \quad (13)$$

This transformation is up to a scale, so in general  $H$  has eight independent parameters  $h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8$  and it can be computed from four corresponding points in the two views. In our case we map the quadrilateral patches to a standard rectangular shape by computing the homography defined by the four corners and mapping the interior points through this homography.



**Fig. 2.** (a) Original patch (b) Warped patch

## 1.3 Texture Variability Modeling

Let  $I_w(t)$  be a rectified texture patch extracted from the image stream and warped to a standard rectangle as described in the previous section. Commonly, the texture  $I_w(t)$  is assumed constant. This is motivated by the fact that planar patches are mapped in a physically correct way to the camera plane. However,

in a real case there will be errors in the estimated plane parameters, and the real world surface that we model may not be quite planar. In addition, we include illumination in our variability model. Hence, here we purposefully focus on the intensity variation. In the following we first motivate analytically that under small motions the image intensity variations in the texture can be modeled as a set of spatial basis filters modulated by the motion, then we show how to statistically estimate this basis. Consider the distribution of texture differences  $I_z(t) = I_w(t) - I_{tref}$  around a texture image  $I_{tref}$ . This distribution can be described in both statistical and structural terms. Assuming small image variation we apply an optic flow constraint,  $I_z = \frac{\partial I_w}{\partial u} \Delta u + \frac{\partial I_w}{\partial v} \Delta v$ , where  $u$  and  $v$  are the camera coordinates.

**Structural image variability.** The above motion equation relates image intensity variation to a motion field. Assuming the texture patch is sourced from a rigid surface, the structure of the motion field is constrained to a low dimensional parameter space. Recall that we obtained the texture patch from a homography based on tracked points. Variation in the warping homography parameters  $h_1 \dots h_8$  introduces image variability of the form:

$$\Delta \mathbf{I}_w = \sum_{i=1}^8 \frac{\partial \mathbf{I}}{\partial h_i} \Delta h_i = \left[ \frac{\partial \mathbf{I}}{\partial u}, \frac{\partial \mathbf{I}}{\partial v} \right] \left[ \frac{\partial u}{\partial h_1} \dots \frac{\partial u}{\partial h_8} \right] \Delta h_i = [\mathbf{B}_1 \dots \mathbf{B}_8][y_1, \dots, y_8]^T, \quad (14)$$

Where  $\mathbf{I}$  is the image patch  $I_w$  flattened into a column vector. Note the form of the above equation: A weighted linear combination of eight filters only dependent on the spatial image derivatives times a constant matrix. This now reduces the space of possible image flow fields to an 8-dimensional variation.

In the above we assumed a planar patch. Real world patches will often not be planar, and this planarity will introduce a parallax error. The intensity variation due to the depth parallax can be written as a linear basis:

$$\Delta \mathbf{I}_d = [\mathbf{B}_{d1}, \mathbf{B}_{d2}][y_{d1}, y_{d2}]^T \quad (15)$$

**Illumination variation.** It has been shown that for a convex Lambertian object, the image variability due to different illumination can be expressed as a three dimensional linear basis[17,7]. For a general object, the illumination component can be approximated with a low dimensional basis. (In practice often three to five basis vectors suffice)

$$\Delta \mathbf{I}_i = [\mathbf{B}_{i1} \dots \mathbf{B}_{i3} \dots][y_{i1} \dots y_{i3} \dots]^T, \quad (16)$$

**Composite variation model.** Adding up the above contributions we can differentially model the composite image intensity variation as:

$$\Delta \mathbf{I} = \Delta \mathbf{I}_w + \Delta \mathbf{I}_d + \Delta \mathbf{I}_i + \Delta \mathbf{I}_e = \mathbf{B} \mathbf{y} + \Delta \mathbf{I}_e, \quad (17)$$



where  $\Delta \mathbf{I}_e$  is a noise term. The structural model in particular is only valid for parameter changes  $\Delta h_i$  which are small with respect to the spatial image derivatives. By extending the basis to a scale space hierarchy[8] (where the spatial derivatives are smoothed in the coarser scales) the applicability can be widened by adding more elements to the basis  $B = [\mathbf{B}_1 \dots \mathbf{B}_k]$ . However, as a result, the basis grows large and indeed is likely to represent more than the actually occurring texture variation.

**Statistical variability modeling.** In principle a variability basis  $B$  could be computed if very accurate a-priori models of the objects, lighting, cameras and their relative geometry are available.<sup>1</sup> In practice this is seldom the case. Instead we propose to estimate a sufficient basis  $\hat{B}$ . Consistent with the estimation of coarse geometry in the previous section we do this from observing an image sequence in an uncalibrated camera,  $\hat{\mathbf{I}}_w(t)$ , and obtaining a sample for  $t = 1 \dots M$ . Note that:

1. The estimated basis only represents the intensity variability actually present in the image sequence, which can be less than the possible variation as described above.
2. The estimated basis  $\hat{B}$  can be any linear transform of the analytically derived basis  $B$  above. It can also contain basis vectors that in addition to what is captured in  $B$  represents other types of variability.

A standard technique to estimate a linear basis  $\hat{B}$  which best captures (in a least squares sense) the part of  $B$  actually present in our observed intensity sequence is principle component analysis. The variation in texture is considered a stationary process with temporal mean  $\bar{\mathbf{I}} = \sum_{t=0}^M \frac{1}{M} \mathbf{I}_w(t)$ . Let the zero mean texture be  $\hat{\mathbf{I}}_z(t) = \mathbf{I}_w(t) - \bar{\mathbf{I}}$ , and a measurement matrix  $A = [\mathbf{I}_z(1), \dots, \mathbf{I}_z(M)]$ . The principle components are the eigen vectors of the covariance matrix  $C = AA^T$ . A dimensionality reduction is achieved by keeping only the first  $k$  of the eigenvectors.

For practical reasons, usually  $k \ll M \ll l$ , where  $l$  is the number of pixels in the texture patch, and the covariance matrix  $C$  will be rank deficient. We can then save computational effort by instead computing  $L = A^T A$  and eigen vector factorization  $L = VDV^T$ , where  $V$  is an ortho-normal and  $D$  a diagonal matrix. From the  $k$  first eigenvectors  $\hat{V} = [\mathbf{v}_1 \dots \mathbf{v}_k]$  of  $L$  we form a  $k$ -dimensional eigenspace  $\hat{B}$  of  $C$  by  $\hat{B} = A\hat{V}$ . Using the estimated  $\hat{B}$  we can now write a least squares optimal estimate of any intensity variation in the patch as

$$\Delta \mathbf{I} = \hat{B} \hat{\mathbf{y}}, \quad (18)$$

the same format as Eq. 17, but without using any a-priori information to model  $B$ . While  $\hat{\mathbf{y}}$  captures the same variation as  $\mathbf{y}$ , it is not parameterized in the

<sup>1</sup> One of the components,  $\Delta \mathbf{I}_w$ , can be computed only from spatial image derivatives. In tracking this commonly used for a low dimensional (in practice 2-4) variability model of planar warps[7]. However, the 8 derivatives for the homography parameters are difficult to compute accurately from only one image.

same coordinates, so in addition we estimate a second transform  $J$  between our pose description and  $\hat{\mathbf{y}}$ . In our application we represent one object using several texture patches, and estimate  $J$  between texture mixing coefficients  $\hat{\mathbf{y}}$  and tracked global camera-object pose  $\hat{\mathbf{x}}$  ( $= [r, s, a, b]^T$ ).

For every training image  $\mathbf{I}_t$  we have from the orthogonality of  $\hat{V}$  that the corresponding texture mixing are the columns of  $[\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_M] = \hat{V}^T$ . From the factorization of geometric structure we also have the corresponding  $\hat{\mathbf{x}}_t$ . Estimating a linear model  $\Delta\hat{\mathbf{y}} = J\Delta\hat{\mathbf{x}}$ , where  $\Delta\hat{\mathbf{y}}_t = \hat{\mathbf{y}}_t - \hat{\mathbf{y}}_{ref}$  we need 6 or more motions  $\Delta\hat{\mathbf{y}}, \Delta\hat{\mathbf{x}}$  (From one center reference and 6 nearby training images) to solve for  $J$  in

$$\begin{bmatrix} \Delta\hat{\mathbf{y}}_1 \\ \vdots \\ \Delta\hat{\mathbf{y}}_m \end{bmatrix} = \begin{bmatrix} \Delta\hat{\mathbf{x}}_1 \\ \vdots \\ \Delta\hat{\mathbf{x}}_m \end{bmatrix} J^T \quad (19)$$

The linear model  $\hat{\mathbf{y}} = \hat{\mathbf{y}}_{ref} + J\Delta\hat{\mathbf{x}}$  is only locally valid and has to be recomputed around each desired pose  $\hat{\mathbf{x}}_{ref}$ . A globally valid spline approximation  $\hat{f}$ , s.t.  $\hat{\mathbf{y}} = \hat{f}(\hat{\mathbf{x}})$  can be computed instead if, in the training samples,  $\hat{\mathbf{x}}_t$  is monotonic and plaid. For a linear spline, this corresponds to organizing the computed matrices  $J$  into an indexable array.

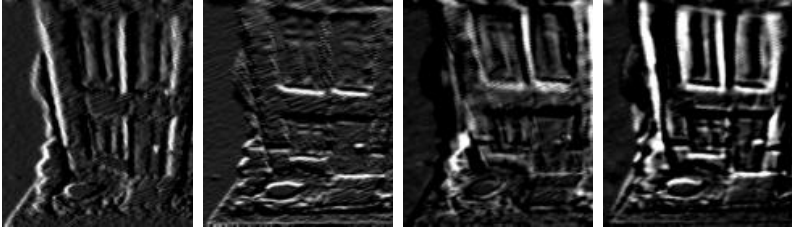
#### 1.4 Interpretation of the Variability Basis

In our application, the geometric model captures gross image variation caused by large movements. The remaining variation in the rectified patches is mainly due to:

1. Tracking errors as well as errors due to the weak perspective approximation cause the texture to be sourced from slightly inconsistent locations in the training images. These errors can be modeled as a small deviation  $\Delta[h_1, \dots, h_8]^T$  in the homography parameters from the true homography, and causes image differences according to Eq. 14. The weak perspective approximations, as well as many tracking errors are persistent, and a function of object pose. Hence they will be captured by  $\hat{B}$  and indexed in pose  $\hat{\mathbf{x}}$  by  $f$ .
2. Depth variation is captured by Eq. 15. Note that  $Z$  and  $\Delta Z$  is depth along the camera optic axis, and hence also varies as a function of object pose.
3. Assuming fixed light sources and a moving camera or object, the light variation is a function of relative camera-object pose as well.

From the form of Equations 14 and 15 we expect that pose variations in the image sequence will result in a texture variability described by combinations of spatial image derivatives. In Fig.3 we compare numerically calculated spatial image derivatives to the estimated variability basis  $B$ .

In synthesizing texture to render a sequence of novel images the function  $f$  modulates the filter bank  $B$  so that the new texture dynamically changes with pose  $\hat{\mathbf{x}}$  according to  $\mathbf{I}_w = Bf(\hat{\mathbf{x}}) + \bar{\mathbf{I}}$ .



**Fig. 3.** Comparison between spatial derivatives  $\frac{\partial I_w}{\partial x}$  and  $\frac{\partial I_w}{\partial y}$  (left two texture patches) and two vectors of the estimated variability basis  $[B_1, B_2]$  (right) for the house in Fig. 1.

## 2 Composition of Geometric and Image-Based Models

Based on the theory described in Section 1 we have developed an algorithm that generates new views using an estimated geometric model and a dynamic texture mapping.

**Training data.** We took sequences of  $M$  images  $I(t)$  and tracked  $N$  fiducial points  $p_t = [\mathbf{u}_t, \mathbf{v}_t]^T$  using SSD trackers from XVision system [7]. The tracked points are grouped into disjunctive quadrilaterals that will cover the scene. For a simple squared object, it is possible to find a decomposition into quadrilateral regions that correspond to physical planes. In practice, considering also the limitation of the tracking algorithm, we are decomposing the object into nonplanar regions. The dynamic texture algorithm will correct the reprojection errors due to nonplanarities.

**Structure from motion.** We use the factorization algorithm from Section 1.1 to acquire the structure of the scene  $P$  and calculate the pose  $\mathbf{x}_t$ .

1. Form the normalized measurement matrix  $\hat{W}$  by registering the image coordinates with respect to their centroid.
2. Compute SVD of  $\hat{W}$  and the approximate rotation  $\hat{R}$  and shape  $\hat{P}$  Equation 8.
3. Impose metric constraints from Equation 11 to calculate the true scaled rotations  $R$  and shape  $P$ .
4. Estimate the camera pose  $[r_t, s_t, a_t, b_t]$  for each view.  $s_t$  is the norm of the rows corresponding to  $\mathbf{i}_t$  and  $\mathbf{j}_t$  in  $R$ .  $r_t = [\psi_t, \theta_t, \varphi_t]$  are the Euler angles corresponding to the rotation matrix formed from  $\mathbf{i}_t, \mathbf{j}_t$  and  $\mathbf{k}_t = \mathbf{i}_t \times \mathbf{j}_t$ .

**Dynamic texture.** For each quadrilateral  $I_q$  we compute a texture basis  $B_q$  and a set of texture coefficients  $\mathbf{y}_q$ :

1. Warp the patch  $I_{qt}$  to a standard shape  $Iw_{qt}$  using the homography (see Section 1.2) determined by the four corners. We choose the biggest rectangle that includes the patch through the image sequence as the standard shape.

2. Form a zero mean sample and compute the PCA as described in Section 1.3. Keep  $k$ , typically about a dozen basis vectors  $B_q$  and the corresponding coefficients in each frame  $\mathbf{y}_{qt}$ .

**New view rendering.** Given a new pose  $\mathbf{x} = [r, s, a, b]^T$  we render the dynamic texture on the reprojected structure.

1. Compute reprojection  $p = [\mathbf{u}, \mathbf{v}]^T$  of shape  $P$  in the desired pose using Equation 12.
2. For each quadrilateral  $q$ 
  - a) interpolate a texture coefficient  $\mathbf{y}_q$  corresponding to the new pose  $\mathbf{x}$  using the nearest neighbors in the training data,
  - b) compute the texture in the standard shape using Equation 18,
  - c) rewrap the texture in the desired image position determined by the reprojected corners.

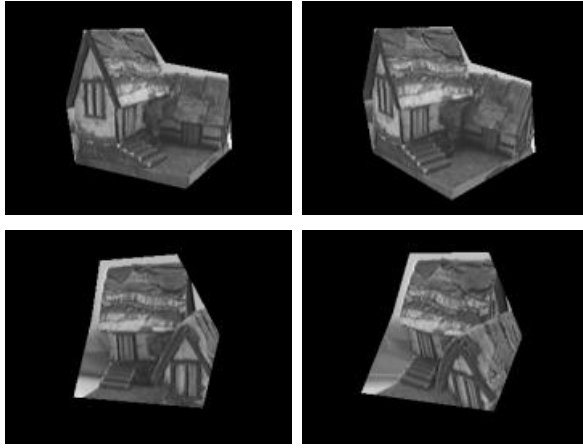
### 3 Experimental Results

To test the proposed algorithm we recorded motions of three types of objects: a calibration pattern (**pattern**), a toy house (**house**) and a flower (**flower**). We tested each for several sequence lengths between 15 and 256 images. The first two objects can be relatively accurately decomposed into a few planar patches. However, this is not the case for the flower. The motion was mostly composed by rotations with some depth variation. We tracked between 8 and 15 points using XVision [7] and grouped them into 3 or 4 non-overlapping quadrilaterals.

We compute the metric structure and pose for the tracked points using the algorithm from Section 1.1 and the texture basis and coefficients for each quadrilateral. As mentioned in Section 1.3, a minimum of 13 basis vectors will capture the errors caused by depth and illumination changes under small motions. In a real case with significant motion variation, and other errors like inaccuracies in tracking, more vectors are required to capture the image variability. In our case we kept about 25 texture basis vectors for the long sequences. For the shorter sequences, where the actual variation in motion was limited, we found that 3 to 5 vectors were enough. From the estimated model, we generated three types of new image sequences by:

1. Interpolating the original motion. New pose points were generated between the original poses and a longer temporally up-sampled movie of both the original and new poses was rendered from the model.
2. Smoothing the original motion. In this case we computed the original pose trajectory, smoothed it by polynomial fitting, and rendered a new video sequence with smooth motion.
3. Perturbing the original rotations by user supplied values (we used up to  $5^\circ$ - $15^\circ$ ) and rendering a motion edited animation.

Figure 4 shows some examples of rendered picture from the **house** sequence. The motion is limited by the fact that the factorization algorithm requires that the fiducial points are visible in all the frames. To generate larger motions we composed renderings from two sequences taken from different view angles. Another solution would be to incrementally update the model when introducing a new view based on the common fiducial points.



**Fig. 4.** Generated pictures from **house** sequence

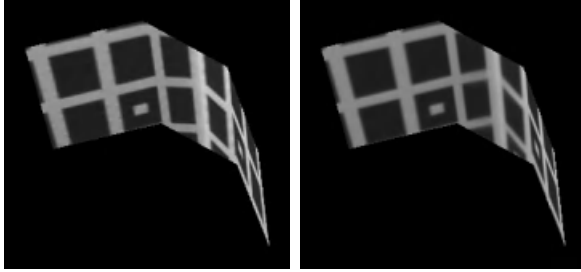
For testing the performance of our algorithm, we compared the *dynamic texture* algorithm with a *static texture* algorithm. For generating the static texture sequences we warped texture from original images onto the reprojected metric structure. Considering that the dynamic texture algorithm requires a mean image and  $k$  basis texture images, we source the static texture from  $k + 1$  original images. The source images are equally spaced through the sequence and the texture for the current generated image will be textured from the closest source image. There are two types of errors that we are investigating for the generated image sequence - *static errors* in individual rendered image frames and *dynamic errors* through the whole sequence.

### 3.1 Static Errors

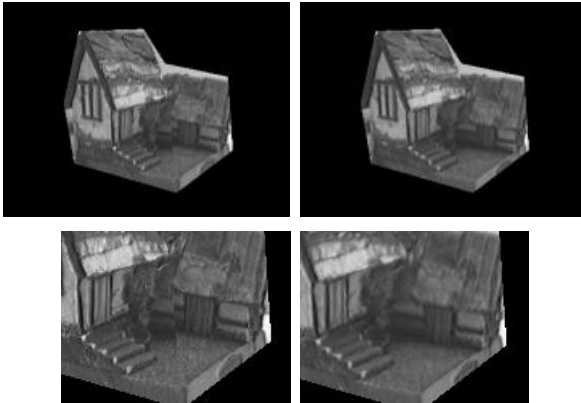
Most of the static errors are caused by nonplanarities of the texture patches that will result in a misalignment in the generated images. Figure 5 shows a rendered picture from the **pattern** sequence using static texture (left) and dynamic texture (right). Notice the effect of bent horizontal line in the left image, that was corrected using the dynamic texture in the right image. The scene was decomposed into three quadrilaterals two planar on the sides and one nonplanar in the middle, where the pattern has a corner. The nonplanar region in the

middle is causing this distortion. A similar effect can be seen in Figure 6, which shows pictures generated from the **house** sequence. The small house (middle of the zoomed picture) appear broken in the case when using static texture.

While in the case of the pattern or house it can be argued that these types of scenes can be better decomposed into planar surfaces, this would not be possible for many scenes with natural instead of man-made objects. As an example we show the **flower** sequence. Figure 7 shows the quadrilateral decomposition (left), a rendered image using static texture with visible geometric errors (middle) and a rendered image from the same pose using dynamic texture (right) where the geometric errors are compensated.



**Fig. 5.** Geometric errors (Left) Static texture: broken line caused by nonplanar patch; (Right) Dynamic texture: error was compensated



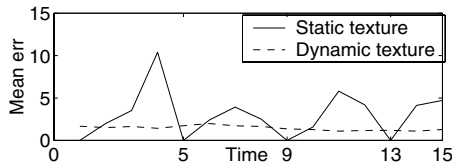
**Fig. 6.** Geometric errors in **house** sequence. (top) Rendered images using static and dynamic texture respectively; (bot) Detail showing the geometric error

To quantify the geometric errors we regenerate the original positions from the **pattern** sequence and compare them with the original images by measuring



**Fig. 7.** Geometric errors in **flower** sequence. (Left) One of the original images and the outline of the quadrilateral patches. (Middle) Image generated using static texture. (Right) Image generated in the same pose using dynamic texture.

the differences in pixel intensities (Figure 8). Notice that the error was almost constant in the case of dynamic texture and very uneven in the case of static texture. For the static texture case we used frame 0,5,9,13 for sourcing the texture (consistent with using three texture basis vectors in the dynamic case) so is expected that the error drops to zero when reproducing these frames. The mean relative intensity error was 1.17% in the case of static texture and 0.56% in the case of dynamic texture.



**Fig. 8.** Intensity pixel error in the rendered images (compared to original image)

### 3.2 Dynamic Errors

For an animation there are global errors through the whole movie that are not visible in one frame but in motion impression from the succession of the frames. We call these dynamic errors. We identified two types of dynamic errors connected to *depth impression* and *motion smoothness*. We again compared the static and dynamic texturing.

If the surfaces that are rendered with static texture are not physical planes, it is very hard to capture the depth impression when re-animating the object. Our dynamic texture algorithm capture the depth variation and give an impression of a “real” 3D object. This is only noticeable in a movie (refer to the attached movie **flower\_animation.mpg**). Notice how the rendering with dynamic texture animates the flower to create a realistic impression of the leaves being spaced in 3D, while in the static texture it looks like the flower is a picture pasted on a few planes.

Another important quality is smoothness of motion. When using static texture we source the texture from a subset of the original images ( $k + 1$  if  $k$  is the number of texture basis) so there is significant jumping when changing the texture source image. We tracked a point through a generated sequence from the **pattern** in the two cases and measure the smoothness of motion. Table 1 shows the average pixel jitter. As an application to motion smoothness, we used the proposed algorithm to correct the unevenness in the original motion. The enclosed movie **flower\_animation.mpg** shows the regenerated smooth motion of the original **flower** sequence.

**Table 1.** Average pixel jitter

	Vertical pixel jitter	Horizontal pixel jitter
Static texture	1.15	0.98
Dynamic texture	0.52	0.71

## 4 Discussion

The presented method allows acquisition of scene structure and appearance using video from an uncalibrated camera. The scenes can then be rendered from novel poses. Our main objectives were to balance requirements for calibration, user interaction, and computation, while rendering realistic images and motion animations. By using only coarse approximate geometric structure, this structure can be extracted from images in a computationally efficient and reliable way. The structure corresponds to the major salient object features. The user can tune the representation by focusing on the most salient structure in the scene, yet the interaction is limited to pointing out only a few (about a dozen) scene points in the first view of the scene to initialize the real-time visual tracking. To be able to accurately capture the scene details we developed a two stage method. First, the coarse geometry is used to warp scene patches into a canonical form. Then, the residual intensity variation is captured using a locally valid linear model based on a multidimensional optic flow-like constraints. The resulting representation is a *dynamic texture* which, when modulated by overall object pose, can add fine scale variation representing fine scale scene geometry.

To validate the model we recorded and re-animated various scenes and compared our dynamic texture rendering to conventional model-based rendering using a static texture image. We found that our method reduced both intensity error and motion jitter to about half of those values obtained by conventional rendering.

Applications of our method include: (1) Animation editing, where a movie segment of a scene with motion is recorded, but the motion trajectory needs to be somewhat adjusted afterwards. We tested this for adjustments up to 15 degrees of angle. (Translation range is unlimited). (2) Motion smoothing. This



can stabilize video where a scene has been recorded, for instance, from a vehicle, and the original video has vibrations and bumps. (3) Visual modeling in general when done for representing intensity variation rather than obtaining detailed geometry.

Currently, in one model, we only represent the scene regions which are simultaneously visible in the whole video sequence. To achieve larger motions we have to piece together several models. In future work we plan to extend our work to a panoramic camera, using cylindrical texture images to capture interior of a whole indoor scene in one model.

## References

1. M. Black, D. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, 78(1):8–31, 2000.
2. M. Black, Y. Yacoob, A. Jepson, and D. Fleet. Learning parameterized models of image motion. In *Computer Vision and Pattern Recognition*, pages 561–567, 1997.
3. D. Cobzas and M. Jagersand. A comparison of non-euclidean image-based rendering. In *Proceedings of Graphics Interface*, 2001.
4. P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Computer Graphics (SIGGRAPH'96)*, 1996.
5. O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Boston, 1993.
6. S. J. Gortler, R. Grzeszczuk, and R. Szeliski. The lumigraph. In *Computer Graphics (SIGGRAPH'96)*, pages 43–54, 1996.
7. G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.
8. M. Jagersand. Saliency maps and attention selection in scale and spatial coordinates: An information theoretic approach. In *Fifth International Conference on Computer Vision*, 1995.
9. M. Jagersand. Image based view synthesis of articulated agents. In *Computer Vision and Pattern Recognition*, 1997.
10. K. Kutulakos and S. Seitz. A theory of shape by shape carving. *International Journal of Computer Vision*, 38:197–216, 2000.
11. M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH'96)*, pages 31–42, 1996.
12. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Computer Graphics (SIGGRAPH'95)*, pages 39–46, 1995.
13. H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.
14. M. Pollefeys. *Tutorial on 3D Modeling from Images*. Lecture Notes, Dublin, Ireland (in conjunction with ECCV 2000), 2000.
15. W. H. Press, B. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1992.
16. S. M. Seitz and C. R. Dyer. View morphing. In *Computer Graphics (SIGGRAPH'96)*, pages 21–30, 1996.
17. A. Shashua. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, MIT, 1993.

18. I. Stamos and P. K. Allen. Integration of range and image sensing for photorealistic 3d modeling. In *ICRA*, 2000.
19. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV (2)*, pages 709–720, 1996.
20. R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.
21. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.
22. D. Weinshall and C. Tomasi. Linear and incremental aquisition of invariant shape models from image sequences. In *Proc. of 4th Int. Conf. on Compute Vision*, pages 675–682, 1993.

# Sensitivity of Calibration to Principal Point Position

R.I. Hartley<sup>1</sup> and R. Kaucic<sup>2</sup>

<sup>1</sup> Dept. of Systems Eng., RSISE, A.N.U., ACT, 0200 Australia  
Richard.Hartley@anu.edu.au

<sup>2</sup> G.E.-CRD, Schenectady, New York, 12309 U.S.A.  
kaucic@crd.ge.com

**Abstract.** A common practice when carrying out self-calibration and Euclidean reconstruction from one or more views is to start with a guess at the principal point of the camera. The general belief is that inaccuracies in the estimation of the principal point do not have a significant effect on the other calibration parameters, or on reconstruction accuracy. It is the purpose of this paper to refute that belief. Indeed, it is demonstrated that the determination of the focal length of the camera is tied up very closely with the estimate of the principal point. Small changes in the estimated (sometimes merely guessed) principal point can cause very large changes in the estimated focal length, and the accuracy of reconstruction. In fact, the relative uncertainty in the focal length is inversely proportional to the distance of the principal point to the epipolar line. This analysis is geometric and exact, rather than experimental.

## 1 Introduction

The principal point of a camera is often quite difficult to determine, and is known often to vary with the zoom setting of a camera. Often, therefore, the principal point is guessed when calibration or self-calibration of the camera takes place ([2, 10]). The generally held opinion is that this makes little difference to the estimate of the other camera parameters, or to the ultimate accuracy of Euclidean reconstruction. It is possible that this may be true under some conditions, but it has been quoted as a general principle often without critical examination. The purpose of this paper is to point out that this assumption is far from safe, and that in fact the estimated position of the principal point may have a large effect on the accuracy of calibration and reconstruction.

We are concerned throughout with natural pinhole cameras, that is cameras with zero skew and unit (or known) aspect ratio. First, we consider calibration of a camera from a single view, in which a horizon line and a vertical direction, or vertical vanishing point may be identified. Given knowledge of the principal point, the focal length may be determined by an easy geometric construction, from which the dependence of the focal length on the principal point estimate is easily seen.

This single-view case is next extended to the two-view case, in which the focal lengths of each camera may be determined from the fundamental matrix alone. It is shown that this problem may be reduced to the previous case of single-view calibration, by considering the horizon line and perpendicular vanishing direction of the plane formed by the base line and one of the camera's principal rays. The sensitivity of this process to variations in the principal point then becomes evident. Estimation of the focal length in

this manner is well known to be unstable ([9,2]. In part this is due to sensitivity in the estimation of the fundamental matrix of the camera pair. It is shown here, however, that inaccuracy in the assumed position of the focal length is also a major culprit.

In an influential paper ([2]) Bougnoux gives a beautifully simple formula for the focal length of the camera estimated from two views. As noted above, he finds that the estimate of the focal length is quite sensitive, which we confirm. However he reiterates the popular belief that the assumed position of the principal point does not have an important effect on the estimated focal length, or on the accuracy of an eventual projective reconstruction of the scene. This view has also been stated (often without sufficient evidence) by other researchers, but is contrary to the findings of the present investigation. Indeed one of the purposes of this paper is to demonstrate that these two findings are not universally true, and must be treated with caution. By giving a new geometrically intuitive interpretation of certain one and two-view self-calibration methods, we provide the user with a tool that can be used to ascertain whether the assumptions are valid in a particular reconstruction scenario.

## 2 Calibration from a Single Image

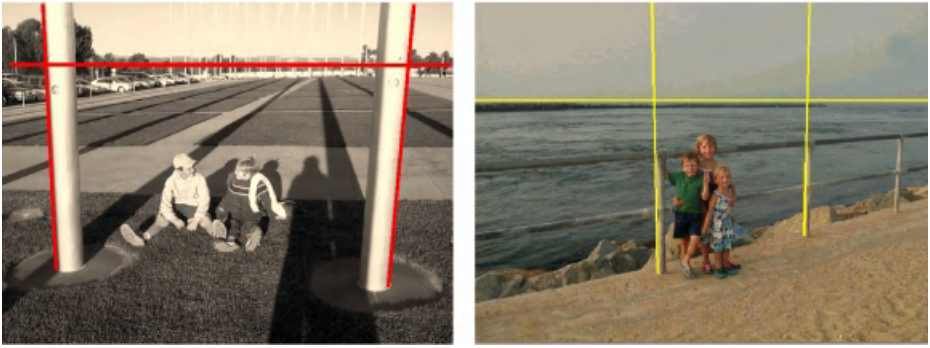
Throughout this paper, we assume that the skew of the camera is zero, and the aspect ratio is equal to one. Thus, the only remaining parameters of the camera are the focal length and the principal point.

We consider first calibration of the camera (determination of the focal length and principal point) from a single view. Clearly this can not be done without some scene information, and many ways have been proposed to do this. For instance, if three orthogonal vanishing points can be identified in the image, then the principal point may be identified as the orthocentre of the triangle formed by the three vanishing points ([3]). In addition, the focal length may be determined by direct geometric construction ([8], page 212, [4]).

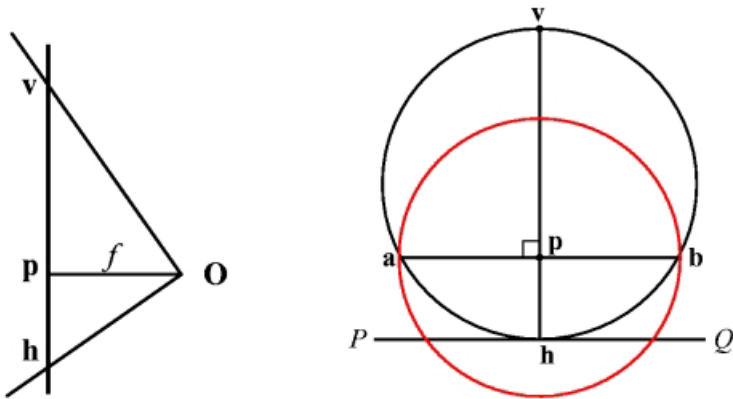
Thus, if the vanishing points of all three orthogonal directions are known, then the camera may be completely calibrated without any further assumptions. Therefore, we consider a calibration problem in which there is a little less information available. Accordingly, we assume that the horizon line and the vertical vanishing point (henceforth known as the apex) are identified in an image. An example of such an image is given in Fig. 1. The principal point in the image must lie on the perpendicular from the apex to the horizon. Thus, the principal point has a single remaining degree of freedom. Suppose that the principal point is the point  $\mathbf{p}$  in the image,  $\mathbf{v}$  is the vertex, and  $\mathbf{h}$  is the foot of the perpendicular from the apex to the horizon. In this case, the focal length may be computed by a simple formula:

$$f^2 = -d(\mathbf{p}, \mathbf{h})d(\mathbf{p}, \mathbf{v}) \quad (1)$$

where  $f$  is the focal length, and  $d(\cdot, \cdot)$  represents the (signed) Euclidean distance. The negative sign in this expression expresses the fact that  $d(\cdot, \cdot)$  is a signed quantity, and that the direction vectors from the principal point to the horizon and the apex must be in opposite directions in order for  $f^2$  to be positive, as given by this formula. In other words, the principal point must lie between the horizon and the apex. Proof of this formula is given in Fig. 2.



**Fig. 1.** Examples of images for which it is possible to compute the focal length of the camera, assuming knowledge of the principal point. To do this, it is sufficient to identify the horizon and a vertical vanishing direction. In the left-hand image, the exact position of the horizon can be established with the aid of the vanishing points of the ground markings and shadows. The image of the apex may be computed as the intersection of the two vertical edges. Focal length may then be computed using (1).



**Fig. 2. Left.** The diagram shows the vertical plane containing the principal ray  $pO$  of the camera. The two rays to the horizon and the apex are perpendicular to each other, meeting the focal plane at points  $h$  and  $v$  respectively. The focal length is the distance from the camera centre to the focal plane. By similar triangles,  $f^2 = -d(p, h)d(p, v)$ .

**Right.** In the image, a circle (black) is drawn with diameter the perpendicular line between the apex  $v$  and the horizon  $PQ$ . The principal point must lie on this line. A line is drawn through  $p$  perpendicular to  $vh$ . This meets the circle in two points  $a$  and  $b$ . The focal length equals the distance  $d(p, a)$ , and the circle with diameter  $ab$  is the  $45^\circ$  circle in the image (shown in red).

The focal length may be computed by a simple geometric construction, as follows. Let  $C$  be the circle with the line from apex  $v$  to horizon  $h$  as diameter. The line in the image through the principal point  $p$  meets the circle  $C$  in two points  $a$  and  $b$ . By elementary geometry of a circle,  $d(p, a)d(p, b) = d(p, v)d(p, h)$ , and since  $d(p, a) = -d(p, b)$

it follows that  $f = d(\mathbf{p}, \mathbf{a})$ . The circle constructed with centre  $\mathbf{p}$  passing through  $\mathbf{a}$  and  $\mathbf{b}$  is the  $45^\circ$  circle in the image. This is illustrated in Fig. 2, and on a real image in Fig. 3. Since  $f$  is the distance from  $\mathbf{p}$  to the circle  $C$  along a line drawn perpendicular to the diameter  $\mathbf{vh}$ , the way the estimated value of  $f$  varies as a function of the presumed principal point location is easily visualized. In particular, as the assumed principal point moves towards the horizon, or the apex in the image, the corresponding estimated value of the focal length  $f$  diminishes towards zero. If an image is taken with a camera aimed directly at the horizon, the principal point corresponds with the point  $\mathbf{h}$  on the horizon, and so  $d(\mathbf{p}, \mathbf{h}) = 0$ . In this case, however, the distance to the apex will be infinite, and hence  $f^2 = 0 \times \infty$ , and the computed value of  $f$  will be indeterminate. From this, we see that it is impossible to compute  $f$  from this configuration if the camera is pointed directly at the horizon, or the apex.

## 2.1 A Different Formula

It is sometimes difficult to measure the distance of the principal point to the apex, particularly if the apex is far away from the visible area of the image. A simple modification of the image takes care of that problem. Suppose that  $\mathbf{l}$  is a line from the apex to the horizon, not passing through the principal point. This can be the image of a vertical object in the scene. Let  $s_1 = d(\mathbf{p}, \mathbf{h})$  be the orthogonal distance from the principal point to the horizon, and let  $s_2 = d(\mathbf{p}, \mathbf{l})$  be the orthogonal distance from the principal point to the line  $\mathbf{l}$ . Let  $\theta$  be the angle between line  $\mathbf{l}$  and the horizon – more specifically, the angle of the region containing the principal point. Then

$$f^2 = s_1 s_2 / \cos \theta \quad (2)$$

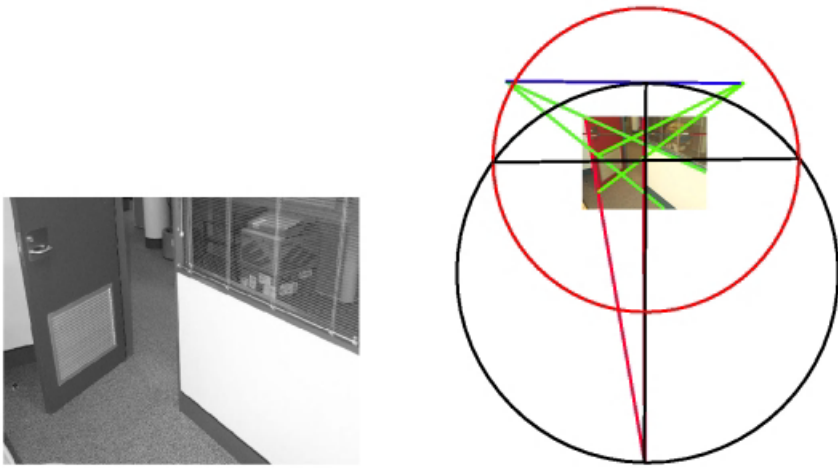
The proof of this is immediate from Fig. 4. Note that for  $f^2$  to be positive, the principal point must lie in the acute-angled section of the plane partitioned by lines  $\mathbf{h}$  and  $\mathbf{l}$ .

*Sensitivity of  $f$ .* In terms of (2) it is easy to write down an expression for the uncertainty of the focal length in terms of small errors in the three parameters  $s_1$ ,  $s_2$  and  $\theta$ . Since  $df = \partial f / \partial s_1 ds_1 + \partial f / \partial s_2 ds_2 + \partial f / \partial \theta d\theta$ , simple calculus leads to the following formula for the relative change in  $f$ .

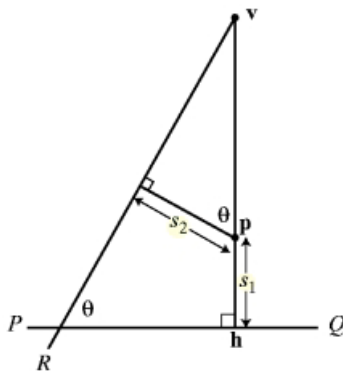
$$df/f = 1/2 (ds_1/s_1 + ds_2/s_2 + \tan \theta d\theta) \quad (3)$$

From this we see that  $f$  is sensitive to small changes in the measurements of  $s_1$ ,  $s_2$  and  $\theta$  when  $s_1$  or  $s_2$  are small, or  $\theta$  is close to  $90^\circ$ . In particular, it is impossible to estimate  $f$  accurately when the principal point is close to the horizon line, since in this case  $s_1$  is small, and the angle  $\theta$  will be close to  $90^\circ$ .

*Dependence of  $f$  on the principal point.* It was seen above that if the horizon and apex are identifiable in an image, then the principal point position may vary with one degree of freedom, along the line through the apex perpendicular to the horizon. If there is only one vertical line feature visible in the image, then the apex may not be identified, but instead may lie anywhere along the image of the vertical line. The principal point may now vary with two degrees of freedom anywhere in the image. Figure 5 gives an example of such an image. For a choice of the principal point, the focal length may be computed

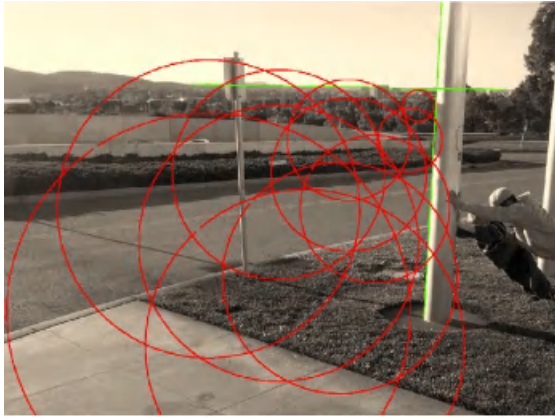


**Fig. 3.** This image shows the computation of the focal length of the camera assuming a principal point at the centre of the image. The original image is shown to the left, and the construction to the right. Since the horizon line (blue) is not directly visible in this image, it is computed as the line joining two horizontal vanishing points, computed from the right-hand wall and the grid on the door (green lines). Note that these are not orthogonal vanishing points. The vertical vanishing point is given by the sides of the door and the door frame (red lines). The red circle is the  $45^\circ$  circle, with radius  $f$ , assuming the principal point is placed as shown (at the centre of the red circle).



**Fig. 4.** Since  $d(\mathbf{p}, \mathbf{v}) = s_2 / \cos \theta$ , it follows from (1) that  $f^2 = s_1 s_2 / \cos \theta$ .

using (2), or by direct geometric construction, as shown previously. However, because of the added degree of freedom, the computed focal length depends more dramatically on the assumed position of the principal point. It may vary from near zero, when the principal point is assumed to be close to the intersection of the horizon and vertical line, to large values when the principal point is far from both these lines.



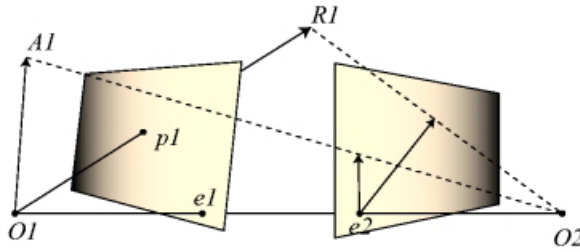
**Fig. 5.** In this image, the apex is unknown, but is constrained to lie somewhere along the vertical direction shown in the image (green). The principal point has two degrees of freedom, but once a position of the principal point is assumed, the focal length of the camera may be computed. The red circles represent various estimates of the focal length for different assumed principal points. Each circle is the locus of all directions  $10^\circ$  from the principal point, which is at the centre of the circle. Thus, the radius of each circle is equal to  $0.176f = \tan(10^\circ)f$ . The focal length has been computed in each case using (2).

### 3 Calibration from Two Views

Next, the calibration methods for a single view discussed above will be extended to consideration of calibration from two views. This problem was first addressed in [5] where an algorithm for computing the two focal lengths was given. This algorithm was quite complicated, and the problem was later considered by several authors. The most compact solution was found by Bougnoux in [2], where a simple formula was given for the two focal lengths in terms of the fundamental matrix and the two principal points (see equation (5) below). This formula will be rederived here in a way that sheds light on its geometric meaning, and its sensitivity to variations in the assumed positions of the principal point. It will be shown that Bougnoux's formula is closely related to (2), and has a simple geometric interpretation.

First, we note that in order to compute the focal lengths, the positions of the two principal points need to be known. The geometry of two views may be summarized in Fig. 6. The two camera centres  $O_1$  and  $O_2$  are joined by a baseline  $O_1O_2$  and the view direction of each camera is defined by the principal rays,  $R_1$  and  $R_2$  of the cameras. Of most interest is the case where these three lines are not coplanar; in other words, the principal rays of the two cameras do not meet. The principal ray  $R_1$  and the baseline  $O_1O_2$  define a plane, which will be denoted by  $R_1O_1O_2$ . In addition, let  $A_1$  be the ray through the camera centre  $O_1$  perpendicular to the plane  $R_1O_1O_2$ . This plane, and the ray  $A_1$  form an orthogonal pair in space. If their projections into the second image can be identified, then we may use the methods of section 2 to compute the focal length of the second camera.





**Fig. 6.** The plane defined by the principal ray  $R_1$  and the base-line  $O_1O_2$  is imaged as an epipolar line in the second (right-hand) image. The ray  $A_1$  is perpendicular to this plane, and also projects to an epipolar line in the second image. The two epipolar lines are the images of an orthogonal plane/line pair. Assuming a position for the principal point  $\mathbf{p}_2$ , the focal length  $f_2$  of the second camera may be computed using the method of section 2.

First of all, consider the first (left) image taken from camera centre  $O_1$ . We want to identify the projection in this image of the plane  $R_1O_1O_2$  and the ray  $A_1$ . The projection of the plane is easily identified as the epipolar line through the principal point, namely  $[\mathbf{e}_1] \times \mathbf{p}_1 = \mathbf{e}_1 \times \mathbf{p}_1$ . Since the ray  $A_1$  is perpendicular to the plane containing the principal ray, it projects to a point at infinity in the image, in the direction perpendicular to the line  $[\mathbf{e}_1] \times \mathbf{p}_1$ . This infinite point may be written as  $\mathbf{I}(\mathbf{e}_1 \times \mathbf{p}_1)$ , or  $\mathbf{I}[\mathbf{e}_1] \times \mathbf{p}_1$ , where  $\mathbf{I}$  is the  $3 \times 3$  matrix  $\text{diag}(1, 1, 0)$ .

To explain this last point more fully, note that  $\mathbf{I}\mathbf{l}$  represents the point at infinity in the direction perpendicular to  $\mathbf{l}$  for any line  $\mathbf{l}$ . For example,  $(a, b)$  is the vector perpendicular to the line  $ax + by + c = 0$ , represented by coordinates  $(a, b, c)^T$ , and  $(a, b, 0)$  is the vanishing point in this direction. This relation will be used often in this paper, so we emphasize it:

**Proposition 3.1.**

1. If  $\mathbf{l}$  is a line, then  $\mathbf{I}\mathbf{l}$  represents the point at infinity in the direction perpendicular to the line.
2. If line  $\mathbf{l}$  passes through point  $\mathbf{e}$ , then  $[\mathbf{e}] \times \mathbf{I}\mathbf{l}$  is the line through  $\mathbf{e}$  perpendicular to  $\mathbf{l}$ .

It is now easy to compute the epipolar lines in the second image corresponding to the plane  $R_1O_1O_2$  and the ray  $A_1$ . This is done simply by transferring the epipolar lines from the first image to the second, using the fundamental matrix. First, consider the vanishing line of the plane  $R_1O_1O_2$ . Since this plane passes through the camera centre  $O_2$ , it is viewed edge-on in the second image. Consequently, its vanishing line is nothing more than the image of any line lying in the plane  $R_1O_1O_2$ . Since the principal ray of the first camera is just such a line we see:

**3.2.** The vanishing line in the second image of the plane  $R_1O_1O_2$  is the epipolar line  $\mathbf{F}\mathbf{p}_1$  corresponding to an epipolar line  $[\mathbf{e}_1] \times \mathbf{p}_1$  in the first image. Here  $\mathbf{F}$  is the fundamental matrix.

Secondly, the image of the ray  $A_1$  is easily computed by transferring its vanishing point using the fundamental matrix:

**3.3.** The image of the ray  $A_1$  in the second image is the epipolar line  $\mathbf{FI}[\mathbf{e}_1] \times \mathbf{p}_1$ . This corresponds to the epipolar line in the first image, perpendicular to  $[\mathbf{e}_1] \times \mathbf{p}_1$ .

Thus, we may conclude:

**3.4.** Given the fundamental matrix  $\mathbf{F}$  for two images, and assumed positions of the principal points  $\mathbf{p}_1$  and  $\mathbf{p}_2$  in the two images, the focal length  $f_2$  may be computed as follows.

1. In the first image, the epipolar line through the principal point  $\mathbf{p}_1$  is given by  $[\mathbf{e}_1] \times \mathbf{p}_1$ . The epipolar line perpendicular to this is  $[\mathbf{e}_1] \times \mathbf{I}[\mathbf{e}_1] \times \mathbf{p}_1$ .
2. The two corresponding epipolar lines in the second image are  $\mathbf{Fp}_1$  and  $\mathbf{FI}[\mathbf{e}_1] \times \mathbf{p}_1$  respectively.
3. Apply equation (2) where  $s_1$  and  $s_2$  are the distances from the principal point  $\mathbf{p}_2$  to these two lines, and  $\theta$  is the angle between them – more specifically, the angle of the segment containing  $\mathbf{p}_2$ .

This construction is illustrated in Fig. 7. An explicit formula is possible. Let  $\mathbf{l}$  and  $\mathbf{l}'$  be the two epipolar lines computed in (3.4). The distance from a point  $\mathbf{p} = (x, y, 1)^\top$  to a line  $\mathbf{l}$  is  $\mathbf{l}^\top \mathbf{p} / \alpha$  where  $\alpha = (l_1^2 + l_2^2)^{1/2}$ , and  $l_i$  here represents the  $i$ -th component of the vector  $\mathbf{l}$ . Furthermore, the angle between two lines  $\mathbf{l}$  and  $\mathbf{l}'$  satisfies the relation  $\cos \theta = -(\mathbf{l}'^\top \mathbf{l}) / \alpha \alpha'$ . Now, writing (2) in terms of the homogeneous coordinates for the two lines  $\mathbf{l}$  and  $\mathbf{l}'$  involved, yields

$$f^2 = -\frac{(\mathbf{l}'^\top \mathbf{p})(\mathbf{l}^\top \mathbf{p})}{\mathbf{l}^\top \mathbf{l}'} \quad (4)$$

since the factors  $\alpha$  and  $\alpha'$  cancel top and bottom.

Substituting the formulas (3.2) and (3.3) for the lines  $\mathbf{l}_2$  and  $\mathbf{l}'_2$  finally gives the formula

$$f_2^2 = -\frac{(\mathbf{p}_1^\top [\mathbf{e}_1] \times \mathbf{IF}^\top \mathbf{p}_2)(\mathbf{p}_1^\top \mathbf{F}^\top \mathbf{p}_2)}{\mathbf{p}_1^\top ([\mathbf{e}_1] \times \mathbf{IF}^\top \mathbf{IF}) \mathbf{p}_1} \quad (5)$$

In this formula,  $\mathbf{e}_1$  is the epipole in the first image, and  $\mathbf{p}_1$  and  $\mathbf{p}_2$  are homogeneous 3-vectors representing the principal points in the two images. It is required however, that  $\mathbf{p}_2 = (\cdot, \cdot, 1)^\top$ , with last coordinate equal to 1.

The formula for the first focal length  $f_1$  may be obtained by interchanging the role of the first and second cameras in this formula.

## 4 Effect of Varying the Principal Points

We consider now the effect of varying the principal point in either or both the images. Without loss of generality, let us restrict attention to (5), and computation of  $f_2^2$ . Since the left hand side of (5) is the square of the focal length, it is necessary that the right hand side be positive, otherwise, no solution exists. To analyze this further, we note that the expression for  $f_2^2$  may be split into parts as follows:

$$\begin{aligned} A(\mathbf{p}_1, \mathbf{p}_2) &= \mathbf{p}_1^\top [\mathbf{e}_1] \times \mathbf{IF}^\top \mathbf{p}_2 \\ B(\mathbf{p}_1, \mathbf{p}_2) &= \mathbf{p}_1^\top \mathbf{F}^\top \mathbf{p}_2 \\ D(\mathbf{p}_1) &= \mathbf{p}_1^\top ([\mathbf{e}_1] \times \mathbf{IF}^\top \mathbf{IF}) \mathbf{p}_1 \end{aligned}$$



**Fig. 7.** A pair of images of the Grenoble museum from which one wishes to compute the focal lengths. It is assumed that the principal point  $\mathbf{p}_1$  in the first image is the centre of the image. Shown overlaid on the first (left-hand) image are the epipolar line  $[\mathbf{e}_1] \times \mathbf{p}_1$  and the perpendicular epipolar line  $[\mathbf{e}_1] \times \mathbf{I}[\mathbf{e}_1] \times \mathbf{p}_1$ . Overlaid on the second (right-hand) image are the corresponding epipolar lines  $\mathbf{Fp}_1$  and  $\mathbf{FI}[\mathbf{e}_1] \times \mathbf{p}_1$ . From these latter two lines one may compute the focal length  $f_2$ . Computation of the focal length  $f_2$  will be very imprecise in this case, since the angle  $\theta$  between these two lines is close to  $90^\circ$ , and also because one of the lines (actually  $\mathbf{Fp}_1$ ) passes very close to the centre of the image, and hence (presumably) close to the principal point  $\mathbf{p}_2$ . Relative accuracy of the focal length  $f_2$  is given by (3), and is proportional to the distance of  $\mathbf{p}_2$  from each of these two epipolar lines, and also to the tangent of the angle  $\theta$ .

At the right is shown the focal length  $f_2$  as a function of the principal point position  $\mathbf{p}_2$ . The focal length is represented by the intensity of the image. For a principal point in the black region of the image, the estimated value of  $f_2^2$  is negative, meaning that the assumptions on position of the principal point are not viable (impossible situation).

Thus,  $f_2^2 = -A(\mathbf{p}_1, \mathbf{p}_2)B(\mathbf{p}_1, \mathbf{p}_2)/D_2(\mathbf{p}_1)$ . Note that the estimated value of the focal length becomes zero where the numerator (that is, either  $A$  or  $B$ ) vanishes, and goes to infinity where the denominator  $D$  vanishes. The value of  $f_2^2$  will change sign on the union of the vanishing sets of  $A$ ,  $B$  and  $D$ . We look at each of these terms independently.

#### 4.1 Varying the Principal Point $\mathbf{p}_2$

This situation has been explored in section 3. The denominator  $D(\mathbf{p}_1)$  does not depend on  $\mathbf{p}_2$ , and so is constant. We look at the vanishing sets of the two terms  $A(\mathbf{p}_1, \mathbf{p}_2)$  and  $B(\mathbf{p}_1, \mathbf{p}_2)$ . As discussed above,  $\mathbf{Fp}_1$  and  $\mathbf{FI}[\mathbf{e}_1] \times \mathbf{p}_1$  represent the epipolar lines in the second image, corresponding to the principal ray in the first image, and a ray perpendicular to it. When  $\mathbf{p}_2$  lies on one of these epipolar lines, the numerator of (5) vanishes. Consequently, the zero set of  $A(\mathbf{p}_1, \mathbf{p}_2)B(\mathbf{p}_2, \mathbf{p}_2)$  for fixed  $\mathbf{p}_1$  and varying  $\mathbf{p}_2$  consists of a pair of epipolar lines in the second image. The image plane is divided into four sections by these epipolar lines and the principal point  $\mathbf{p}_2$  must lie in the acute-angled region of the plane. This is illustrated in Fig. 7.

It may be noted that here and elsewhere (particularly in Fig. 9) the most probable principal point (image centre) sometimes lies in the dark part of the diagram, indicating an imaginary focal length. This is because when the principal ray passes near the centre of the image, the angle between the two lines dividing the image is necessarily close to  $90^\circ$ . If because of the instability of the computed fundamental matrix it falls on the

wrong side of  $90^\circ$ , then the regions may swap sign, and lead to impossible values for the focal length. In these examples the fundamental matrix was computed using a Maximum Likelihood bundle-adjustment method, which should give optimal results – nevertheless this unfavourable situation can occur. A method that avoids this problem by placing *a priori* conditions on the positions of the principal point and (optionally) the focal lengths was presented in [7]. It uses Bougnoux's formula to constrain the values of the focal length and principal points while finding a compatible fundamental matrix.

## 4.2 Varying the Principal Point $\mathbf{p}_1$

Next we consider the effect of varying the principal point  $\mathbf{p}_1$  on the estimate of the focal length  $f_2$ . In this section we will assume that the principal point  $\mathbf{p}_2$  of the second camera is known and fixed at a given value. As  $\mathbf{p}_1$  varies, formula (5) give an estimate for the focal length  $f_2$ .

*Vanishing set of  $A(\mathbf{p}_1, \mathbf{p}_2)B(\mathbf{p}_1, \mathbf{p}_2)$ .* Since  $F$  is the fundamental matrix,  $F^\top \mathbf{p}_2$  is the epipolar line in the first image corresponding to the point  $\mathbf{p}_2$ . Thus the set of points  $\mathbf{p}_1$  at which  $B(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{p}_1^\top F^\top \mathbf{p}_2$  vanishes is precisely the set of points on the epipolar line  $F^\top \mathbf{p}_2$ .

On the other hand, according to Proposition 3.1,  $[\mathbf{e}_1]_\times IF^\top \mathbf{p}_2$  is the line perpendicular to  $F^\top \mathbf{p}_2$ . Hence  $A(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{p}_1^\top [\mathbf{e}_1]_\times IF^\top \mathbf{p}_2$  vanishes when  $\mathbf{p}_1$  lies on the epipolar line perpendicular to  $F^\top \mathbf{p}_2$ .

**4.1.** For a given value of  $\mathbf{p}_2$ , the set of points  $\mathbf{p}_1$  such that the numerator  $A(\mathbf{p}_1, \mathbf{p}_2)B(\mathbf{p}_1, \mathbf{p}_2)$  of (5) vanishes consists of the epipolar line  $F^\top \mathbf{p}_2$  and the epipolar line perpendicular to it.

*Vanishing set of  $D$ .* We now turn to the denominator of (5), namely  $D(\mathbf{p}_1) = \mathbf{p}_1^\top ([\mathbf{e}_1]_\times IF^\top IF) \mathbf{p}_1$ . Note that this is independent of  $\mathbf{p}_2$ . From the form of this expression, the vanishing set is a conic. It will be shown that in fact it is a degenerate conic, consisting of two perpendicular lines through the epipole  $\mathbf{e}_1$ .

First of all one observes (by simple substitution and using the fact that  $F\mathbf{e}_1 = 0$ ) that the epipole  $\mathbf{e}_1$  belongs to this vanishing set. If  $\mathbf{p}_1$  is any point such that  $D(\mathbf{p}_1) = 0$ , then  $D(\mathbf{p}_1 + \alpha \mathbf{e}_1) = 0$  for any  $\alpha$ , and hence the vanishing set is made up of lines passing through  $\mathbf{e}_1$ , epipolar lines.

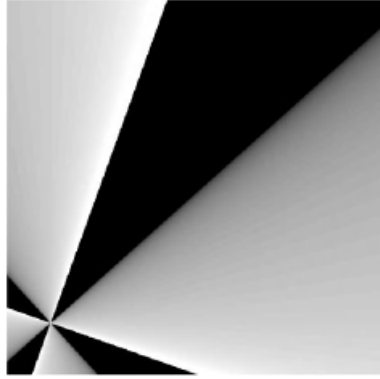
Now let  $\mathbf{p}_1$  be a point at infinity, and let  $\mathbf{p}'_1$  be the point at infinity on the epipolar line normal to  $[\mathbf{e}_1]_\times \mathbf{p}_1$ . By symmetry,  $\mathbf{p}''_1 = (\mathbf{p}'_1)' = \mathbf{p}_1$ . By Proposition 3.1,  $\mathbf{p}'_1 = I[\mathbf{e}_1]_\times \mathbf{p}_1$ , and so  $D(\mathbf{p}_1) = \mathbf{p}_1'^\top (F^\top IF) \mathbf{p}_1$ . Substituting  $\mathbf{p}'_1$  for  $\mathbf{p}_1$  in this expression gives

$$D(\mathbf{p}'_1) = \mathbf{p}_1''^\top (F^\top IF) \mathbf{p}'_1 = \mathbf{p}_1^\top (F^\top IF) \mathbf{p}'_1 = \mathbf{p}_1'^\top (F^\top IF) \mathbf{p}_1 \text{ (by transposing)} = D(\mathbf{p}_1) .$$

Thus,  $D(\mathbf{p}'_1)$  vanishes if and only if  $D(\mathbf{p}_1)$  does. Consequently, the zero set of  $D(\mathbf{p}_1)$  consists of a pair of orthogonal epipolar lines. Summarizing this complete discussion.

**4.2.** For a given value of  $\mathbf{p}_2$ , the set of points  $\mathbf{p}_1$  for which the estimated value of  $f_2^2$  changes sign consists of two pairs of orthogonal epipolar lines in image 1. The value of  $f_2^2$  tends to zero at one pair of lines, and to infinity at the other pair of lines.

This is illustrated in Fig. 8.



**Fig. 8.** Assuming  $\mathbf{p}_2$  is fixed, and  $\mathbf{p}_1$  is varying, the singular set for estimation of  $f_2$  consists of points  $\mathbf{p}_1$  lying on two pairs of orthogonal epipolar lines. The estimated value of  $f_2$ , according to (5) tends to 0 and  $\infty$  as  $\mathbf{p}_1$  approaches the alternating epipolar lines. The value of  $f_2$  is only defined for  $\mathbf{p}_1$  in alternating sectors of the image as shown, since  $f_2^2$  is negative for  $\mathbf{p}_1$  in the other (dark) sectors.

## 5 Varying both Principal Points

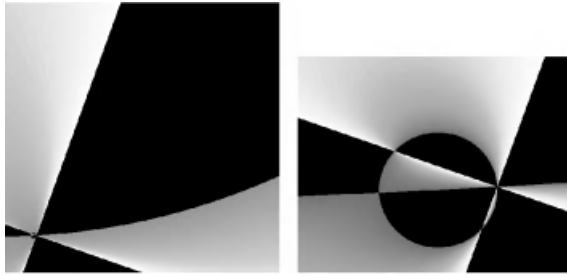
The previous sections have assumed that one of the principal points was fixed, and the other was allowed to vary. In some instances it may be more realistic to assume that the principal point is the same in both images, albeit not known exactly. Therefore, in this section, we assume that  $\mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}$ , and examine how the focal length estimate depends on the position of this common principal point. In this case, the formula for  $f_2$  is as follows.

$$f_2^2 = -\frac{(\mathbf{p}^\top [\mathbf{e}_1]_\times \mathbf{I} \mathbf{F}^\top \mathbf{p})(\mathbf{p}^\top \mathbf{F}^\top \mathbf{p})}{\mathbf{p}^\top ([\mathbf{e}_1]_\times \mathbf{I} \mathbf{F}^\top \mathbf{I} \mathbf{F}) \mathbf{p}} = -\frac{A(\mathbf{p}, \mathbf{p})B(\mathbf{p}, \mathbf{p})}{D(\mathbf{p})}. \quad (6)$$

We have seen already that the zero-set of  $D(\mathbf{p})$  consists of a pair of orthogonal epipolar lines through the epipole  $\mathbf{e}_1$ . The zero sets of the expressions  $A(\mathbf{p}, \mathbf{p}) = \mathbf{p}^\top \mathbf{F}^\top \mathbf{p}$  and  $B(\mathbf{p}, \mathbf{p}) = \mathbf{p}^\top [\mathbf{e}_1]_\times \mathbf{I} \mathbf{F}^\top \mathbf{p}$  are conics. It is easy to see that each of these expressions vanishes at both epipoles  $\mathbf{e}_1$  and  $\mathbf{e}_2$ , hence the conics pass through the two epipoles. In addition, it may be verified that the tangents to these two conics at the epipole  $\mathbf{e}_1$  are orthogonal. To see that, recall that the tangent at a point  $\mathbf{x}$  on a conic defined by matrix  $\mathbf{C}$  is the line  $(\mathbf{C} + \mathbf{C}^\top)\mathbf{x}$ . Applying this, one finds that the tangents to these two conics at the point  $\mathbf{e}_1$  are  $(\mathbf{F}^\top + \mathbf{F})\mathbf{e}_1 = \mathbf{F}^\top \mathbf{e}_1$  and  $([\mathbf{e}_1]_\times \mathbf{I} \mathbf{F}^\top - \mathbf{F} \mathbf{I} [\mathbf{e}_1]_\times)\mathbf{e}_1 = [\mathbf{e}_1]_\times \mathbf{I} \mathbf{F}^\top \mathbf{e}_1$ , which according to Proposition 3.1 are orthogonal lines through  $\mathbf{e}_1$ . Summarizing this gives the following result.

**5.1.** As the position of the principal point  $\mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}$  varies, the singular set where the value of the focal length  $f_2^2$  changes sign has the following components.

1. Two orthogonal epipolar lines passing through the epipole  $\mathbf{e}_1$ . As the assumed position of  $\mathbf{p}$  approaches these lines, the value of  $f_2^2$  tends to infinity.



**Fig. 9.** Museum images : the geometry of the allowable positions of the principal point, assuming that it is the same in both images (shown at left). The plane is divided into regions by three conics (one of them degenerate). Black regions represent impossible locations for the principal point. Inside the allowable (white) regions, the focal length approaches infinity at the pair of orthogonal lines (degenerate conic), and zero at the non-degenerate conics. At right is an enlargement of the bottom left-hand corner, showing the conic corresponding to the denominator of (6).

2. Two conics passing through the two epipoles  $\mathbf{e}_1$  and  $\mathbf{e}_2$ . The conics are orthogonal at  $\mathbf{e}_1$ . As  $\mathbf{p}$  approaches these conics, the estimated value of  $f_1^2$  tends to zero.

These lines divide the plane into regions. The value of  $f_2^2$  is negative on alternate regions, which therefore represent impossible positions for the principal point.

For an illustration of this, see Fig. 9.

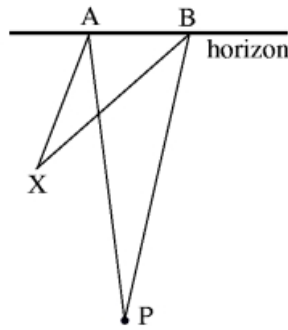
## 6 Reconstruction Accuracy

Previous sections have demonstrated that the camera calibration, particularly the focal length, depends closely on the assumed position of the principal point. It has often been objected that this may be true, but it does not affect the reconstructed scene. This belief will be shown to be unsafe by an analysis of how angles in the reconstructed scene vary with the principal point and focal length estimates. The dependency of scene angles on calibration parameters is well known, and was noted in [2]. Here it will be shown that the effect can be quite marked, and in particular a simple method of visualizing the effect will be given.

A simple method of measuring some angles in a reconstructed scene utilizes a concept called the *conformal point* in an image. This concept was introduced in [6] where its application to visual odometry was demonstrated. The conformal point is defined relative to a plane of interest in the scene.

- The conformal point relative to any plane in the scene is located a distance  $(f^2 + d^2)^{1/2}$  from the vanishing line of the plane (the horizon), along a perpendicular to the horizon through the principal point. Here  $f$  is the focal length of the camera, and  $d$  is the distance in the image from the principal point to the horizon.

Note that the position of the conformal point is directly dependent on the focal length and the position of the principal point. Note that in Fig. 2 the conformal point is located along the line  $\mathbf{h}\mathbf{v}$  a distance from  $\mathbf{h}$  equal to the length of the chord  $\mathbf{a}\mathbf{h}$ .



**Fig. 10.** To measure an angle between two lines in the image, the lines are extended to meet the horizon at points A and B. The true angle between the two lines is equal to the angle  $\angle APB$ , where P is the conformal point. The position of the conformal point is explained in the text.

The interesting property of the conformal point is illustrated in Fig. 10, showing how it may be used to measure angles between lines in a plane. A proof of the correctness of this method is given in [6]. We apply this method of angle-measurement to the image shown in Fig. 3, and use it to estimate the angle between the door and the wall. According to the construction of Fig. 10, the required angle is the angle subtended at the conformal point by the vanishing points of the two directions under consideration. Note that in this example these two vanishing points are the two points (intersection of the green lines) used to construct the horizon line. The conformal point may be easily constructed assuming positions of the principal point of the image at the top, centre and bottom of the image along the vertical line. The estimated angles are

1. For principal point at the top of the image :  $88^\circ$ .
2. For principal point at the centre of the image :  $69^\circ$ .
3. For principal point at the bottom of the image :  $56^\circ$ .

Thus in this example at least computed angles in the scene (properties of a Euclidean reconstruction based on a computed calibration) are very dependent on the assumed position of the principal point.

Beyond illustrating the variation of estimated scene geometry, this method gives a quick intuitive method of evaluating the degree of variation as a function of assumed principal point. This allows an operator to ascertain the effect of an incorrect placement of the principal point.

## 7 Conclusion

We have argued that the assumed position of the principal point may have a large effect on the estimated focal length of the cameras, and also the 3D reconstruction accuracy, for certain single-view and two-view calibration scenarios. The two-view calibration method analyzed here has been shown previously ([9]) to have degenerate configurations when the principal rays of the cameras meet. We have given here a simple analysis of how the

quality of the focal length estimate degenerates when the principal ray of one camera lies close to the epipolar line corresponding to the other principal point. It is shown that in fact, the relative uncertainty in the the focal length estimated by this method is inversely proportional to the distance of the principal point to this epipolar line. Because of this sensitivity, the practicality of estimating focal lengths from two views is doubtful. However, in separately published work ([7]) we have proposed a method for mitigating some of the instabilities involved in estimating the principal point and focal length by assigning a priori bounds and variances for their values.

The method and analysis of this paper apply to other imaging scenarios, such as planar motion ([1]) in which it is also possible to compute the image positions of the horizon and apex. A new interpretation of Bougnoux's focal length formula, is also given here, relating it to the geometry of the principal rays.

## References

- [1] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In *Proc. 4th European Conference on Computer Vision, Cambridge*, LNCS 1064/5, pages 3–16. Springer-Verlag, 1996.
- [2] S. Bougnoux. From Projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 790–796, January 1998.
- [3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4:127–140, 1990.
- [4] R. Cipolla, D. P. Robertson, and E. G. Boyer. Photobuilder – 3D models of architectural scenes from uncalibrated images. In *Proc. IEEE International Conference on Multimedia Computing and Systems*, volume I, pages 25 – 31, June 1999.
- [5] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proc. 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, LNCS 588, pages 579–587. Springer-Verlag, 1992.
- [6] R. I. Hartley and C. Silpa-Anan. Visual navigation in a plane using the conformal point. In *International Symposium on Robotics Research*, pages to appear – available at <http://www.anu.edu.au/hartley>, 2001.
- [7] R. I. Hartley and C. Silpa-Anan. Reconstruction from two views using approximate calibration. In *ACCV*, pages 338–343, 2002.
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [9] G. Newsam, D. Q. Huynh, M. Brooks, and H. P. Pan. Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. In *Int. Arch. Photogrammetry & Remote Sensing*, volume XXXI-B3, pages 575–80, Vienna, 1996.
- [10] M. Pollefeys, R. Koch, and L. Van Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 90–96, 1998.



# Critical Curves and Surfaces for Euclidean Reconstruction

Fredrik Kahl<sup>(1)</sup> and Richard Hartley<sup>(2)</sup>

<sup>(1)</sup>Centre for Mathematical Sciences, Lund University, Lund, Sweden

`fredrik@maths.lth.se`

<sup>(2)</sup>RSISE, Australian National University, Canberra, Australia

`Richard.Hartley@anu.edu.au`

**Abstract.** The problem of recovering scene structure and camera motion from images has a number of inherent ambiguities. In this paper, configurations of points and cameras are analyzed for which the image points alone are insufficient to recover the scene geometry uniquely. Such configurations are said to be critical. For two views, it is well-known that a configuration is critical only if the two camera centres and all points lie on a ruled quadric. However, this is only a necessary condition. We give a complete characterization of the critical surfaces for two calibrated cameras and any number of points. Both algebraic and geometric characterizations of such surfaces are given. The existence of critical sets for  $n$ -view projective reconstruction has recently been reported in the literature. We show that there are critical sets for  $n$ -view Euclidean reconstruction as well. For example, it is shown that for any placement of three calibrated cameras, there always exists a critical set consisting of any number of points on a fourth-degree curve.

## 1 Introduction

Early in the twentieth century it was noticed that for two cameras and any number of scene points all lying on a particular surface, the solution to the reconstruction problem is not unique. There are several ambiguous configurations of cameras and scene points, which produce identical images. In the German photogrammetry community, such surfaces were termed ‘gefährliche Örter’ [7], which is translated as critical surfaces. In this paper, such critical configurations are analyzed for two and more calibrated cameras.

For two views it is well-known that a configuration is critical only if all points and the two camera centres lie on a ruled quadric, that is, a hyperboloid of one sheet, or one of its degenerate versions [8]. In [9], it was proven that six points and any number of cameras lying on a ruled quadric are critical. This was shown to be the dual to the two-view case [5] in the Carlsson duality sense [2]. Recently, the existence of critical configurations with more than two views and arbitrarily many points has been reported. In [4], it was shown that for any three cameras, there exists a critical fourth-degree curve containing the camera centres. The extension to arbitrarily many views was done in [6] where it was shown that

cameras centres and scene points lying on the intersection of two quadrics are critical for projective reconstruction. However, critical configurations for more than two *calibrated* views have previously not been investigated in the published literature (except in the case of lines [1]).

In this paper, we focus on critical configurations for Euclidean reconstruction, starting with the calibrated two-view case. The most complete investigation of this is Maybank's book [8], to which the present paper owes much. A basic result of [8] is that a critical surface is necessarily a ruled, rectangular quadric, but the question of sufficiency is not substantially considered there. In the present paper, we obtain a necessary and sufficient condition for a configuration to be critical. The key is a restriction on the possible positions of the camera centres on the critical quadric. The condition is related to an algebraic constraint obtained by Maybank, but formulated here in a simple geometric form. The two-view analysis is used as a basis for exploring Euclidean multiview critical configurations. Previously unknown examples of critical configurations for more than two views are exhibited.

## 2 Background

A camera is represented by a  $3 \times 4$  matrix of rank 3, the camera matrix  $P$ . Provided the camera centre (which is the generator of the right null-space of  $P$ ) is a finite point, the camera matrix may be decomposed as  $P = K[R | t]$ , where  $R$  is a rotation matrix and  $K$  is an upper-triangular matrix, the calibration matrix. If  $K$  is known, then the camera is called *calibrated*. Matrix  $K$  represents a coordinate transformation in the image, this transformation may be undone by multiplying image coordinates by  $K^{-1}$ . The camera matrix may therefore be assumed to be of the form  $P = [R | t]$ , which will be referred to as a *calibrated camera*.

Consider a configuration of  $n \geq 2$  cameras and a set of points. Denote the camera matrices by  $P^i$  for  $i = 0, \dots, n - 1$  and the set of points with  $P_j$ . The set  $\{P^i, P_j\}$  is said to be *critical* if there exists an inequivalent configuration of cameras  $Q^i$  and points  $Q_j$  such that  $P^i P_j = Q^i Q_j$  for all  $i$  and  $j$ . Two configurations are considered to be equivalent if the essential matrices for all pairs of views are the same.<sup>1</sup> The alternative configuration  $\{Q^i, Q_j\}$  is called a *conjugate configuration*. In a critical configuration it is not possible to recover the cameras and the scene points unambiguously from the image points alone, as there are two alternative solutions. Note that when considering critical configurations for calibrated cameras all camera matrices  $P^i$  and  $Q^i$  are required to be of the form  $[R | t]$ , which is quite a restrictive condition. Consequently critical configurations for calibrated cameras may be expected to be a subset of uncalibrated critical configurations.

For two calibrated cameras, the relative pose is encapsulated by the essential matrix. We define an *essential matrix* to be a  $3 \times 3$  matrix writable as a product of

<sup>1</sup> This definition excludes the trivial ambiguity that arises from points lying on a line containing all camera centres, as such points may vary along this base line. In addition, it excludes the two-view "twisted-pair ambiguity" discussed later – such an ambiguity being deemed in some sense trivial also.

a rotation and a skew-symmetric matrix. Thus,  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_{\times}$ , where  $\mathbf{R}$  is a rotation, and  $[\mathbf{t}]_{\times}$  denotes the matrix such that  $[\mathbf{t}]_{\times}\mathbf{v} = \mathbf{t} \times \mathbf{v}$  for any 3-vector  $\mathbf{v}$ .

### 3 Critical Sets for Euclidean Reconstruction

The basic equation for the shape of a critical set in the calibrated (Euclidean) case is not very different from that for the uncalibrated (projective) case. Following the derivation in [4] one arrives at the following result.

**Theorem 1.** *Let  $(\mathbf{P}^0, \mathbf{P}^1)$  and  $(\mathbf{Q}^0, \mathbf{Q}^1)$  be two pairs of calibrated cameras, and let  $\mathbf{E}_{\mathbf{P}}$  and  $\mathbf{E}_{\mathbf{Q}}$  be the corresponding essential matrices for the two camera pairs. The surfaces  $S_{\mathbf{P}}$  and  $S_{\mathbf{Q}}$  defined by*

$$S_{\mathbf{P}} = \mathbf{P}^{0\top} \mathbf{E}_{\mathbf{Q}} \mathbf{P}^1 + \mathbf{P}^{1\top} \mathbf{E}_{\mathbf{Q}}^{\top} \mathbf{P}^0 \quad \text{and} \quad S_{\mathbf{Q}} = \mathbf{Q}^{0\top} \mathbf{E}_{\mathbf{P}} \mathbf{Q}^1 + \mathbf{Q}^{1\top} \mathbf{E}_{\mathbf{P}}^{\top} \mathbf{Q}^0 \quad (1)$$

*are ruled quadric surfaces<sup>2</sup>, which are critical for reconstruction. In particular:*

1. *If  $\mathbf{P}$  is a point on  $S_{\mathbf{P}}$ , then there exists a point  $\mathbf{Q}$  on  $S_{\mathbf{Q}}$  such that  $\mathbf{P}^i \mathbf{P} = \mathbf{Q}^i \mathbf{Q}$  for  $i = 0, 1$ .*
2. *Conversely, if  $\mathbf{P}$  and  $\mathbf{Q}$  are points such that  $\mathbf{P}^i \mathbf{P} = \mathbf{Q}^i \mathbf{Q}$  for  $i = 0, 1$ , then  $\mathbf{P}$  lies on  $S_{\mathbf{P}}$  and  $\mathbf{Q}$  lies on  $S_{\mathbf{Q}}$ .*

This leads us to the following definition.

**Definition 1.** *A triple  $(S, \mathbf{P}^0, \mathbf{P}^1)$  where  $S$  is a symmetric  $4 \times 4$  matrix representing a quadric, and  $\mathbf{P}^0$  and  $\mathbf{P}^1$  are calibrated camera matrices representing cameras with centres lying on the quadric  $S$  is called critical for Euclidean reconstruction if there exists an essential matrix  $\mathbf{E}$  such that*

$$S = \mathbf{P}^{0\top} \mathbf{E} \mathbf{P}^1 + \mathbf{P}^{1\top} \mathbf{E}^{\top} \mathbf{P}^0. \quad (2)$$

**The twisted-pair ambiguity.** In Definition 1, matrix  $\mathbf{E}$  is the essential matrix for a conjugate camera pair  $(\mathbf{Q}^0, \mathbf{Q}^1)$ . Note however that (unlike in the projective case), the essential matrix  $\mathbf{E}$  does not determine the two camera matrices  $(\mathbf{Q}^0, \mathbf{Q}^1)$  uniquely even up to a similarity, because of the “twisted-pair ambiguity.” Thus, (apart from sign and scale ambiguities) there are two essentially different ways of decomposing  $\mathbf{E}$ , namely  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_{\times} = \mathbf{R}'[\mathbf{t}]_{\times}$ , involving different rotation matrices  $\mathbf{R}$  and  $\mathbf{R}'$  (see [8,3] for more details). Note that the vector  $\mathbf{t}$  is the same in both cases, since it is the generator of the null-space of  $\mathbf{E}$ .

**Normalized camera matrices.** As shown in [4], for a configuration of points and cameras to be critical it is sufficient to consider only the positions of the cameras, and not their orientation<sup>3</sup>. In particular, in investigating whether a configuration is critical, we may assume that the two cameras have the form  $[\mathbf{I} | -\mathbf{t}_i]$ , where  $\mathbf{t}_i$  is an inhomogeneous 3-vector representing the location of

<sup>2</sup> A quadric  $S$  is defined by the set of points  $\mathbf{P} \in \mathcal{P}^3$  such that  $\mathbf{P}^{\top} S \mathbf{P} = 0$ .

<sup>3</sup> Proved in [4] for the uncalibrated case, but easily extended to calibrated cameras.

the camera centre. Camera matrices in this form will be referred to as *normalized*. The form of a critical quadric is particularly simple for normalized camera matrices:

$$S = \begin{bmatrix} \mathbf{E}^{01} + \mathbf{E}^{10} & -\mathbf{E}^{10}\mathbf{t}_0 - \mathbf{E}^{01}\mathbf{t}_1 \\ -\mathbf{t}_0^\top \mathbf{E}^{01} - \mathbf{t}_1^\top \mathbf{E}^{10} & 2\mathbf{t}_0^\top \mathbf{E}^{01}\mathbf{t}_1 \end{bmatrix}. \quad (3)$$

## 4 Rectangular Quadrics

We now define *rectangular quadric* surfaces, which will turn out to be the critical surfaces for calibrated cameras.

**Definition 2.** A quadric represented by a symmetric matrix  $S$  is called a *rectangular quadric* if the upper left hand  $3 \times 3$  block  $\mathbf{M}$  of  $S$  may be written as  $\mathbf{E} + \mathbf{E}^\top$  for some essential matrix  $\mathbf{E}$ .

In [8] rectangular quadrics are characterized by different algebraic conditions on  $\mathbf{M}$ , which are next seen to be equivalent to Definition 2.

**Proposition 1.** Let  $\mathbf{M}$  be a  $3 \times 3$  symmetric matrix. The following conditions are equivalent.

1.  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$  for some essential matrix  $\mathbf{E}$ .
2.  $\mathbf{M} = \mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top - 2\mathbf{m}\mathbf{n}^\top \mathbf{I}$  for two 3-vectors  $\mathbf{m}$  and  $\mathbf{n}$  called the principal points on  $\mathbf{M}$ .
3. The eigenvalues of  $\mathbf{M}$  are of the form  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_1 + \lambda_2$ , where  $\lambda_1\lambda_2 \leq 0$ .

*Proof.* 1  $\rightarrow$  2 Suppose  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$ . Let  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times$ . Without substantially altering the problem,  $\mathbf{E}$  may be replaced by an essential matrix  $\mathbf{U}\mathbf{E}\mathbf{U}^\top$ , where  $\mathbf{U}$  is a rotation. Using this observation, one may without loss of generality assume that the rotation axis of  $\mathbf{R}$  is the vector  $\mathbf{n} = (0, 0, 1)^\top$ . Thus,  $\mathbf{R}$  is a rotation about the Z-axis. Let  $\mathbf{t} = (x, y, z)^\top$ . Then

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times = \begin{bmatrix} c-s & & \\ s & c & \\ & & 1 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} = \begin{bmatrix} -sz & -cz & cy+sx \\ cz & -sz & sy-cx \\ -y & x & 0 \end{bmatrix}$$

where  $c = \cos(\theta)$  and  $s = \sin \theta$  and  $\theta$  is the angle of rotation. So

$$\mathbf{E} + \mathbf{E}^\top = \begin{bmatrix} -2sz & 0 & sx + (c-1)y \\ 0 & -2sz & sy - (c-1)x \\ sx + (c-1)y & sy - (c-1)x & 0 \end{bmatrix}. \quad (4)$$

Now, setting  $\mathbf{m} = (sx + (c-1)y, sy - (c-1)x, sz)^\top$  and  $\mathbf{n} = (0, 0, 1)^\top$ , it is easily verified that  $\mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top - 2\mathbf{m}\mathbf{n}^\top \mathbf{I} = \mathbf{E} + \mathbf{E}^\top$  as required. In addition, apart from scaling, or swapping  $\mathbf{m}$  and  $\mathbf{n}$ , the choice of  $\mathbf{m}$  and  $\mathbf{n}$  is unique.

2  $\rightarrow$  1 Suppose  $\mathbf{M} = \mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top - 2\mathbf{m}\mathbf{n}^\top \mathbf{I}$ . Once again, one may rotate coordinates to ensure that  $\mathbf{n} = (0, 0, 1)^\top$ , and that furthermore  $\mathbf{m}$  lies in the xz-plane, and so  $\mathbf{m}$  is of the form  $\mathbf{m} = (2p, 0, q)^\top$ . Then

$$\mathbf{M} = \mathbf{m}\mathbf{m}^\top + \mathbf{n}\mathbf{n}^\top - 2\mathbf{m}^\top \mathbf{n} \mathbf{I} = 2 \begin{bmatrix} -q & p \\ & -q & 0 \\ p & 0 & 0 \end{bmatrix}. \quad (5)$$

Now, it is easily verified that if

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -q & -p \\ q & 0 & -p \\ p & p & 0 \end{bmatrix} = \begin{bmatrix} -q & 0 & p \\ 0 & -q & -p \\ p & p & 0 \end{bmatrix} \quad (6)$$

then  $\mathbf{E} + \mathbf{E}^\top$  equals (5) as required.

$2 \longleftrightarrow 3$ . See [8] for a proof. ■

Note that matrix  $\mathbf{M}$ , the upper-left  $3 \times 3$  block of  $S$  represents the conic in which the quadric  $S$  meets the plane at infinity. The principal points  $\mathbf{m}$  and  $\mathbf{n}$  lie on this conic, since  $\mathbf{m}^\top \mathbf{M} \mathbf{m} = \mathbf{n}^\top \mathbf{M} \mathbf{n} = 0$ . Points on  $\mathbf{M}$  may also be thought of as representing asymptotic directions of the quadric. When  $S$  is a ruled quadric, points on  $\mathbf{M}$  are the direction vectors of the generators (straight lines) on  $S$ .

**Restriction.** Henceforth in this paper, we will avoid having to deal with special cases by assuming that the matrix  $\mathbf{M}$  is non-singular, i.e.,  $S$  meets the plane at infinity in a non-degenerate conic. In terms of the representation of  $\mathbf{M}$  by principal points, this corresponds to an assumption that  $\mathbf{m}$  and  $\mathbf{n}$  represent neither collinear, nor orthogonal directions. Equivalently, in (5) neither  $p$  nor  $q$  is zero. It is shown in [8] that a rectangular quadric with two equal principal points is a circular cylinder, which can not be a critical surface.

Proposition 1 gives *algebraic* conditions for a quadric to be rectangular. Various equivalent geometric conditions for a quadric to be rectangular are given in the appendix. These help to provide geometric intuition.

## 5 Standard Position for a Rectangular Quadric

The definition of a rectangular quadric given in Definition 2 specifies only the form of the top-left block of the matrix  $S$ . In other words whether a quadric is rectangular or not depends only on its intersection with the plane at infinity. However, if  $\mathbf{M}$  is non-singular (which we are now assuming), then by a translation of coordinates,  $S$  may be transformed to a block-diagonal matrix of the form

$$S = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0}^\top & d \end{bmatrix}. \quad (7)$$

Note that this quadric is symmetric about the origin, in that if  $(\mathbf{X}^\top, k)^\top$  lies on the quadric  $S$ , then so does  $(-\mathbf{X}^\top, k)^\top$ . We may also assume, as before that the two principal points of the quadric are  $\mathbf{n} = (0, 0, 1)^\top$  and  $\mathbf{m} = (2p, 0, q)^\top$ . In this case, the matrix representing the quadric is of the form (7), where  $\mathbf{M}$  has the form given by (5). Such a quadric is said to be in *standard position*.

**Symmetry of a rectangular quadric.** A quadric in standard position has a rotational symmetry about the the  $Y$ -axis. Namely, if  $\mathbf{X} = (x, y, z)^\top$  lies on the quadric, then so does  $(-x, y, -z)^\top$ . There is a further symmetry mapping

$\mathbf{X}$  to  $(x, -y, z)^\top$ , as well as a symmetry swapping the two principal points, but we will not be so concerned with these. Let  $\mathbf{R}_{sym}$  represent this rotation of the quadric about the  $Y$ -axis. As a matrix,  $\mathbf{R}_{sym}$  has the form  $\text{diag}(-1, 1, -1)$ . For a quadric not in standard position, we still use  $\mathbf{R}_{sym}$  to represent this rotational symmetry of the quadric, though it will be an arbitrary rotation matrix.

**Different choices of  $\mathbf{E}$ .** Let  $S$  be a rectangular quadric and  $\mathbf{M}$  the upper left-hand block. By definition,  $\mathbf{M}$  can be written as  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$  for some essential matrix. The decomposition (6) involved a choice of  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times$  for which  $\mathbf{R}$  was a rotation through 90 degrees about the axis  $\mathbf{n} = (0, 0, 1)^\top$ . It will next be shown that this decomposition of the quadric is not unique. The following result enumerates all possible ways of expressing  $\mathbf{M}$  as  $\mathbf{E} + \mathbf{E}^\top$ .

**Terminology:** By left and right *epipoles* of an essential matrix  $\mathbf{E}^{01}$  are meant the vectors  $\mathbf{e}_0$  and  $\mathbf{e}_1$  such that  $\mathbf{E}^{01\top}\mathbf{e}_0 = \mathbf{E}^{01}\mathbf{e}_1 = \mathbf{0}$ .

**Theorem 2.** *Let  $\mathbf{M}$  be a non-singular matrix  $\mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top - 2\mathbf{m}^\top\mathbf{n}\mathbf{I}$ , where  $\mathbf{m}$  and  $\mathbf{n}$  are two principal points. Let  $\mathbf{E}$  be an essential matrix such that  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$ .*

1. *The left and right epipoles  $\mathbf{e}_0$  and  $\mathbf{e}_1$  of  $\mathbf{E}$  as well as the two principal points  $\mathbf{m}$  and  $\mathbf{n}$  lie on the conic  $\mathbf{M}$ .*
2. *If  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times = \mathbf{R}'[\mathbf{t}]_\times$  are the two distinct ways of decomposing  $\mathbf{E}$  into a rotation and skew-symmetric matrix, then the rotation axes of  $\mathbf{R}$  and  $\mathbf{R}'$  are the two principal points of  $\mathbf{M}$ .*
3. *For every point  $\mathbf{x}$  lying on the conic  $\mathbf{M}$ , with the exception of the two principal points, there exists a unique  $\mathbf{E}$  such that  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$  and  $\mathbf{E}\mathbf{x} = \mathbf{0}$ .*
4. *If  $\mathbf{e}_0$  is one of the epipoles of  $\mathbf{E}$ , then the other one,  $\mathbf{e}_1$  is the point  $\mathbf{R}_{sym}\mathbf{e}_0$  obtained by rotating  $\mathbf{e}_0$  about the symmetry axis of the quadric.*
5. *For any  $\mathbf{E}$  satisfying  $\mathbf{M} = \mathbf{E} + \mathbf{E}^\top$ , the relation  $\mathbf{R}_{sym}\mathbf{E}\mathbf{R}_{sym} = \mathbf{E}^\top$  holds.*

**Proof. Part 1.** A point  $\mathbf{x}$  lies on the conic  $\mathbf{M}$  if and only if  $\mathbf{x}^\top\mathbf{M}\mathbf{x} = 0$ . The fact that the two principal points  $\mathbf{m}$  and  $\mathbf{n}$  lie on  $\mathbf{M}$  is easily verified. Similarly, if  $\mathbf{E}\mathbf{e} = \mathbf{0}$ , then  $\mathbf{e}^\top(\mathbf{E} + \mathbf{E}^\top)\mathbf{e} = 0$ , so  $\mathbf{e}$  lies on  $\mathbf{M}$ .

**Part 2.** We may without loss of generalization assume that  $\mathbf{R}$  is a rotation about the  $Z$ -axis. If for some  $\mathbf{m}$  and  $\mathbf{n}$  one has  $\mathbf{E} + \mathbf{E}^\top = \mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top - 2\mathbf{m}^\top\mathbf{n}\mathbf{I}$ , then  $\mathbf{E} + \mathbf{E}^\top - \text{tr}(\mathbf{E} + \mathbf{E}^\top)/2 = \mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top$ . However, the form of  $\mathbf{E} + \mathbf{E}^\top$  is given by (4), from which it follows that  $\mathbf{E} + \mathbf{E}^\top - \text{tr}(\mathbf{E} + \mathbf{E}^\top)/2$  has an upper left-hand  $2 \times 2$  block of zeros. From this it easily follows that either  $\mathbf{m}$  or  $\mathbf{n}$  is  $(0, 0, 1)^\top$  and hence equal to the the rotation axis of  $\mathbf{R}$ . The axis of the other rotation matrix  $\mathbf{R}'$  is distinct from that of  $\mathbf{R}$ , and by the same argument must therefore be the other of the two principal points.

**Part 3.** Let  $\mathbf{M}$  be in standard position, with principal points given by  $(0, 0, 1)^\top$  and  $(2p, 0, 1)^\top$ . In this case,  $\mathbf{M}$  has the form given by (5) with  $q = 1$ . In any decomposition of  $\mathbf{M}$ , the rotation axis is one of these two principal points, and so we may assume  $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times$  where  $\mathbf{R}$  is a rotation about the  $Z$  axis. Let the rotation angle be  $\theta$  and  $\mathbf{t} = (x, y, z)^\top$ , then  $\mathbf{E}$  is given by (4). Equating (5) with (4) the unique solution is  $\mathbf{t} = (p(c+1), -ps, 1)^\top$ , provided that  $s \neq 0$ .

Thus, with  $\mathbf{M}$  given,  $\mathbf{E}$  is uniquely determined by its rotation angle,

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_{\times} = \begin{bmatrix} -s & -c & ps \\ c & -s & -p(c+1) \\ ps & p(c+1) & 0 \end{bmatrix}. \quad (8)$$

Now, as a conic (5), with  $q = 1$ ,  $\mathbf{M}$  may be written as  $X^2 + Y^2 - 2pXZ = 0$ . Dehomogenizing by setting  $Z = 1$  and completing the square, this becomes  $(X - p)^2 + Y^2 = p^2$ , which is a circle centred at  $(p, 0)$  of radius  $p$ . As  $c = \cos(\theta)$  and  $s = \sin(\theta)$  vary with  $\theta$ , the epipole  $\mathbf{t} = (p(c+1), -ps, 1)^{\top}$  traces out this circle. Thus, the points  $\mathbf{x}$  on  $\mathbf{M}$  are in one-to-one correspondence with the rotation angle, and so  $\mathbf{E}$  is uniquely determined by  $\mathbf{x}$ .

**Part 4.** Simply observe that the left epipole of  $\mathbf{E}$  in (8) is  $(-p(c+1), -ps, -1)^{\top}$  which is the rotation of the right epipole about the  $Y$ -axis.

**Part 5.** The identity  $\mathbf{R}_{sym}\mathbf{E}\mathbf{R}_{sym} = \mathbf{E}^{\top}$  is verified by direct computation using (8) and  $\mathbf{R}_{sym} = \text{diag}(-1, 1, -1)$  for the symmetry of a quadric in standard position. ■

## 6 Characterization of Critical Surfaces

We are now ready to determine the critical surfaces for calibrated reconstruction.

**Theorem 3.** *If  $(S, \mathbf{P}^0, \mathbf{P}^1)$  is a critical configuration, then  $S$  is a ruled rectangular quadric. Further, the camera centres  $\mathbf{t}_0$  and  $\mathbf{t}_1$  of  $\mathbf{P}^0$  and  $\mathbf{P}^1$  satisfy the condition that  $\mathbf{R}_{sym}\mathbf{t}_0$  and  $\mathbf{t}_1$  lie on a common generator of the quadric.*

*Conversely, if  $\mathbf{R}_{sym}\mathbf{t}_0$  and  $\mathbf{t}_1$  lie on a common generator, then the configuration is critical, provided that the generator does not pass through a principal point of the quadric.*

That the quadric is rectangular follows directly from (3) and the definition of a rectangular quadric in the case where the two cameras are normalized. For general cameras of the form  $\mathbf{R}_0[\mathbf{I}|\mathbf{t}_0]$  and  $\mathbf{R}_1[\mathbf{I}|\mathbf{t}_1]$ , the upper left-hand block of (2) is of the form  $\mathbf{M} = \mathbf{R}_0^{\top}\mathbf{E}\mathbf{R}_1 + \mathbf{R}_1^{\top}\mathbf{E}^{\top}\mathbf{R}_0$ . However, if  $\mathbf{E}$  is an essential matrix, then so is  $\mathbf{R}_0^{\top}\mathbf{E}\mathbf{R}_1$ , and so  $S$  is a rectangular quadric.

The necessary condition on the camera centres is completed by the following lemma, which specifies the relationship between the camera centres and the essential matrix more precisely.

**Lemma 1.** *If  $S$  is a critical quadric for normalized camera matrices with centres at  $\mathbf{t}_0$  and  $\mathbf{t}_1$ , and  $\mathbf{E}^{01}$  is the essential matrix satisfying (3), then  $\mathbf{R}_{sym}\mathbf{t}_0$  and  $\mathbf{t}_1$  lie on a generator of  $S$  with direction vector given by  $\mathbf{e}_1$ , the right epipole of  $\mathbf{E}^{01}$ .*

*Proof.* We may assume that  $S$  is in standard position. From this and (3) it follows that  $\mathbf{E}^{10}\mathbf{t}_0 + \mathbf{E}^{01}\mathbf{t}_1 = 0$ . Multiplying on the left by  $\mathbf{e}_1$  gives  $\mathbf{e}_1^{\top}\mathbf{E}^{01}\mathbf{t}_1 = 0$ .

Two things need to be proved: (i) For all  $\alpha$ , point  $\mathbf{t}_1 + \alpha\mathbf{e}_1$  lies on  $S$ , and (ii)  $\mathbf{R}_{sym}\mathbf{t}_0 = \mathbf{t}_1 + \alpha\mathbf{e}_1$ . Since the quadric has the diagonal block form  $S = \text{diag}(\mathbf{E}^{01} + \mathbf{E}^{10}, d)$ , the first point is proved by showing that

$$\mathbf{t}_1^{\top}\mathbf{M}\mathbf{t}_1 + 2\alpha\mathbf{e}_1^{\top}\mathbf{M}\mathbf{t}_1 + \alpha^2\mathbf{e}_1^{\top}\mathbf{M}\mathbf{e}_1 + d = 0.$$

However,  $\mathbf{e}_1^\top \mathbf{M} \mathbf{e}_1 = 0$ , because  $\mathbf{e}_1$  is on  $\mathbf{M}$  and  $\mathbf{t}_1^\top \mathbf{M} \mathbf{t}_1 + d = 0$ , because  $\mathbf{t}_1$  is on  $S$ . The remaining term reduces to  $2\alpha \mathbf{e}_1 \mathbf{E}^{01} \mathbf{t}_1$ , which is zero, as was just shown.

Now to the second point. Showing that  $\mathbf{R}_{sym} \mathbf{t}_0 = \mathbf{t}_1 + \alpha \mathbf{e}_1$  is equivalent to showing that  $\mathbf{E}^{01}(\mathbf{R}_{sym} \mathbf{t}_0 - \mathbf{t}_1) = 0$ . However,

$$\begin{aligned} \mathbf{E}^{01}(\mathbf{R}_{sym} \mathbf{t}_0 - \mathbf{t}_1) &= \mathbf{E}^{01} \mathbf{R}_{sym} \mathbf{t}_0 + \mathbf{E}^{10} \mathbf{t}_0 \quad \text{since} \quad \mathbf{E}^{10} \mathbf{t}_0 + \mathbf{E}^{01} \mathbf{t}_1 = 0 \\ &= (\mathbf{R}_{sym} + \mathbf{I}) \mathbf{E}^{10} \mathbf{t}_0 \quad \text{since} \quad \mathbf{E}^{10} \mathbf{R}_{sym} = \mathbf{R}_{sym} \mathbf{E}^{01}. \end{aligned}$$

It has been shown that  $\mathbf{e}_1^\top \mathbf{E}^{01} \mathbf{t}_1 = 0$ , and trivially  $\mathbf{e}_0^\top \mathbf{E}^{01} \mathbf{t}_1 = 0$ . Thus  $\mathbf{E}^{01} \mathbf{t}_1$  is perpendicular to both  $\mathbf{e}_0$  and  $\mathbf{e}_1$ . Consequently  $\mathbf{E}^{01} \mathbf{t}_1 = k(\mathbf{e}_0 \times \mathbf{e}_1)$ . In turn,

$$\mathbf{R}_{sym}(\mathbf{e}_0 \times \mathbf{e}_1) = (\mathbf{R}_{sym} \mathbf{e}_0) \times (\mathbf{R}_{sym} \mathbf{e}_1) = \mathbf{e}_1 \times \mathbf{e}_0 = -(\mathbf{e}_0 \times \mathbf{e}_1).$$

So  $(\mathbf{R}_{sym} + \mathbf{I}) \mathbf{E}^{01} \mathbf{t}_1 = k(\mathbf{R}_{sym} + \mathbf{I})(\mathbf{e}_0 \times \mathbf{e}_1) = 0$ , as required.  $\blacksquare$

**Converse.** Suppose that  $S$  is in standard position, and that  $\mathbf{t}_1$  and  $\mathbf{R}_{sym} \mathbf{t}_0$  lie on a common generator, whose direction vector we denote by  $\mathbf{e}_1$ . According to hypothesis,  $\mathbf{e}_1$  is not coincident with one of the principal points of the quadric. In this case, according to Theorem 2, there exists an essential matrix  $\mathbf{E}^{10}$  such that  $\mathbf{E}^{01} \mathbf{e}_1 = \mathbf{0}$  and  $\mathbf{M} = \mathbf{E}^{01} + \mathbf{E}^{10}$ . According to Theorem 2 again,  $\mathbf{e}_0 = \mathbf{R}_{sym} \mathbf{e}_1$  is the other epipole of  $\mathbf{E}^{10}$ , satisfying  $\mathbf{e}_0^\top \mathbf{E}^{01} = 0$ .

Our goal is to demonstrate that (3) holds for this choice of  $\mathbf{E}^{10}$ . Since  $S$  is assumed to be in standard position,  $S = \text{diag}(\mathbf{M}, d)$ , it suffices to prove that  $\mathbf{E}^{10} \mathbf{t}_0 + \mathbf{E}^{01} \mathbf{t}_1 = \mathbf{0}$  and  $2\mathbf{t}_0^\top \mathbf{E}^{01} \mathbf{t}_1 = d$ . Let  $\mathbf{w} = \mathbf{E}^{10} \mathbf{t}_0 + \mathbf{E}^{01} \mathbf{t}_1$ . Then

$$\mathbf{R}_{sym} \mathbf{w} = \mathbf{R}_{sym} \mathbf{E}^{10} \mathbf{t}_0 + \mathbf{R}_{sym} \mathbf{E}^{01} \mathbf{t}_1 = \mathbf{E}^{01} \mathbf{R}_{sym} \mathbf{t}_0 + \mathbf{E}^{10} \mathbf{R}_{sym} \mathbf{t}_1 = \mathbf{E}^{01} \mathbf{t}_1 + \mathbf{E}^{10} \mathbf{t}_0 = \mathbf{w}.$$

On the other hand, by assumption  $\mathbf{t}_1$  lies on a generator with direction vector  $\mathbf{e}_1$ . Mimicking part of the proof of Lemma 1 leads to the conclusion that  $\mathbf{e}_1^\top \mathbf{E}^{01} \mathbf{t}_1 = 0$ , and hence  $\mathbf{e}_1^\top \mathbf{w} = 0$ . Similarly  $\mathbf{e}_0^\top \mathbf{w} = 0$ . Consequently, up to scale,  $\mathbf{w} = \mathbf{e}_1 \times \mathbf{e}_0$ . As in the proof of Lemma 1, it follows that  $\mathbf{R}_{sym} \mathbf{w} = -\mathbf{w}$ , and so  $\mathbf{w} = \mathbf{E}^{10} \mathbf{t}_0 + \mathbf{E}^{01} \mathbf{t}_1 = \mathbf{0}$  as required.

Finally, since  $\mathbf{t}_0$  lies on  $S$ , it follows that  $d = -\mathbf{t}_0^\top (\mathbf{E}^{01} + \mathbf{E}^{10}) \mathbf{t}_0$ . Using  $\mathbf{E}^{10} \mathbf{t}_0 + \mathbf{E}^{01} \mathbf{t}_1 = 0$ , it follows that  $d = 2\mathbf{t}_0^\top \mathbf{E}^{01} \mathbf{t}_1$ , and the proof is complete.  $\blacksquare$

**Number of conjugate configurations.** This theorem gives us insight into how many conjugate configurations (that is, different essential matrices  $\mathbf{E}_q$ ) exist for a given critical configuration. It was shown that the essential matrix  $\mathbf{E} = \mathbf{E}_q$  is uniquely determined by the quadric  $S$  and the vanishing point of the generator containing  $\mathbf{t}_1$  and  $\mathbf{R}_{sym} \mathbf{t}_0$ . The only possibility for there to exist two distinct essential matrices  $\mathbf{E}_q$  is if  $\mathbf{t}_1 = \mathbf{R}_{sym} \mathbf{t}_0$ , in which case each of the generators through  $\mathbf{t}_1$  leads to a different essential matrix  $\mathbf{E}_q$ .

## 7 A Condition for Ambiguity in 3 Views and More

Since Euclidean ambiguities are special cases of projective ambiguities, it is useful to have a (nearly) necessary and sufficient condition for ambiguity. Such a condition is given by the following theorem, which is a restatement of Theorem 1 and Corollary 1 of [4], in slightly simpler form.



**Theorem 4.** Let  $(\mathbf{P}^0, \mathbf{P}^1, \mathbf{P}^2)$  and  $(\mathbf{Q}^0, \mathbf{Q}^1, \mathbf{Q}^2)$  be two triplets of camera matrices. For each of the pairs  $(i, j) = (0, 1), (0, 2)$  and  $(1, 2)$ , let  $S_{\mathbf{P}}^{ij}$  and  $S_{\mathbf{Q}}^{ij}$  be the ruled quadric critical surfaces defined in (1) for camera pairs  $(\mathbf{P}^i, \mathbf{P}^j)$  and  $(\mathbf{Q}^i, \mathbf{Q}^j)$ , respectively.

- (i) If there exist points  $\mathbf{P}$  and  $\mathbf{Q}$  such that  $\mathbf{P}^i \mathbf{P} = \mathbf{Q}^i \mathbf{Q}$  for all  $i = 0, 1, 2$  then  $\mathbf{P}$  must lie on the intersection  $S_{\mathbf{P}}^{01} \cap S_{\mathbf{P}}^{02} \cap S_{\mathbf{P}}^{12}$  and  $\mathbf{Q}$  must lie on  $S_{\mathbf{Q}}^{01} \cap S_{\mathbf{Q}}^{02} \cap S_{\mathbf{Q}}^{12}$ .
- (ii) Conversely, if  $\mathbf{P}$  is a point lying on the intersection of quadrics  $S_{\mathbf{P}}^{01} \cap S_{\mathbf{P}}^{02} \cap S_{\mathbf{P}}^{12}$ , but not satisfying the condition

$$\begin{bmatrix} (e_{\mathbf{Q}}^{10} \times e_{\mathbf{Q}}^{20})^\top \mathbf{P}^0 \\ (e_{\mathbf{Q}}^{21} \times e_{\mathbf{Q}}^{01})^\top \mathbf{P}^1 \\ (e_{\mathbf{Q}}^{02} \times e_{\mathbf{Q}}^{12})^\top \mathbf{P}^2 \end{bmatrix} \mathbf{P} = 0, \quad (9)$$

where each  $e_{\mathbf{Q}}^{ij}$  is an epipole (the image of the camera centre of  $\mathbf{Q}^i$  in the image formed by  $\mathbf{Q}^j$ ), then there exists a point  $\mathbf{Q}$  lying on  $S_{\mathbf{Q}}^{01} \cap S_{\mathbf{Q}}^{02} \cap S_{\mathbf{Q}}^{12}$  such that  $\mathbf{P}^i \mathbf{P} = \mathbf{Q}^i \mathbf{Q}$  for all  $i = 0, 1, 2$ .

If a point  $\mathbf{P}$  happens to satisfy the condition (9) then there may or may not be a conjugate point  $\mathbf{Q}$ . In a reasonable sense, most points lying on the intersection  $S_{\mathbf{P}}^{01} \cap S_{\mathbf{P}}^{02} \cap S_{\mathbf{P}}^{12}$  are critical. Notice, however that if the three cameras  $\mathbf{Q}^i$  are collinear, then each of the vector products  $(e_{\mathbf{Q}}^{ij} \times e_{\mathbf{Q}}^{kj})$  vanishes, and so condition (9) is satisfied for all  $\mathbf{P}$ . In this case we can make no conclusion regarding the existence of a conjugate point  $\mathbf{Q}$ . However, if the three cameras  $\mathbf{Q}$  are not collinear then we may say more.

**Proposition 2.** Given the assumptions of Theorem 4, suppose further that the three cameras  $\mathbf{Q}^i$  are distinct and non-collinear. Then any point satisfying the condition (9) must lie on the intersection of quadrics  $S_{\mathbf{P}}^{01} \cap S_{\mathbf{P}}^{02} \cap S_{\mathbf{P}}^{12}$ .

*Proof.* Let  $i, j$  and  $k$  represent the three indices 0, 1 and 2 in some permuted order, i.e.  $i \neq j \neq k$ . If the three cameras are non-collinear, then for each  $j$  the cross product  $e_{\mathbf{Q}}^{ij} \times e_{\mathbf{Q}}^{kj}$  is non-vanishing. Let  $\mathbf{P}$  be a point satisfying (9). Then  $(e_{\mathbf{Q}}^{ij} \times e_{\mathbf{Q}}^{kj})^\top (\mathbf{P}^j \mathbf{P}) = 0$ , which implies that  $\mathbf{P}^j \mathbf{P}$  lies in the span of  $e_{\mathbf{Q}}^{ij}$  and  $e_{\mathbf{Q}}^{kj}$ , and so we write  $\mathbf{P}^j \mathbf{P} = \alpha_{ij} e_{\mathbf{Q}}^{ij} + \alpha_{kj} e_{\mathbf{Q}}^{kj}$  for some constants  $\alpha_{ij}$  and  $\alpha_{kj}$ . Now,  $\mathbf{P}$  lies on  $S_{\mathbf{P}}^{ij}$  if and only if  $\mathbf{P}^\top (\mathbf{P}^i \mathbf{P}^j \mathbf{F}_{\mathbf{Q}}^{ij}) \mathbf{P} = 0$ . Substituting for  $\mathbf{P}^i \mathbf{P}$  and  $\mathbf{P}^j \mathbf{P}$  gives

$$\mathbf{P}^\top (\mathbf{P}^i \mathbf{P}^j \mathbf{F}_{\mathbf{Q}}^{ij}) \mathbf{P} = (\alpha_{ji} e_{\mathbf{Q}}^{ji} + \alpha_{ki} e_{\mathbf{Q}}^{ki})^\top \mathbf{F}_{\mathbf{Q}}^{ij} (\alpha_{ij} e_{\mathbf{Q}}^{ij} + \alpha_{kj} e_{\mathbf{Q}}^{kj}) = (\alpha_{ki} e_{\mathbf{Q}}^{ki})^\top \mathbf{F}_{\mathbf{Q}}^{ij} (\alpha_{kj} e_{\mathbf{Q}}^{kj}).$$

The last equality holds, because  $e_{\mathbf{Q}}^{ji \top} \mathbf{F}_{\mathbf{Q}}^{ij} = \mathbf{F}_{\mathbf{Q}}^{ij} e_{\mathbf{Q}}^{ij} = 0$ . Finally,  $e_{\mathbf{Q}}^{ki \top} \mathbf{F}_{\mathbf{Q}}^{ij} e_{\mathbf{Q}}^{kj} = 0$ , since  $e_{\mathbf{Q}}^{ki}$  and  $e_{\mathbf{Q}}^{kj}$  are a matching point pair in images  $i$  and  $j$ , corresponding to the camera centre of  $\mathbf{Q}^k$ . Thus,  $\mathbf{P}^\top (\mathbf{P}^i \mathbf{P}^j \mathbf{F}_{\mathbf{Q}}^{ij}) \mathbf{P} = 0$  and so  $\mathbf{P}$  lies on  $S_{\mathbf{P}}^{ij}$ . ■

The points  $\mathbf{P}$  that satisfy (9) must be either a single point, a line or a plane lying in the intersection of the three quadrics  $S_{\mathbf{P}}^{ij}$ . If this quadric intersection does not contain a complete line or a plane, then the latter two cases are not possible. In addition, it may be shown by continuity that if (9) defines a single

point, then this point must be critical (a conjugate  $\mathbf{Q}$  exists) unless it is an isolated single point in the intersection of the  $S_p^{ij}$ . We may therefore state a general ambiguity result:

**Theorem 5.** *Let  $(\mathbf{P}^0, \mathbf{P}^1, \mathbf{P}^2)$  and  $(\mathbf{Q}^0, \mathbf{Q}^1, \mathbf{Q}^2)$  be two triplets of camera matrices, with cameras  $\mathbf{Q}^i$  non-collinear. Then for any point  $\mathbf{P}$  in the intersection  $S_p^{01} \cap S_p^{02} \cap S_p^{12}$  there exists a conjugate point  $\mathbf{Q}$  satisfying  $\mathbf{P}^i \mathbf{P} = \mathbf{Q}^i \mathbf{Q}$  for all  $i$ , with the possible exception of*

1. *A single isolated point  $\mathbf{P}$  in  $S_p^{01} \cap S_p^{02} \cap S_p^{12}$ , or*
2. *Points  $\mathbf{P}$  on a single line or plane contained in the intersection  $S_p^{01} \cap S_p^{02} \cap S_p^{12}$ .*

This theorem simplifies the search for critical configurations, since it is not necessary to worry about the points. It is sufficient to find sets of cameras that define quadric intersections of interest. If we are searching for critical calibrated configurations, then the two sets of cameras must of course be calibrated.

The question arises as to whether the exceptional conditions of Theorem 5 really occur (the points that are non-critical). It was shown in [9] that if the three quadrics intersect in 8 points then indeed one of these points (the exceptional point identified in Theorem 5) does not have a conjugate. For the case where the three quadrics intersect in a line, an example is given later in which the points on the line in fact do not have conjugates.

## 8 Euclidean Ambiguities in 3 Views or More

As seen in the previous section, calibrated critical configurations involving three views and seven points abound. It is natural to ask if calibrated critical configurations exist involving more than two views and infinite numbers of points. In the projective case, it has been shown that elliptic quartics (a fourth-degree curve given as the intersection of two quadrics) are critical for projective reconstruction [6]. The calibration information restricts the class of critical sets to a class which is strictly smaller than in the projective case. Still, we will show that for any three cameras, there exists an elliptic quartic through the three camera centres such that the points on the quartic and the three cameras form a critical configuration. First some properties of pencils of rectangular quadrics are given.

**Lemma 2.** *Let  $S_1$  and  $S_2$  be two rectangular quadrics with principal points  $(\mathbf{m}_1, \mathbf{n}_1)$  and  $(\mathbf{m}_2, \mathbf{n}_2)$ , respectively.*

- (i) *There exists in general a third rectangular quadric in the pencil  $\alpha S_1 + \beta S_2$ .*
- (ii) *All the quadrics in the pencil  $\alpha S_1 + \beta S_2$  are rectangular if and only if (a) one of the principal points  $(\mathbf{m}_1, \mathbf{n}_1)$  coincides with one of  $(\mathbf{m}_2, \mathbf{n}_2)$ , or (b) all four principal points are collinear.*

*Proof.* From Proposition 1 it follows that  $\text{tr}(M)/2$  is an eigenvalue of  $M$ . Thus, a necessary constraint for a rectangular quadric is that  $\det[M - \frac{\text{tr}(M)}{2}I] = 0$ , which is also sufficient (provided the product of the two other eigenvalues is positive

- otherwise the principal points will be complex). Applying the constraint to  $\alpha S_1 + \beta S_2$  yields

$$\alpha^2 \beta \det [\mathbf{n}_1 \ \mathbf{n}_2 \ \mathbf{m}_1] \det [\mathbf{m}_1 \ \mathbf{m}_2 \ \mathbf{n}_1] + \alpha \beta^2 \det [\mathbf{n}_1 \ \mathbf{n}_2 \ \mathbf{m}_2] \det [\mathbf{m}_1 \ \mathbf{m}_2 \ \mathbf{n}_2] = 0.$$

This is a homogeneous polynomial constraint in  $(\alpha, \beta)$  where two solutions are  $(1, 0)$  and  $(0, 1)$ . Since it is a cubic constraint, there is always a third solution which proves (i). All quadrics in the pencil are rectangular if and only if the two coefficients of the polynomial vanish. It follows that (1) either  $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1)$  or  $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{n}_1)$  are collinear and (2) either  $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_2)$  or  $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{n}_2)$  are collinear, which occurs exactly in the two cases given by (ii) above. ■

Based on the observations in Theorem 3 and Lemma 2 we are now ready to prove the following result on critical configurations for calibrated cameras.

**Theorem 6.** *Given three calibrated cameras  $(\mathbf{P}^0, \mathbf{P}^1, \mathbf{P}^2)$ , then there exists an elliptic quartic curve (given as the intersection of two quadrics) which contains the three camera centres and such that the points lying on the quartic curve and the three cameras constitute a critical configuration.*

*Proof.* According to Theorem 4, we need to find a triplet of conjugate cameras  $(\mathbf{Q}^0, \mathbf{Q}^1, \mathbf{Q}^2)$  which are calibrated and where the corresponding three quadrics  $S_{\mathbf{P}^0}^{01}, S_{\mathbf{P}^0}^{02}$  and  $S_{\mathbf{P}^0}^{12}$  are linearly dependent. Without loss of generality we can assume normalized cameras and that the two camera centres of  $\mathbf{P}^0$  and  $\mathbf{P}^1$  are given by  $\mathbf{t}_0 = (0, 0, 0)^\top$  and  $\mathbf{t}_1 = (0, 1, 0)^\top$ , respectively.

An explicit solution to the problem will given, but first we will describe how the solution was discovered. Start with three general camera matrices  $(\mathbf{Q}^0, \mathbf{Q}^1, \mathbf{Q}^2)$  with  $\mathbf{Q}^0 = [I | 0]$  and  $\mathbf{Q}^i = \mathbf{R}_q^i [I | -\mathbf{t}_{q,i}]$  for  $i = 1, 2$ . According to Theorem 2, the rotation axes of  $\mathbf{R}_q^1, \mathbf{R}_q^2$  and  $\mathbf{R}_q^1 \mathbf{R}_q^2$  coincide with one of the two principal points in the quadrics  $S_{\mathbf{P}^0}^{01}, S_{\mathbf{P}^0}^{02}$  and  $S_{\mathbf{P}^0}^{12}$ , respectively. At the same time, the pencil should be rectangular. By choosing a fix rotation axis, denoted by  $\mathbf{m}$ , for  $\mathbf{R}_q^1$  and  $\mathbf{R}_q^2$ , implies that the rotation axis of  $\mathbf{R}_q^1 \mathbf{R}_q^2$  will also be  $\mathbf{m}$ . Furthermore, one of the principal points for  $S_{\mathbf{P}^0}^{01}, S_{\mathbf{P}^0}^{02}$  and  $S_{\mathbf{P}^0}^{12}$  will be  $\mathbf{m}$  and hence the pencil spanned by  $S_{\mathbf{P}^0}^{01}$  and  $S_{\mathbf{P}^0}^{02}$  is rectangular according to Lemma 2. So ensuring these constraints is sufficient in order to generate a pencil of rectangular quadrics.

Now let  $\mathbf{m} = (1, 0, 0)^\top$ . Denote the camera centre coordinates of  $\mathbf{P}^2$  with  $\mathbf{t}_2 = (x, y, z)^\top$  and let  $\mu = \sqrt{2(y - 1/2)^2 + 2(z + 1/2)^2}$  and  $\nu = y^2 - y + z^2$ . Straightforward calculations show that

$$\mathbf{Q}^1 = \begin{bmatrix} 1 & 0 & 0 & x(-2y+1) \\ 0 & 0 & 1 & -\nu \\ 0 & -1 & 0 & \nu \end{bmatrix}, \quad \mathbf{Q}^2 = \begin{bmatrix} \mu & 0 & 0 & \mu x(z-y-\mu+1) \\ 0 & z-y+1 & y+z & -(y+z)\nu \\ 0 & -z-y & z-y+1 & \nu(y-z+\mu-1) \end{bmatrix} \quad (10)$$

generates a pencil spanned by

$$S_{\mathbf{P}^0}^{01} = \begin{bmatrix} 0 & -2\nu & 0 & \nu \\ -2\nu & 2x(2y-1) & 0 & x(-2y+1) \\ 0 & 0 & 2x(2y-1) & 0 \\ \nu & x(-2y+1) & 0 & 0 \end{bmatrix} \quad \text{and}$$

$$S_P^{02} = \begin{bmatrix} 0 & -2\nu\xi & 0 & \nu(\nu + y\mu) \\ -2\nu\xi & 2x(y + z)\xi & 0 & -x(y + z)\xi \\ 0 & 0 & 2x(y + z)\xi & x\nu(-2y - \mu + 1) \\ \nu(\nu + y\mu) - x(y + z)\xi & x\nu(-2y - \mu + 1) & 0 & 0 \end{bmatrix},$$

where  $\xi = y - z + \mu - 1$  and  $S_P^{12} = \alpha S_P^{01} + \beta S_P^{02}$  for some  $(\alpha, \beta) \in \mathcal{P}^1$ . The pencil contains the three cameras centres  $\mathbf{t}_0$ ,  $\mathbf{t}_1$  and  $\mathbf{t}_2$  and the whole intersection curve of the pencil (which is an elliptic quartic) is critical as the exception condition (9) contains in general only a single point, cf. Proposition 2 and Theorem 5. The above solution breaks down, when for example  $\mu = 0$ . By interchanging the roles of two cameras, say  $P^0$  and  $P^1$ , will then generally produce a valid solution. ■

As the proof is constructive, it is easy to generate examples.

*Example 1.* Let  $(P^0, P^1, P^2)$  be three normalized cameras with centres  $\mathbf{t}_0 = (0, 0, 0)^\top$ ,  $\mathbf{t}_1 = (0, 1, 0)^\top$  and  $\mathbf{t}_2 = (1, 1, 3)^\top$  lying on a pencil spanned by

$$S_1 = \begin{bmatrix} 0 & 18 & 0 & -9 \\ 18 & -2 & 0 & 1 \\ 0 & 0 & -2 & 0 \\ -9 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} 0 & 0 & 0 & 18 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 2 & -9 \\ 18 & -1 & -9 & 0 \end{bmatrix}. \quad (11)$$

Let  $g(t)$  be the solution to the quadratic equation  $(2t + 1)X^2 + (-18t + 9)X - t - t^2 + 2t^3 = 0$ , then the intersection curve can be written in homogeneous form  $\mathbf{P}(t) = (g(t), t(2t + 1), g(t)(2t + 1), 2t + 1)^\top$ . Further, according to (10), the conjugate cameras are

$$Q^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad Q^1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -9 \\ 0 & -1 & 0 & 9 \end{bmatrix} \quad \text{and} \quad Q^2 = \begin{bmatrix} 5 & 0 & 0 & -10 \\ 0 & 3 & 4 & -36 \\ 0 & -4 & 3 & 18 \end{bmatrix}.$$

The corresponding quadrics  $S_P^{01}$ ,  $S_P^{02}$  and  $S_P^{12}$  lie in the pencil  $\alpha S_1 + \beta S_2$ . The elliptic quartic given by quadrics  $S_Q^{01}$  and  $S_Q^{02}$  can be parametrized by  $\mathbf{Q}(t) = (2(t + 5)(t - 1), 2(t + 5)(18t - 9 - (2t + 1)g(t)), 2(t + 5)(t - 1)(2t - 1), 2t^2 + 17t - 10 - (2t + 1)g(t))^\top$ . Finally, one verifies that  $P^i P(t) = Q^i Q(t)$  (up to scale) for  $i = 0, 1, 2$  and all  $t$ . Thus, the configuration is indeed critical.

Next, we wish to study critical configurations of  $n > 3$  calibrated views.

**Theorem 7.** *A configuration of  $n \geq 3$  calibrated cameras  $P^i$ ,  $i = 0, \dots, n - 1$  and points  $\mathbf{P}_j$  is critical if the set of cameras  $(P^0, P^1, P^k)$  and points  $\mathbf{P}_j$  is critical with respect to some conjugate cameras  $(Q^0, Q^1, Q^k)$  for  $k = 2, \dots, n - 1$ .*

*Proof.* We prove the result for 4 views. The general result for  $n$  views follows by induction. The three cameras  $P^0, P^1$  and  $P^2$  along with the points form a critical set, and hence a conjugate configuration exists. Similarly a second conjugate configuration exists for the cameras  $P^0, P^1$  and  $P^3$  and the points. The goal is to show that these two conjugate configurations are consistent.

By assumption, the conjugate pair  $(Q^0, Q^1)$  is the same for both triplets  $(P^0, P^1, P^2)$  and  $(P^0, P^1, P^3)$ . Denote the conjugate points by  $Q_j$  and  $Q'_j$  in the first and second triplet, respectively. Consider the way the conjugate points  $Q_j$  are obtained in the first triplet. From the image points in the first two views, one can determine the position of the conjugate points by triangulation using  $Q^0$  and  $Q^1$ . However, and this is the main point, the third camera is not used in this construction. It follows that  $Q_j = Q'_j$  and the theorem is proved. ■

*Example 2.* Consider again the pencil spanned by  $S_1$  and  $S_2$  in (11). Are there any additional camera positions in Example 1 for which the configuration remains critical? Yes, the following camera pair does not break the ambiguity:

$$P^3 = \begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 9/5 \\ 0 & 0 & 1 & -78/5 \end{bmatrix} \quad \text{and} \quad Q^3 = \begin{bmatrix} 5 & 0 & 0 & -45 \\ 0 & -4 & -3 & 27 \\ 0 & 3 & -4 & 81 \end{bmatrix},$$

How were these cameras discovered? Well, if a camera  $P^3$  is to be critical, Theorem 7 says that we only need to show that  $(P^0, P^1, P^3)$  and the points on the quartic curve are critical. The way to do that is by means of Theorem 4. Thus, the constraints that have to be satisfied are (i) the camera centre of  $P^3$  lies on both  $S_1$  and  $S_2$  and (ii) the quadrics  $S_P^{ij}$  in (1) for pairs  $(P^0, P^3)$  and  $(P^1, P^3)$ , respectively, lie in the pencil  $\alpha S_1 + \beta S_2$ . Again, without loss of generality, one can assume that  $P^3$  is normalized. The only valid solution to this system of polynomial equations is the one given above.

In the uncalibrated case,  $n$  cameras with centres and points lying on an elliptic quartic are critical [6]. The previous example shows that this is not true in the calibrated case. One might suspect that there are only critical configurations with a finite number of cameras.

*Example 3.* Consider the pencil  $\alpha S_1 + \beta S_2$  where

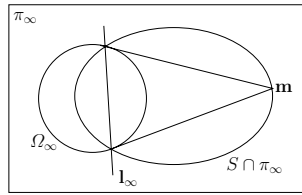
$$S_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & -1 \\ 1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix}.$$

The intersection curve splits up into a line and a twisted cubic, where the line is the  $X$ -axis and the points on the twisted cubic can be parametrized by  $P(\theta) = (2\theta(2\theta^2 - 2\theta + 1), \theta^2(2\theta - 1), (2\theta - 1)(-\theta + 1), 2\theta^2 - 2\theta + 1)^\top$ . Let

$$P^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P^1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad P^2 = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

A conjugate configuration is given by

$$Q^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, Q^1 = \begin{bmatrix} 2\sqrt{2} & 0 & 0 & 2(\sqrt{2} - 1) \\ 0 & 2 & 2 & -1 \\ 0 & -2 & 2 & -1 \end{bmatrix} \quad \text{and} \quad Q^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$



**Fig. 1.** Intersection of a rectangular quadric  $S$  with the plane at infinity  $\pi_\infty$  with principal point  $\mathbf{m}$ .

as the corresponding quadrics  $S_p^{01}$ ,  $S_p^{02}$  and  $S_p^{12}$  lie in the pencil  $\alpha S_1 + \beta S_2$ . However, the whole intersection curve is not critical. The exception condition in (9) consists of the  $X$ -axis and by inspection, one finds that there are no conjugate points for the  $X$ -axis. Thus, the points on the twisted cubic and the cameras  $(P^0, P^1, P^2)$  form a critical configuration. This can also be verified by direct computations: The conjugate points to the twisted cubic is a conic curve, which can be parametrized by  $\mathbf{Q}(\theta) = (-4\theta^2 + 4\theta - 2, \theta(-2\theta + 1), (2\theta - 1)(\theta - 1), 4\theta^2 - 4\theta + 2)^\top$  and  $\mathbf{P}^i \mathbf{P}(\theta) = \mathbf{Q}^i \mathbf{Q}(\theta)$  for  $i = 0, 1, 2$  and all  $\theta$ .

Are there any additional camera positions for which the configuration remains critical? Yes, for any camera  $\mathbf{P}$  lying on the twisted cubic there is a conjugate calibrated camera  $\mathbf{Q}$ ,

$$\mathbf{P}(\eta) = \begin{bmatrix} 1 & 0 & 0 & -2\eta \\ 0 & 1 & 0 & \frac{\eta^2(-2\eta+1)}{2\eta^2-2\eta+1} \\ 0 & 0 & 1 & \frac{\eta(2\eta-1)(\eta-1)}{2\eta^2-2\eta+1} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}(\eta) = \begin{bmatrix} \xi & 0 & 0 & \xi(-\xi+1) \\ 0 & -\eta+1 & \eta & -\eta^2 \\ 0 & -\eta & -\eta+1 & \eta(\eta-1) \end{bmatrix},$$

where  $\xi = \sqrt{2(\eta - 1/2)^2 + 1/2}$ . Notice that the camera centres in the conjugate configuration lie on a conic. In order to verify that the configuration is indeed critical, it is enough to check that the corresponding critical quadrics for pairs  $(P^0, P(\eta))$  and  $(P^1, P(\eta))$  lie in the pencil  $\alpha S_1 + \beta S_2$  or, alternatively, that  $\mathbf{P}(\eta) \mathbf{P}(\theta) = \mathbf{Q}(\eta) \mathbf{Q}(\theta)$  for all  $\eta, \theta$ .

## 9 Conclusions

In this paper we have given a complete characterization of critical surfaces for two calibrated cameras. We have shown several new results on critical configurations for multiple views. For example, for any placement of three calibrated cameras there exists a critical elliptic quartic curve. Further, the existence of critical configurations containing arbitrarily many points and cameras have been shown, even though they are less frequent than in the uncalibrated case.

## Appendix

*Geometric interpretation.* The definition of a rectangular quadric and its properties as stated in Proposition 1 are purely based on algebraic concepts. We will now give a more geometrically oriented characterization.

**Proposition 3.** *A principal point  $\mathbf{m}$  of a rectangular quadric is contained in the intersection of the quadric with the plane at infinity with the following property: The tangents from  $\mathbf{m}$  to the absolute conic meet the absolute conic at points lying on the quadric.*

In [8], this was used for defining a principal point. See Figure 1 for an illustration. This is still quite abstract. Before we give another interpretation, we need two simple facts about the absolute conic. For justification, refer to [10].

**Proposition 4.** *A planar conic is a circle if and only if it meets the plane at infinity at two (imaginary) points lying on the absolute conic.*

Now consider any line in space meeting the plane at infinity at a point  $\mathbf{m}$ . The polar of  $\mathbf{m}$  with respect to the absolute conic is the line  $\mathbf{l}_\infty$  joining the two points of tangency from  $\mathbf{m}$  to the absolute conic (see Figure 1). This line is the vanishing line of a plane perpendicular to the line first mentioned.

**Proposition 5.** *A plane and a line are perpendicular if and only if they meet the plane at infinity in a polar line-point pair with respect to the absolute conic.*

Now, refer to Figure 1. Let  $\pi$  be a plane that vanishes at the line  $\mathbf{l}_\infty$  on  $\pi_\infty$ . This plane meets the quadric  $S$  in a conic curve. At the plane at infinity  $\pi_\infty$ , the quadric  $S$ , the absolute conic  $\Omega_\infty$  and the plane  $\pi$  all meet. According to Proposition 4, this means that  $\pi$  and the quadric  $S$  meet in a circle.

The point  $\mathbf{m}$  is the polar of the line  $\mathbf{l}_\infty$  with respect to the absolute conic, and hence represents the vanishing direction perpendicular to the plane  $\pi$ . If  $\mathbf{m}$  is a principal point of the quadric  $S$ , then it lies on  $S$ .

**Proposition 6.** *A quadric  $S$  is rectangular if there exists a plane that meets the quadric in a circle and such that the perpendicular direction to the plane is asymptotic to the quadric.*

## References

1. T. Buchanan. Critical sets for 3d reconstruction using lines. In G. Sandini, editor, *European Conf. Computer Vision*, pages 730–738, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.
2. S. Carlsson. Duality of reconstruction and positioning from projective views. In *IEEE Workshop on Representation of Visual Scenes*, pages 85–92, Cambridge Ma, USA, 1995.
3. R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conf. Computer Vision*, pages 579–587, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.
4. R. Hartley. Ambiguous configurations for 3-view projective reconstruction. In *European Conf. Computer Vision*, volume I, pages 922–935, Dublin, Ireland, 2000.
5. R. Hartley and G. DeBunne. Dualizing scene reconstruction algorithms. In *3D Structure from Multiple Image of Large-Scale Environments, European Workshop, SMILE*, pages 14–31, Freiburg, Germany, 1998.
6. F. Kahl, R. Hartley, and K. Åström. Critical configurations for N-view projective reconstruction. In *Conf. Computer Vision and Pattern Recognition*, volume II, pages 158–163, Hawaii, USA, 2001.

7. J. Krames. Zur Ermittlung eines Objectes aus zwei Perspektiven (Ein Beitrag zur Theorie der gefährlichen Örter). *Monatsh. Math. Phys.*, 49:327–354, 1940.
8. S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, Berlin, Heidelberg, New York, 1993.
9. S. Maybank and A. Shashua. Ambiguity in reconstruction from images of six points. In *Int. Conf. Computer Vision*, pages 703–708, Mumbai, India, 1998.
10. J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Clarendon Press, Oxford, 1952.



# View Synthesis with Occlusion Reasoning Using Quasi-Sparse Feature Correspondences

David Jelinek and Camillo J. Taylor

GRASP Laboratory, CIS Department  
University of Pennsylvania  
3401 Walnut Street, Rm 335C  
Philadelphia, PA, 19104-6229  
{davidj2,cjtaylor}@grasp.cis.upenn.edu  
Phone: (215) 898 0376  
Fax: (215) 573 2048

**Abstract.** The goal of most image based rendering systems can be stated as follows: given a set of pictures taken from various vantage points, synthesize the image that would be obtained from a novel viewpoint. In this paper we present a novel approach to view synthesis which hinges on the observation that human viewers tend to be quite sensitive to the motion of features in the image corresponding to intensity discontinuities or edges. Our system focuses its efforts on recovering the 3D position of these features so that their motions can be synthesized correctly. In the current implementation these feature points are recovered from image sequences by employing the epipolar plane image (EPI) analysis techniques proposed by Bolles, Baker, and Marimont. The output of this procedure resembles the output of an edge extraction system where the edgels are augmented with accurate depth information. This method has the advantage of producing accurate depth estimates for most of the salient features in the scene including those corresponding to occluding contours. We will demonstrate that it is possible to produce compelling novel views based on this information.

The paper will also describe a principled approach to reasoning about the 3D structure of the scene based on the quasi-sparse features returned by the EPI analysis. This analysis allows us to correctly reproduce occlusion and disocclusion effects in the synthetic views without requiring dense correspondences. Importantly, the technique could also be used to analyze and refine the 3-D results returned by range finders, stereo systems or structure from motion algorithms. Results obtained by applying the proposed techniques to actual image data sets are presented.

**Keywords:** Structure From Motion, Surface Geometry, Image Based Rendering

# 1 Introduction

The goal of most image based rendering systems can be stated as follows: given a set of pictures taken from various vantage points, synthesize the image that would be obtained from a novel viewpoint. In this paper we present a novel approach to view synthesis which hinges on the observation that human viewers tend to be quite sensitive to the motion of features in the image corresponding to intensity discontinuities or edges. These discontinuities may arise from a number of sources including albedo changes, cast shadows and depth discontinuities in the scene. Our system focuses its efforts on recovering the 3D position of these features so that their motions can be synthesized correctly. In the current implementation these feature points are recovered from image sequences by employing the epipolar plane image (EPI) analysis techniques proposed by Bolles, Baker, and Marimont [1]. The output of this procedure resembles the output of an edge extraction system where the edgels are augmented with accurate depth information. In the sequel we will demonstrate that it is possible to produce compelling novel views based on this information.

Section 3 will describe a principled approach to reasoning about the 3D structure of the scene based on the quasi-sparse feature set produced by the EPI analysis. This analysis proceeds by considering the freespace volumes defined by the depth maps associated with each viewpoint. This analysis provides us with an implicit description of the 3D structure of the scene and allows us to correctly reproduce occlusions and disocclusions in the synthetic views. Importantly, this analysis could also be applied to the 3D results returned by range finders, stereo systems and structure from motion algorithms.

For the purposes of this discussion, previous approaches to the image based rendering problem can be divided into three categories. The first set of approaches are based on the plenoptic sampling approach described by Levoy and Hanrahan [9] and Gortler et al [5]. In these schemes, novel views are reproduced by sampling the appropriate rays from the input images. Shum and He [19] describe an interesting and effective approach for extending these techniques to immersive environments using a sampling system based on concentric mosaics. The method proposed in this paper differs from the techniques in this category by attacking the view generation problem as a morphing task rather than a plenoptic sampling problem.

The second category of approaches consists of techniques which proceed by constructing a detailed geometric model of the scene in the form of per pixel depth information for every image in the data set. Laveau and Faugeras [8], Pollefeys et al [13], Kanade et al [7] and Werner et al [23,6] all propose stereo based techniques for recovering the required depth or disparity maps from the input image data. Other researchers assume that the depth maps can be obtained from auxiliary range sensors [15,12]. Once these depth maps have been obtained, it is a relatively straightforward matter to produce a novel view of the scene by computing where each of the pixels in the original views will appear in the novel image. The Layered Depth Image representation proposed by Shade et al. [18] provides a particularly efficient method for rendering data sets of this

form. These authors also describes a method for determining the relative depth of points in the scene by estimating the motion of various layers in the input imagery.

The problem with recovering dense depth maps from image sequences is that there are two important situations where it is exceedingly difficult for traditional correlation-based stereo algorithms to produce accurate depth estimates. The first situation corresponds to texture free regions in the scene, such as blank walls, which do not produce a sufficiently distinctive correlation signature. Occluding edges in the scene can also cause significant difficulties since the regions in the image surrounding such an edge will contain half occluded regions which cannot be adequately matched between frames by the correlation metric. The proposed method overcomes these problems by employing an epipolar plane image analysis to recover the positions of the feature points rather than a correlation based approach and by using an interpolation scheme which produces acceptable results in texture free regions.

The third category of approaches consists of techniques that draw inspiration from the image morphing schemes described by Chen and Williams [2]. Seitz and Dyer [16] propose a technique for producing physically correct images by interpolating between a given pair of views. Lhuiller and Quan [10,11] describe a view morphing technique which also seeks to produce interpolated views which correctly reproduce the motion of salient points in the scene. They describe a scheme for triangulating the input image in such a way as to respect intensity discontinuities. These papers demonstrate that it is possible to produce compelling interpolated images from a relatively sparse set of correspondences. Both of these approaches deliberately avoid the problem of estimating the actual 3D locations of the feature points that are used as correspondences. This means that the techniques can be applied to uncalibrated imagery but it also limits the systems to producing views that lie along the straight line connecting the two original images. The technique proposed in this paper eliminates this restriction by estimating the actual 3D locations of the observed feature points. This allows the system to predict where the features will appear in any viewpoint.

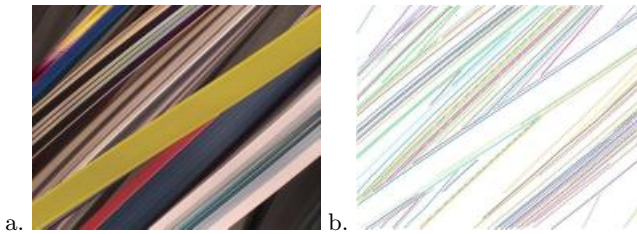
Genc and Ponce [4] propose an interesting approach to view interpolation by characterizing mathematically the possible positions of a selected set of feature points in the imagery. The proposed technique improves on this work by providing an automated technique for reproducing the motion of a much larger set of features in the scene, all of the edgels, and a more direct method for specifying the desired position of the novel viewpoint.

## 2 Obtaining Quasi-Sparse Feature Correspondences

As mentioned in the previous section, the Epipolar Plane Image (EPI) analysis technique proposed by Baker, Bolles, and Marimont [1] is used to recover the 3D positions of salient features in the scene. Other methods, such as feature based stereo or structure from motion could also be used for this purpose without affecting the correctness of subsequent arguments. EPI analysis is simply an

expedient, reliable, automatic technique for recovering the correspondences we are interested in.

In EPI analysis we consider the spatiotemporal intensity volume formed by collecting images at regularly spaced intervals as the camera moves along an axis parallel to one of the image axes. Figure 1a shows a slice of a typical spatiotemporal volume. This slice indicates how the intensity pattern along one row of the image changes as the camera moves along an axis parallel to that row. Note the characteristic banded structure of this image which can be explained by noting that feature points in the scene will correspond to straight lines in the epipolar plane imagery.



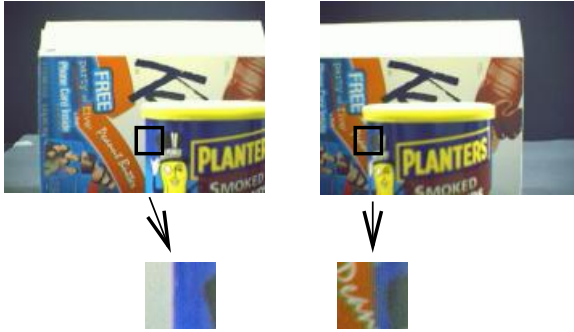
**Fig. 1.** a. A typical epipolar plane image obtained from the imagery. b. The result of applying the edge enhancement procedure to the epipolar plane image.

Each of these epipolar plane images is subjected to an analysis which seeks to extract these straight line features. This analysis is based on the techniques described by Baker et al [1] and Yamamoto [24]. A series of filters designed to enhance edges at various orientations is run over each epipolar plane image and the resulting edge elements are linked together to form straight line segments. Line segments that are deemed long enough and straight enough are interpreted as 3D feature points. Typical results obtained by invoking this procedure are shown in Figure 1b. The 3D location of the corresponding feature point in the scene is computed from the position and slope of the extracted line segments. The end result of the EPI analysis is a set of 3D points corresponding precisely to intensity discontinuities in the image.

In situations where the intensity discontinuity corresponds to the occluding edge of a curved object the trajectory in the spatiotemporal volume will correspond to the motion of the osculating ray. In most imaging situations this curve will still be a line to a good approximation since there will not be much variation in the position of the tangent point over the camera's trajectory.

Note that unlike correlation based approaches, this method for recovering the depth of feature points in the imagery does *not* assume that the intensity values surrounding corresponding points in the images will be strongly correlated. It simply exploits the fact that feature points in the scene correspond to straight lines in the EPI. The resulting line fitting problem is heavily overdetermined which serves to improve the accuracy and robustness of the method. This means

that the technique produces accurate estimates for the depth of occluding edges and other features that are problematic for correlation based methods. Occluding edges are particularly salient features in the scene and it is important to correctly predict where these features will appear in the novel view. Figure 2 shows an example of how drastically image regions corresponding to occluding edges can vary as the camera moves from one location to another.



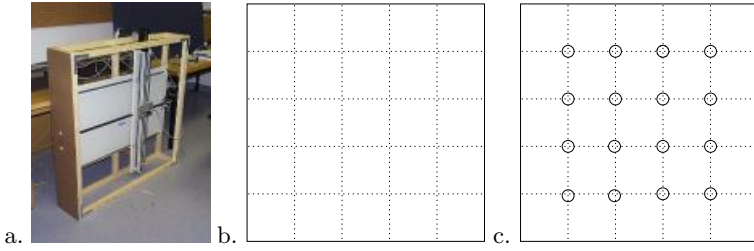
**Fig. 2.** The two neighborhoods shown differ greatly in appearance, even though they are centered around the same feature. Nevertheless, this feature corresponds to a single edge in the EPI and can, therefore, be recovered accurately.

One limitation of EPI analysis is that it will have difficulty recovering the depth of edges that are parallel to the direction of motion of the camera due to the aperture problem. In other words, a horizontal camera motion can be used to accurately recover the location of vertical edges but not horizontal edges and vice versa.

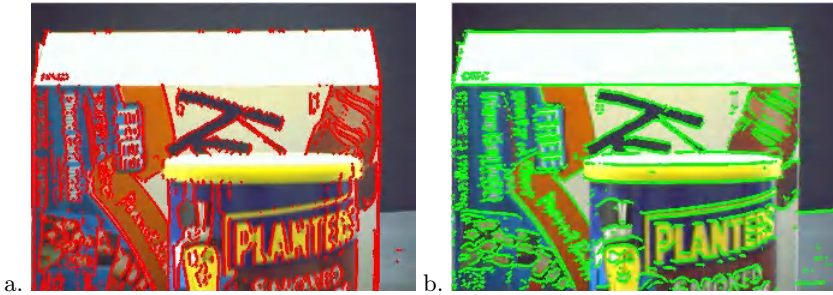
For this reason the gantry robot shown in Figure 3a is used to move the camera along two orthogonal axes. Images are acquired by scanning the camera along a set of horizontal and vertical paths as shown in Figure 3b. EPI analysis is applied to each scan in turn. For the images at the intersections of these scan paths we can combine the results of the horizontal and vertical analysis to produce depth estimates for all of the edgels in the image as shown in Figure 4. We refer to this structure as a Quasi-Sparse depth map since we have estimates for depth at a quasi-sparse collection of locations in the image, the intensity boundaries.

## 2.1 Interpolating Depth Values

Once a quasi-sparse depth map has been obtained for a particular image, one can produce a conservative approximation for the depth map associated with that viewpoint by obtaining the Delaunay triangulation of the projections of



**Fig. 3.** a. The XY-positioning table used in the experiments b. An illustration of the camera scan paths. c. The circles show the camera locations corresponding to the intersections of the scan paths. For these camera positions the system can combine the results of the horizontal and vertical analyses to produce a quasi-sparse depth map.



**Fig. 4.** a. A sample image with the pixels corresponding to visible recovered points from the horizontal camera motion shown in red. Notice that many points were recovered near vertical edges, and few were recovered near horizontal edges. b. The same image, but with visible recovered points shown in green. For vertical camera motion, there is a bias towards recovery near horizontal edges.

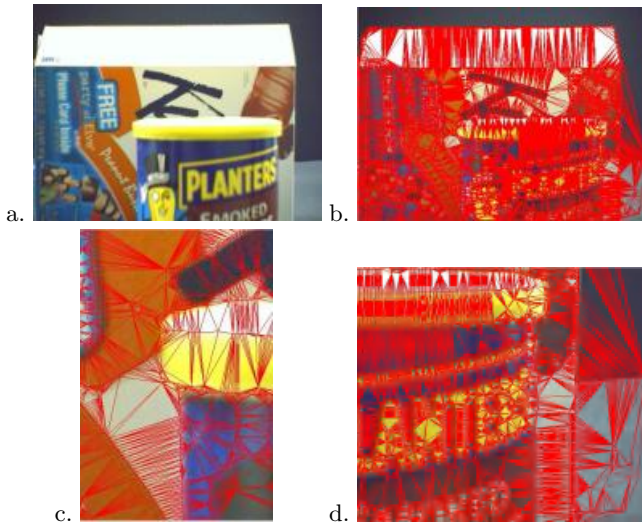
the feature points that are visible in that image<sup>1</sup> and linearly interpolating the  $1/z$  values within each triangle. This corresponds to fitting planar surfaces to each of triangles in the scene. To see this, consider the equation for a plane in space,  $ax + by + cz = 1$ . Dividing each side of this equation by  $z$  we obtain  $a(x/z) + b(y/z) + c = (1/z)$ , that is  $1/z$  is an affine function of  $x/z$  and  $y/z$  which implies that  $a'u + b'v + c' = (1/z)$  where  $(u, v)$  corresponds to the projection of the point  $(x, y, z)$  in the image. We conclude that for a plane  $1/z$  should be an affine function of the image coordinates  $u, v$ .<sup>2</sup>

Since the recovered scene points correspond to edges in the image, this construction has the agreeable property that the triangles tessellate the image without crossing image intensity discontinuities, which means that the resulting tri-

<sup>1</sup> It is a simple matter to record where various extracted features appear in the spatiotemporal volume used in the EPI analysis. We use this information to ensure that only the features that are known to be visible in a particular viewpoint are used to construct the depth map associated with that view.

<sup>2</sup> Hardware based texture mapping schemes also exploit this fact to their advantage.

angles have a strong tendency to correspond to facets of actual surfaces in the scene. Even triangles that do not correspond to surface facets encompass homogeneous regions in the image, such as blank walls, which can be morphed to the novel view in a convincing manner. The scheme also has an adaptive sampling property in that regions of the image that have a lot of intensity discontinuities are tessellated quite finely while other regions of the images that are less interesting are covered with fewer facets. This is appropriate from the perspective of view synthesis since we can usually get away with less refined interpolation in portions of the image that correspond to untextured areas. Figure 5 shows the results of a typical triangulation.

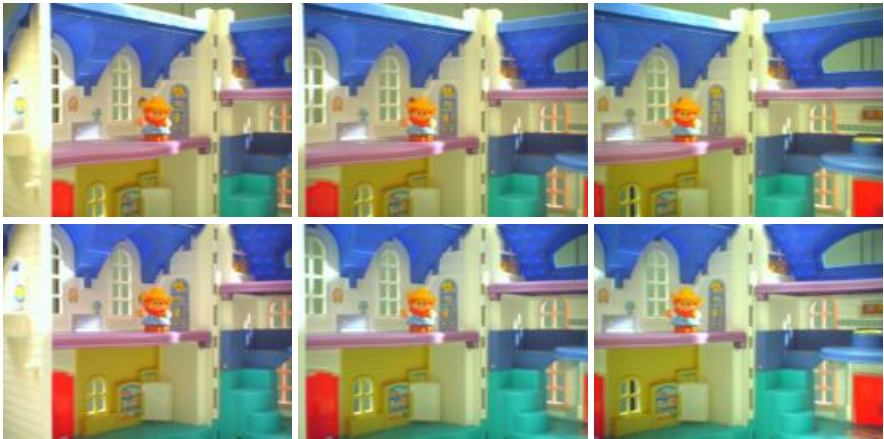


**Fig. 5.** a. An image taken from the intersection of a vertical and horizontal path. b. The triangulation of the projections of the visible recovered points. Notice that the vertices of these triangles lie along intensity edges. c. A close-up view of the left side of the canister. d. A close-up view of the right side of the canister. Notice that texture-free areas are covered by large triangles, while more detailed image areas have dense triangulations.

At the end of this stage, the system has constructed a set of triangular facets for each of the input images and the locations of the vertices of these triangles in the scene are known. At this point the system can generate novel views of the scene by supplying these triangles to a standard rendering pipeline and using the original images as texture maps. The rendering system correctly accounts for the parallax induced as a result of the motion of the virtual viewpoint and hidden surface removal reproduces the majority of the occlusion and disocclusion events that would be observed as the camera moves.

Note that in this framework any of the original images could be morphed to the novel viewpoint. In the current implementation the simple expedient of choosing the closest original viewpoint as a basis for morphing is employed with the idea that minimizing the difference in position between novel and original viewpoints will minimize the errors introduced by the viewpoint morphing operation. Results obtained by applying this simple morphing operation to a sample scene are shown in Figure 7. In most situations, this simple scheme produces acceptable results since it correctly reproduces the motion of the most perceptually salient features in the image, the intensity edges.

Unlike plenoptic sampling techniques, this approach only requires us to store the images taken at the intersections of the scanpaths and the associated depth maps. However, we are still able to produce convincing synthetic images from a reasonable range of viewpoints.



**Fig. 6.** 6 Original images of a doll house scene used in our view synthesis experiments



**Fig. 7.** Synthetic images produced by using the interpolated depth maps to morph the original images to novel vantage points

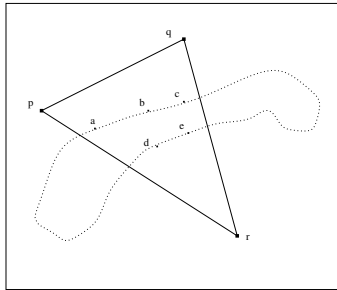


### 3 Reasoning about 3D Structure

While the procedure described in the previous section produces depth maps that can be used to perform novel view synthesis, it cannot predict the appearance of regions which are not visible from a given input view. In order to correctly reproduce the appearance of these regions we must combine information from multiple images into a composite description of the scene.

When accurate range scans are available, several excellent techniques are available for merging this information into a coherent surface [3,7,20]. Unfortunately, these techniques cannot be directly applied in this case since the interpolated range maps may underestimate the depth of the scene in a given image. We propose an alternative approach based on the following observation: while the individual interpolated depth maps may underestimate the depth of the surface in places, they do accurately predict the *freespace* in the scene. The following theorem states this observation more concisely:

**Freespace Theorem :** Suppose that three recovered space points  $P$ ,  $Q$ , and  $R$  project to the pixel locations  $p$ ,  $q$ , and  $r$  respectively in a given input image, and suppose further that  $\triangle pqr$  is one of the Delaunay triangles formed in that image. Then the tetrahedron formed by the camera center and the space triangle  $\triangle PQR$  must consist entirely of free space, i.e. no surface in the scene intersects this tetrahedron.



**Fig. 8.** If the space triangle  $\triangle PQR$  is occluded by another surface in the scene when viewed from a particular vantage point then the corresponding triangle in that image,  $\triangle pqr$ , would contain points corresponding to the boundary of the occluding surface. This would preclude the triangle  $\triangle pqr$  from the Delaunay triangulation of the image.

**Proof:** Assume, to the contrary, that there *is* a surface within this tetrahedron. Then the surface cannot occlude any of the points  $P$ ,  $Q$ , or  $R$  (or else the occluded point would not be visible in the image). Therefore the boundary of the occluding surface must lie at least partially inside the tetrahedron. Points from the boundary would then show up as edges in either the EPI generated by horizontal camera motion or the EPI generated by vertical camera motion, and so we would expect this to cause at least one point to be recovered that has its

projection inside  $\triangle pqr$  (see Figure 8). This contradicts our assumption that  $p$ ,  $q$ , and  $r$  are part of a single Delaunay triangle, since the interiors of Delaunay triangles cannot contain other vertices<sup>3</sup>. Hence, the tetrahedron must consist entirely of free space. ♣

Note that this argument rests on the assumption that we have a procedure that is capable of accurately reconstructing the depths of the majority of the edge features in an image. If the reconstruction system only returned the depths of isolated corner features in the image, this property might not hold.

We can use the freespace theorem to draw conclusions about the 3D structure of the scene. First consider the star shaped freespace volumes defined by the interpolated depth maps associated with each of the input viewpoints. Then consider the union of these freespace regions as shown in Figure 9. Notice that the union provides a more accurate approximation of the structure of the scene than any of the original depth maps. It is also important to notice that the input depth maps need not be particularly accurate; they may underestimate the scene depth in places but these errors can be discovered and corrected through the freespace union procedure.

Given the coordinates of a point in space,  $P$ , we can easily test whether that point is in the union of the freespace volumes by projecting the point into each of the original viewpoints and determining whether the depth of the point  $P$  with respect to the image center is less than the depth of the corresponding entry in the original interpolated depth map for that image.

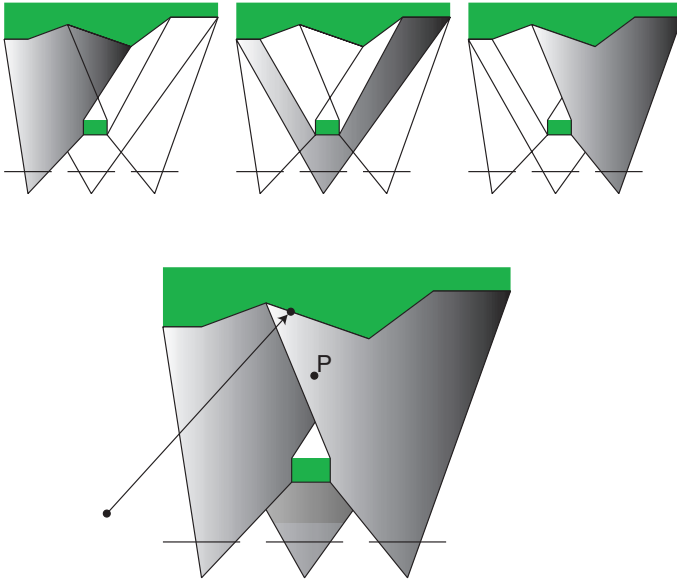
This procedure can be represented by an indicator function  $\Phi(P) : R^3 \rightarrow [0, 1]$  which returns 0 whenever the point  $P$  lies within the union of the freespace volumes and 1 otherwise. This function can be thought of as an *implicit* representation of the total freespace volume.

Armed with this function,  $\Phi(P)$ , we can determine where a ray in space intersects the freespace boundary by sampling this function at various points along the ray in search of a transition. Once a transition has been found it can be localized rapidly using standard bisection search techniques [14].

This ray intersection procedure can be used to determine the depth from any given vantage point to the freespace boundary as shown in Figure 9. It can also be used to determine whether or not a particular point is visible from a given camera position since this amounts to testing whether the ray between the given point and the camera center lies entirely in freespace.

The boundary of the freespace volume can be thought of as a fair approximation for the surface geometry in the sense that it will correspond to a surface which is consistent with all of the available information. Notice, however, that the actual surface may contain regions that are not visible from any of the camera positions. The freespace union method will still produce reasonable results in these situations.

<sup>3</sup> In fact, the defining property of Delaunay triangulations is that the interior of the *circumcircle* of every triangle does not contain any other vertices; however, we do not require such a strong property for our argument.



**Fig. 9.** The first row of figures depicts the freespace volumes associated with the triangulations of each of the 3 input images. The union of these volumes provides a more accurate approximation of the 3D structure of the scene than any of the original depth maps. It is a simple matter to construct a function  $\Phi(P)$  which indicates whether a particular point  $P$  lies within the union of the freespace volumes.

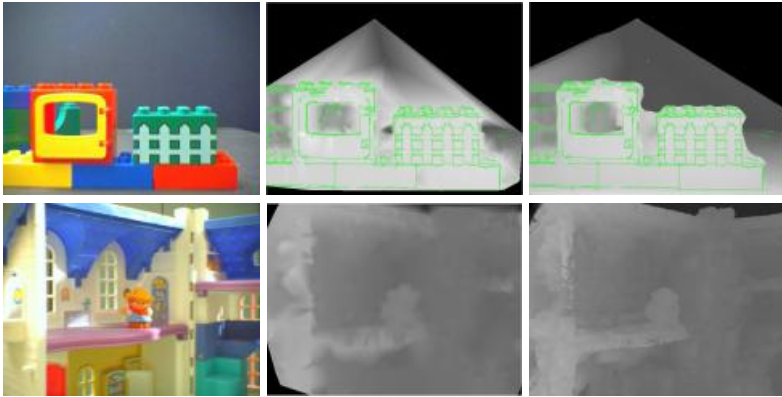
One could, in principle, construct an explicit representation for the freespace boundary from the indicator function  $\Phi(P)$  by invoking an isosurface algorithm such as Marching Cubes [17]. However, this would involve deciding on an appropriate discretization of the scene volume. Coarse discretizations would result in meshes that failed to capture fine details of the surface but fine discretizations would produce exceedingly large meshes. Fortunately, novel views of the scene can be synthesized without the aid of an explicit surface.

## 4 Novel View Synthesis

The techniques described in the previous section can be employed to refine the conservative depth maps associated with each of the input images. For every pixel in a given input image we can construct the ray passing between the camera center and the pixel center and determine where that ray intersects the freespace boundary. This analysis gives us a refined estimate for the depth of the scene at that pixel.

The resulting depth maps can be further refined by applying a smoothing procedure similar to anisotropic diffusion which takes into account the locations of intensity discontinuities in the original image and preserves salient depth discontinuities in the scene. The ultimate goal of this smoothing procedure is to

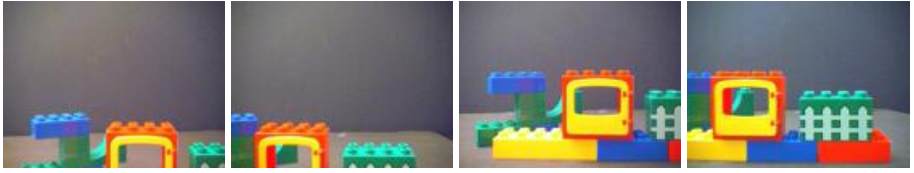
produce depth maps that have crisp boundaries corresponding to edges in the intensity image. In this sense, the method is similar in spirit to the scheme proposed by Tao, Sawhney and Kumar [21,22]. However, our approach to producing these depth maps based on an analysis of the freespace volume is completely different.



**Fig. 10.** The first column of figures corresponds to the input images, the second column depicts the original interpolated depth maps and the last column shows the results of applying the freespace analysis procedure to refine the depth maps. Notice that freespace analysis correctly recovers many of the sharp depth discontinuities that were missing in the original interpolated depth maps.

Figure 10 shows the results of applying this procedure to typical input images. The refined depth maps shown in Figure 10 are remarkably accurate considering that they were produced from quasi-sparse depth samples. Note that in both cases the procedure correctly recovers many of the sharp depth discontinuities associated with occluding edges in the scene, discontinuities that were not present in the original interpolated depth maps. When these depth maps are used to morph the corresponding input images to a novel vantage point, these depth discontinuities can produce holes in the synthetic image. This problem is handled by morphing several input views to the new viewpoint and combining the resulting images to fill the gaps. Figures 12 and 13 show some of the results obtained by applying this view synthesis procedure to actual image data.

The scene with the Lego blocks was chosen precisely because the texture free surfaces and occluding edges would pose a challenge for most correlation based stereo algorithms. Notice that the view synthesis procedure correctly reproduces the appearance of the distant green block which is seen through the window of the foreground Lego block. This demonstrates that the system is able to reason correctly about the complex 3D structure of the scene.



**Fig. 11.** Original images of a scene of blocks that were used to produce novel views



**Fig. 12.** Synthetic images produced by freespace analysis method.



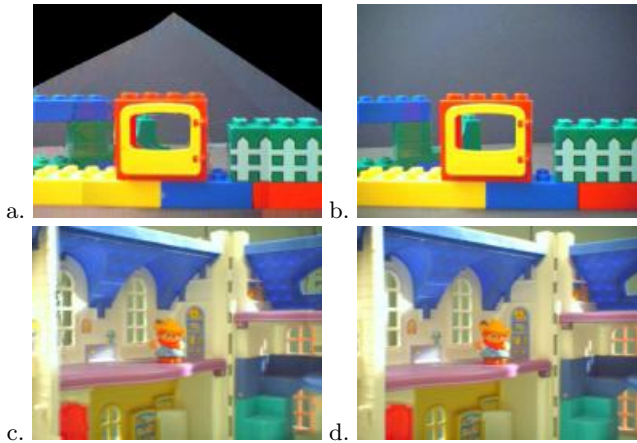
**Fig. 13.** Synthetic images produced by freespace analysis method.

Figure 14 shows a side by side comparison of actual images and the corresponding synthetic views. The normalized correlation coefficient between the synthetic views and the actual images is greater 0.94 in both cases.

## 5 Conclusions

This paper describes an approach to novel view synthesis based on a quasi-sparse set of feature correspondences. This scheme hinges on the observation that human viewers tend to be sensitive to the motion of intensity discontinuities so the system focuses its efforts on recovering accurate estimates for the depths of these features. The resulting quasi-sparse depth maps resemble the output of an edge extraction procedure where the edgels have been augmented with accurate depth information. In each of the input images a Delaunay triangulation of the feature points is used to produce an interpolated depth maps which provide an approximation of the surface structure that can be used to morph the input views to novel vantage points.

We have also presented a principled approach to reasoning about the 3D structure of the scene by analyzing the union of the freespace volumes associated with each of the interpolated depth maps. This freespace union procedure provides a powerful constraint which allows us to produce relatively accurate



**Fig. 14.** The first column corresponds to synthetic views and the second to actual images of the scenes. Note that the view synthesis procedure correctly reproduces the motion of the green block which is viewed through the window of the foreground lego block.

depth maps from quasi-sparse feature correspondences in multiple images. Importantly, the arguments made about analyzing the freespace volumes could also be applied to situations where depth information is obtained using other means such as range finders or stereo systems. This analysis allows us to correctly reproduce occlusions and disocclusions in the synthetic views.

## 5.1 Future Work

The current system makes use of Epipolar Plane Image analysis since it provided a simple and accurate automated method for obtaining the depth of the salient features in the scene. It should be possible to obtain similar information about the scene features from standard video sequences by applying structure from motion techniques. It should also be possible to extend the view synthesis technique so that it could take advantage of correspondences provided by correlation based stereo methods whenever they are available.

Better methods for smoothing the refined depth maps based on information in the intensity images are being investigated. We also intend to consider extensions of the scheme that will allow us to model how the radiance of observed scene points changes as a function of viewing direction so that we can correctly reproduce specularities and other view dependent effects [15].

## References

1. R.C. Bolles, H.H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

2. S. E. Chen and L. Williams. View interpolation from image synthesis. In *SIGGRAPH*, pages 279–288, August 1993.
3. Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH 96. In Computer Graphics Proceedings, Annual Conference Series*, pages 31–43, New Orleans, LA, August 4-9 1996. ACM SIGGRAPH.
4. Y. Genc and J. Ponce. Parameterized image varieties: A novel approach to the analysis and synthesis of image sequences. In *International Conference on Computer Vision*, pages 11–16, January 1998.
5. Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael Cohen. The lumigraph. In *Proceedings of SIGGRAPH 96. In Computer Graphics Proceedings, Annual Conference Series*, pages 31–43, New Orleans, LA, August 4-9 1996. ACM SIGGRAPH.
6. V. Hlavac, A. Leonardis, and T. Werner. Automatic selection of reference views for image-based scene representations. In *European Conference on Computer Vision*, pages 526–535, 1996.
7. T. Kanade, P.W. Rander, and J. P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.
8. S. Laveau and O.D Faugeras. 3-d scene representation as a collection of images. In *International Conference on Pattern Recognition*, pages 689–691, 1994.
9. Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of SIGGRAPH 96. In Computer Graphics Proceedings, Annual Conference Series*, pages 31–43, New Orleans, LA, August 4-9 1996. ACM SIGGRAPH.
10. Maxime Lhuiller and Long Quan. Image interpolation by joint view triangulation. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 139–145, 1999.
11. Maxime Lhuiller and Long Quan. Edge-constrained joint view triangulation for image interpolation. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 218–224, 2000.
12. Ko Nishino, Yoichi Sato, and Katsui Ikeuchi. Eigen-texture method: Appearance compression based on 3d model. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 618–624, 1999.
13. M. Pollefeys, L Van Gool, and M. Proesmans. Euclidean 3d reconstruction from image sequences with variable focal lengths. In *European Conference on Computer Vision*, pages 31–42, 1996.
14. W. Press, B.Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
15. Y. Sato, M.D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Proceedings of SIGGRAPH 97. In Computer Graphics Proceedings, Annual Conference Series*, pages 379–387. ACM SIGGRAPH, August 1997.
16. Steven Seitz and Charles R. Dyer. View morphing. In *Proceedings of SIGGRAPH 96. In Computer Graphics Proceedings, Annual Conference Series*, pages 31–43, New Orleans, LA, August 4-9 1996. ACM SIGGRAPH.
17. J.A. Sethian. *Level Set Methods; Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences*. Cambridge University Press, 1996.

18. Jonathan Shade, Steven Gortler, Li wei He, and Richard Szeliski. Layered depth images. In *Proceedings of SIGGRAPH 98. In Computer Graphics Proceedings, Annual Conference Series*, pages 231–242. ACM SIGGRAPH, August 1998.
19. Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics. In *SIGGRAPH*, pages 299–306, August 1999.
20. Ioannis Stamos and Peter Allen. 3-d model construction using range and image data. In *IEEE Conference on Computer Vision and Pattern Recogniton*, 2000.
21. Hai Tao and Harpreet Sawhney. Global matching criterion and color segmentation based stereo. In *Workshop on the Applications of Computer Vision*, pages 246–253, December 2000.
22. Hai Tao, Harpreet Sawhney, and Rakesh Kumar. A global matching framework for stereo computation. In *International Conference on Computer Vision*, pages 532–539, 2001.
23. T. Werner, R.D. Hersch, and V. Hlavac. Rendering real-world objects using view interpolation. In *International Conference on Computer Vision*, pages 957–962, 1995.
24. Masanobu Yamamoto. *The Image Sequence Analysis of Three-Dimensional Dynamic Scenes*. PhD thesis, Electrotechnical Laboratory, Agency of Industrial Science and Technology, May 1988.



# Eye Gaze Correction with Stereovision for Video-Teleconferencing

Ruigang Yang<sup>\*1</sup> and Zhengyou Zhang<sup>2</sup>

<sup>1</sup> Dept. of Computer Science, University of North Carolina at Chapel Hill, North Carolina, USA  
ryang@cs.unc.edu,

<sup>2</sup> Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA  
zhang@microsoft.com

**Abstract.** The lack of eye contact in desktop video teleconferencing substantially reduces the effectiveness of video contents. While expensive and bulky hardware is available on the market to correct eye gaze, researchers have been trying to provide a practical software-based solution to bring video-teleconferencing one step closer to the mass market. This paper presents a novel approach that is based on stereo analysis combined with rich domain knowledge (a personalized face model). This marriage is mutually beneficial. The personalized face model greatly improved the accuracy and robustness of the stereo analysis by substantially reducing the search range; the stereo techniques, using both feature matching and template matching, allow us to extract 3D information of objects other than the face and to determine the head pose in a much more reliable way than if only one camera is used. Thus we enjoy the versatility of stereo techniques without suffering from their vulnerability. By emphasizing a 3D description of the scene on the face part, we synthesize virtual views that maintain eye contact using graphics hardware. Our current system is able to generate an eye-gaze corrected video stream at about 5 frames per second on a commodity PC.

**Keywords:** Stereoscopic vision, Eye-gaze correction, Structure from motion.

## 1 Introduction

Video-teleconferencing, a technology enabling communicating with people face-to-face over remote distances, does not seem to be as widespread as predicted. Among many problems faced in video-teleconferencing, such as cost, network bandwidth, and resolution, the lack of eye-contact seems to be the most difficult one to overcome[18]. The reason for this is that the camera and the display screen cannot be physically aligned in a typical desktop environment. It results in unnatural and even awkward interaction. Special hardware using half-silver mirrors has been used to address this problem. However this arrangement is bulky and the cost is substantially high. What's more, as a piece of dedicated equipment, it does not fit well to our familiar computing environment, thus its usability is greatly reduced. We aim to address the eye-contact problem by synthesizing videos as if they were taken from a camera behind the display screen, thus to establish natural eye-contact between video-teleconferencing participants without using any kind

---

<sup>\*</sup> This work was mainly conducted while the first author was at Microsoft Research as a summer intern.

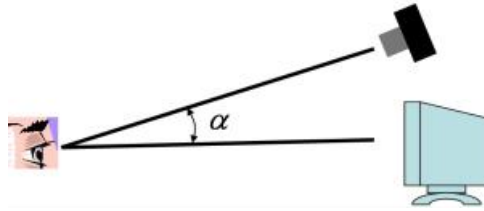
of special hardware. The approach we take involves three steps: pose tracking, view matching, and view synthesis. We use a pair of calibrated stereo cameras and a personalized face model to track the head pose in 3D. The use of strong domain knowledge (a personalized face model) and a stereo camera pair greatly increase the robustness and accuracy of the 3D head pose tracking. The stereo camera pair also allows us to match parts not modelled in the face model, such as the hands and shoulders, thus providing wider coverage of the subject. Finally, the results from the head tracking and stereo matching are combined to generate a virtual view. Unlike other methods that only “cut and paste” the face part of the image, our method generates natural looking and seamless images, as shown in figure 1.



**Fig. 1.** Eye-gaze Correction: The two images shown on the left are taken from a pair of stereo cameras mounted on the top and bottom sides of a monitor; the image shown above is a synthesized virtual view that preserves eye-contact.

There have been attempts to address the eye-contact problem using a single camera with a face model [12],[10], or using dense stereo matching techniques[19], [15]. We will discuss these approaches in more details in Section 2. With a single camera, we found it difficult to maintain both the real-time requirement and the level of accuracy we want with head tracking. Existing model-based monocular head tracking methods [11], [4], [2], [7], [1] either use a simplistic model so they could operate in real time but produce less accurate results, or use some sophisticated models and processing to yield highly accurate results but take at least several seconds to compute. A single-camera configuration also has difficulties to deal with occlusions. Considering these problems with a monocular system, we decided to adopt a stereo configuration. The important epipolar constraint in stereo allows us to reject most outliers without using expensive robust estimation techniques, thus keeping our tracking algorithm both robust and simple enough to operate in real-time. Furthermore, two cameras usually provide more coverage of the scene.

One might raise the question that why we do not use a dense stereo matching algorithm. We argue that, first, doing a dense stereo matching on a commodity PC in real time is difficult, even with today’s latest hardware. Secondly and most importantly, a dense stereo matching is unlikely to generate satisfactory results due to the limitation on camera placement. Aiming at desktop video conferencing applications, we could only put the cameras around the frame of a display monitor. If we put the cameras on the opposite edges of the display, given the normal viewing distance, we have to converge the cameras towards the person sitting in front of the desktop, and such a stereo system will have a long baseline. That makes stereo matching very difficult; even if we were able to get a perfect matching, there would still be a significant portion of the subject which is occluded in one view or the other. Alternatively, if we put the cameras close to



**Fig. 2.** Camera-Screen displacement causes the lose of eye-contact.

each other on the same edge of the monitor frame, the occlusion problem is less severe, but generalization to new distant views is poor because a significant portion of the face is not observed. After considering various aspects, we have decided to put one camera on the upper edge of the display and the other on the lower edge, and follow a model-based stereo approach to eye-gaze correction.

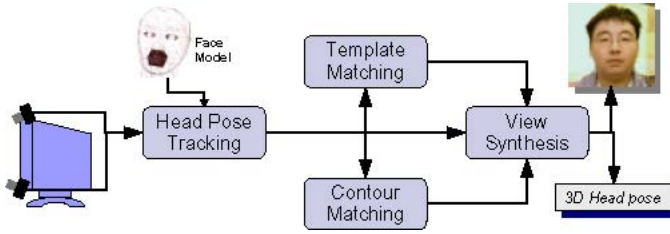
## 2 Related Works

In a typical desktop video-teleconferencing setup, the camera and the display screen cannot be physically aligned, as depicted in figure 2. A participant looks at the image on the monitor but not directly into the camera, thus she does not appear to make eye contact with the remote party. Research [23] has shown that if the divergence angle ( $\alpha$ ) between the camera and the display is greater than five degrees, the loss of eye-contact is noticeable. If we mount a small camera on the side of a 21-inch monitor, and the normal viewing position is at 20 inches away from the screen, the divergence angle will be 17 degrees, well above the threshold at which the eye-contact can be maintained. Under such a setup, the video loses much of its communication value and becomes un-effective compared to telephone.

Several systems have been proposed to reduce or eliminate the angular deviation using special hardware. They make use of half-silvered mirrors or transparent screens with projectors to allow the camera to be placed on the optical path of the display. A brief review of these hardware-based techniques has been given in [14]. The expensive cost and the bulky setup prevent them to be used in a ubiquitous way.

On the other track, researchers have attempted to create eye-contact using computer vision and computer graphics algorithms. Ott et al. [19] proposed to create a virtual center view given two cameras mounted on either side of the display screen. Stereoscopic analysis of the two camera views provides a depth map of the scene. Thus it is possible to “rotate” one of the views to obtain a center virtual view that preserves eye contact. Similarly, Liu et al. [15] used a trinocular stereo setup to establish eye contact. In both cases, they perform dense stereo matching without taking into account the domain knowledge. While they are generic enough to handle a variety of objects besides faces, they are likely to suffer from the vulnerability of brute-force stereo matching. Furthermore, as discussed in the previous section, we suspect that direct dense stereo matching is unlikely to generate satisfactory results due to the constraint of camera placement imposed by the size of the display monitor – a problem that may be less severe back in the early 90’s, when the above two algorithms were proposed .

Cham and Jones at Compaq Cambridge Research Laboratory [6] approached this problem from a machine learning standpoint. They first register a 2D face model to



**Fig. 3.** The components of our eye-gaze correction system

the input image taken from a single camera, then morph the face model to the desired image. The key is to learn a function that maps the registered face model parameters to the desired morphed model parameters [12]. They achieve this by non-linear regression from sufficient instances of registered-morphed parameter pairs which are obtained from training data. As far as we know, their research is still in a very early stage, so it is not clear if this approach is capable of handling dramatic facial expression changes. Furthermore, they only deal with the face part of the image – the morphed face image is superimposed on the original image frame, which sometime leads to errors near the silhouettes due to visibility changes.

The GazeMaster project at Microsoft Research [10] uses a single camera to track the head orientation and eye positions. Their view synthesis is quite unique in that they first replace the human eyes in a video frame with synthetic eyes gazing in the desired direction, then texture-map the eye-gaze corrected video frame to a generic rigid face model rotated to the desired orientation. The synthesized photos they published look more like avatars, probably due to the underlying generic face model. Another drawback is that, as noted in their report, using synthetic eyes sometime inadvertently changes the facial expression as well.

From a much higher level, this GazeMaster work is similar to our proposed approach, in the sense that they both use strong domain knowledge (a 3D face model) to facilitate the tracking and view synthesis. However, our underlying algorithms, from tracking to view synthesis, are very different from theirs. We incorporate a stereo camera pair, which provides the important epipolar constraint that we use throughout the entire process. Furthermore, the configuration of our stereo camera provides much wider coverage of the face, allowing us to generate new distant views without having to worry about occlusions.

### 3 System Overview

Figure 3 illustrates the block diagram of our eye-gaze correction system. We use two digital video cameras mounted vertically, one on the top and the other on the bottom of the display screen. They are connected to a PC through 1394 links. The cameras are calibrated using the method in [24]. We choose the vertical setup because it provides wider coverage of the subject and higher disambiguation power in feature matching. Matching ambiguity usually involves symmetric facial features such as eyes and lip contours aligned horizontally. The user’s personalized face model is acquired using a rapid face modelling tool [16]. Both the calibration and model acquisition require little human interaction, and a novice user can complete these tasks within 15 minutes.

With these prior knowledge, we are able to correct the eye-gaze using the algorithm outlined as follows:

1. Background model acquisition
2. Face tracker initialization
3. For each image pair, perform
  - Background subtraction
  - Temporal feature tracking in both images
  - Updating head pose
  - Correlation-based stereo feature matching
  - Stereo silhouette matching
  - Hardware-assisted view synthesis

Currently, the only manual part of the system is the face tracker initialization which requires the user to interactively select a few markers. We are currently working on automatic initialization.

The tracking subsystem includes a feedback loop that supplies fresh salient features at each frame to make the tracking more stable under adversary conditions, such as partial occlusions and facial expression changes. Furthermore, an automatic tracking recovery mechanism is also implemented to make the whole system even more robust over extended period of time. Based on the tracking information, we are already able to manipulate the head pose by projecting the live images onto the face model. However, we also want to capture the subtleties of facial expressions and other foreground objects, such as hands and shoulders. So we further conduct correlation-based feature matching and silhouette matching between the stereo images. All the matching information, together with the tracked features, is used to synthesize a *seamless* virtual image that looks as if it were taken from a camera behind the display screen. We have implemented the entire system under the MS Windows environment. Without any effort spending on optimizing the code, our current implementation runs about 4-5 frames per second on a single CPU 1 GHz PC.

## 4 Stereo 3D Head Pose Tracking

Our stereo head pose tracking problem can be stated as follows:

- Given** (i) a set of triplets  $S = \{(\mathbf{p}, \mathbf{q}, \mathbf{m})\}$  at time  $t$ , where  $\mathbf{p}$  and  $\mathbf{q}$  are respectively points in the upper (first) and the lower (second) camera images, and  $\mathbf{m}$  is their corresponding point in the face model, and (ii) a pair of images from the stereo cameras at time  $t + 1$ ,
- determine** (i)  $S' = \{(\mathbf{p}', \mathbf{q}', \mathbf{m})\}$  at time  $t + 1$  where  $\mathbf{p}'$  and  $\mathbf{q}'$  are the new positions of  $\mathbf{p}$  and  $\mathbf{q}$ , and (ii) compute the head pose, so that the projections of  $\mathbf{m}$  in time  $t + 1$  are  $\mathbf{p}'$  and  $\mathbf{q}'$  in the stereo image pair, respectively.

We use the KLT tracker to track feature points  $\mathbf{p}, \mathbf{q}$  from time  $t$  to  $t + 1$  [22]. Note that there is one independent feature tracker for each camera, thus we need apply the epipolar constraint to remove any stray point. The epipolar constraint states that if a point  $\mathbf{p} = [u, v, 1]^T$  (expressed in homogeneous coordinates) in the first image and a

point  $\mathbf{q} = [u', v', 1]^T$  in the second image correspond to the same 3D point  $\mathbf{m}$  in the physical world, they must satisfy the following equation:

$$\mathbf{q}^T \mathbf{F} \mathbf{p} = 0 \quad (1)$$

where  $\mathbf{F}$  is the fundamental matrix that encodes the epipolar geometry between the two images [9]. In fact,  $\mathbf{F} \mathbf{p}$  defines the epipolar line in the second image, thus Equation (1) means that the point  $\mathbf{q}$  must pass through the epipolar line  $\mathbf{F} \mathbf{p}$  and vice versa.

In practice, due to camera noise and inaccuracy in camera calibration and feature localization, we define a band of uncertainty along the epipolar line. For every triplet  $(\mathbf{p}', \mathbf{q}', \mathbf{m})$ , if the distance from  $\mathbf{q}'$  to the  $\mathbf{p}'$ 's epipolar line is greater than a certain threshold, this triplet will be discarded. We use a distance threshold of three pixels in our experiment.

After we have removed all the stray points that violate the epipolar constraint, we update the head pose, represented by a  $3 \times 3$  rotational matrix  $\mathbf{R}$  and a 3D translation vector  $\mathbf{t}$ , so that the sum of re-projection errors of  $\mathbf{m}$  to  $\mathbf{p}'$  and  $\mathbf{q}'$  is minimized. The re-projection error  $e$  is defined as

$$e = \sum_i \|\mathbf{p}'_i - \phi(\mathbf{A}^0(\mathbf{R}\mathbf{m}_i + \mathbf{t}))\|^2 + \|\mathbf{q}'_i - \phi(\mathbf{A}^1[\mathbf{R}^{10}(\mathbf{R}\mathbf{m}_i + \mathbf{t}) + \mathbf{t}^{10}])\|^2 \quad (2)$$

where  $\phi(\cdot)$  represents the standard pinhole projection,  $\mathbf{A}^0$  and  $\mathbf{A}^1$  are the cameras' intrinsic parameters, and  $(\mathbf{R}^{10}, \mathbf{t}^{10})$  is the transformation from the second camera's coordinate system to the first camera's. Solving  $(\mathbf{R}, \mathbf{t})$  by minimizing (2) is a nonlinear optimization problem. We can use the head pose from time  $t$  as the initial guess and conduct the minimization by means of, for example, the Levenberg-Marquardt algorithm.

After the head pose is determined, we replenish the matched set  $S'$  by adding more good feature points selected using the criteria in [22]. A good feature point is a point with salient textures in its neighborhood. We must be careful not to add feature points in the non-rigid parts of the face, such as the mouth region. To do so, we define a bounding box around the tip of the nose that covers the forehead, eyes, and nose region. Any good feature points outside this bounding box will not be added to the matched set. However, they will be used in the next stereo matching stage, which we will discuss in Section 5.2.

The replenish scheme greatly improves the robustness of the tracking algorithm. Our experiments have shown that our tracking can keep tracking under large head rotations and dramatic facial expression changes.

#### 4.1 Tracker Initialization and Auto-Recovery

The tracker needs to know the head pose at time 0 to start tracking. We let the user interactively select seven landmark points in each image, from which the initial head pose can be determined. The initial selection is also used for tracking recovery when the tracker loses tracking. This may happen when the user moves out of the camera's field of view or rotates her head away from the cameras. Fortunately, for our video-conferencing application, we could just send one of the video streams un-modified for these cases. When she turns back to look at the screen, we do need to continue tracking with no human intervention, which requires automatic recovery of the head pose. During the tracking recovery process, the initial set of landmark points is used as templates to find the best match in the current image. When a match with a high confidence value is found, the

tracker continues the normal tracking. Our recovery scheme is effective because unlike most other tracking applications, we only need to track the user when she is looking at the display window. This is exactly the scenario when the initial landmark templates are recorded.

Furthermore, we also activate the auto-recovery process whenever the current head pose is close to the initial head pose. This prevents the tracker from drifting. A more elaborated scheme that uses multiple templates from different head poses can further improve the effectiveness of automatic tracker reset. This can be further extended to a non-parametric description of head poses that can self-calibrate over time.

## 5 Stereo View Matching

The result from the 3D head pose tracking gives a set of good matches within the *rigid* part of the face between the stereo pair. To generate convincing and photo-realistic virtual views, we need to find more matching points over the entire foreground images, especially along the contour and the non-rigid parts of the face. We incorporate both feature matching and template matching to find as many matches as possible. During this matching process, we use the reliable information obtained from tracking to constrain the search range. In areas where such information is not available, such as hands and shoulders, we relax the search threshold, then apply the disparity gradient limit to remove false matches. To facilitate the matching (and later view synthesis in Section 6), we rectify the images using the technique described in [17], so that the epipolar lines are horizontal.

### 5.1 Disparity and Disparity Gradient Limit

Before we present the details of our matching algorithm, it is helpful to define disparity, disparity gradient, and the important principle of disparity gradient limit, which will be exploited through out the matching process.

Disparity is well defined for parallel cameras (i.e., the two image planes are the same) [9], and this is the case because we perform stereo rectification such that the horizontal axes are aligned in both images. Given a pixel  $(u, v)$  in the first image and its corresponding pixel  $(u', v')$  in the second image, disparity is defined as  $d = u' - u$  ( $v = v'$  as images have been rectified). Disparity is inversely proportional to the distance of the 3D point to the cameras. A disparity of 0 implies that the 3D point is at infinity.

Consider now two 3D points whose projections are  $\mathbf{m}_1 = [u_1, v_1]^T$  and  $\mathbf{m}_2 = [u_2, v_2]^T$  in the first image, and  $\mathbf{m}'_1 = [u'_1, v'_1]^T$  and  $\mathbf{m}'_2 = [u'_2, v'_2]^T$  in the second image. Their disparity gradient is defined to be the ratio of their difference in disparity to their distance in the cyclopean image, i.e.,

$$DG = \left| \frac{d_2 - d_1}{u_2 - u_1 + (d_2 - d_1)/2} \right| \quad (3)$$

Experiments in psychophysics have provided evidence that human perception imposes the constraint that the disparity gradient  $DG$  is upper-bounded by a limit  $K$ . The limit  $K = 1$  was reported in [5]. The theoretical limit for opaque surfaces is 2 to ensure that the surfaces are visible to both eyes [20]. Also reported in [20], less than 10% of world surfaces viewed at more than 26cm with 6.5cm of eye separation will present with disparity gradient larger than 0.5. This justifies use of a disparity gradient limit well below the theoretical value (of 2) without imposing strong restrictions on the world

surfaces that can be fused by the stereo algorithm. In our experiment, we use a disparity gradient limit of 0.8 ( $K = 0.8$ ).

## 5.2 Feature Matching Using Correlation

For unmatched good features in the first (upper) image, we try to find corresponding points, if any, in the second (lower) image by template matching. We use normalized correlation over a  $9 \times 9$  window to compute the matching score. The disparity search range is confined by existing matched points from tracking, when available.

Combined with matched points from tracking, we build a sparse disparity map for the first image and use the following procedure to identify potential outliers (false matches) that do not satisfy the disparity gradient limit principle. For a matched pixel  $\mathbf{m}$  and a neighboring matched pixel  $\mathbf{n}$ , we compute their disparity gradient between them using (3). If  $DG \leq K$ , we register a vote of good match for  $\mathbf{m}$ ; otherwise, we register a vote of bad match for  $\mathbf{m}$ . After we have counted for every matched pixel in the neighborhood of  $\mathbf{m}$ , we tally the votes. If the "good" votes are less than the "bad" votes,  $\mathbf{m}$  will be removed from the disparity map. This is conducted for every matched pixel in the disparity map; the result is a disparity map that conforms to the principle of disparity gradient limit.

## 5.3 Contour Matching

Template-matching assumes that corresponding images patches present some similarity. This assumption may be wrong at occluding boundaries, or object contours. Yet object contours are very important cues for view synthesis. The lack of matching information along object contours will result in excessive smearing or blurring in the synthesized views. So it is necessary to include a module that extracts and matches the contours across views in our system.

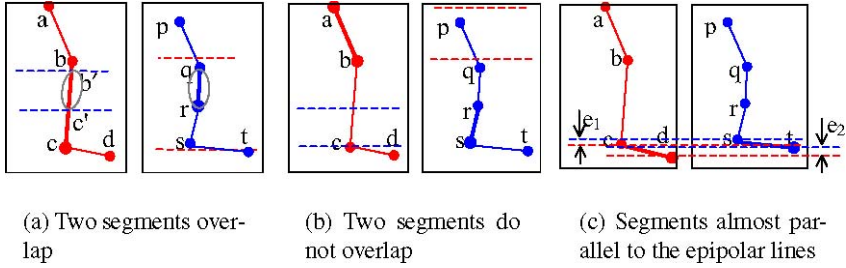
The contour of the foreground object can be extracted after background subtraction. It is approximated by polygonal lines using the Douglas-Poker algorithm[8]. The control points on the contour are further refined to sub-pixel accuracy using the "snake" technique[13]. Once we have two polygonal contours, denoted by  $P = \{\mathbf{v}_i | i = 1..n\}$  in the first image and  $P' = \{\mathbf{v}'_i | i = 1..m\}$  in the second image, we use the dynamic programming technique (DP) to find the global optimal match across them.

Since it is straightforward to formulate contour matching as a dynamic programming problem with states, stage, and decisions, we will only discuss in detail the design of the cost functions (The reader is referred to Bellman's book about dynamic programming techniques [3]). There are two cost functions, the *matching* cost and the *transition* cost. The matching cost function  $C(i, j)$  measures the "goodness" of matching between segment  $V_i = \overline{\mathbf{v}_i \mathbf{v}_{i+1}}$  in  $P$  and segment  $V'_j = \overline{\mathbf{v}'_j \mathbf{v}'_{j+1}}$  in  $P'$ . The lower the cost, the better the matching. The transition cost function  $W(i, j | i_0, j_0)$  measures the smoothness from segment  $V_{i_0}$  to segment  $V_i$ , assuming that  $(V_i, V'_j)$  and  $(V_{i_0}, V'_{j_0})$  are matched pairs of segments. Usually,  $V_i$  and  $V_{i_0}$  are continuous segments, i.e.,  $\|i_0 - i\| \leq 1$ . It penalize for matches that are out of order. The scoring scheme of DP, formulated as a forward recursion function, is then given by

$$\begin{aligned} M(i, j) = \min( & M(i-1, j-1) + C(i, j) + W(i, j | i-1, j-1), \\ & M(i, j-1) + C(i, j) + W(i, j | i, j-1), \\ & M(i-1, j) + C(i, j) + W(i, j | i-1, j) ) . \end{aligned}$$



**The Matching Cost.** It takes into account the epipolar constraint, the orientation difference, and the disparity gradient limit.



**Fig. 4.** Applying the epipolar constraint to contour matching.

- **The epipolar constraint:** We distinguish three configurations, as shown in Figure 4 where the red line is the contour in the first (upper) image; the blue line is the contour in the second (lower) image. The dotted lines are the corresponding epipolar lines. In Figure 4(a), segment  $bc$  and segment  $qr$  are being matched, and  $C_e = 0$ . The epipolar constraint limits  $qr$  corresponding to segment  $b'c'$ , instead of  $bc$ . In Figure 4(b), the epipolar constraint tells that segment  $ab$  cannot match segment  $rs$  because there is no overlap. In that case, a sufficiently large cost ( $T_{highcost}$ ) is assigned to this match. When the orientation of at least one line segment is very close to that of epipolar lines, intersection of the epipolar line with the line segment cannot be computed reliably. In that case, the cost is the average inter-epipolar distance ( $d_e = (e_1 + e_2)/2$ ), as illustrated in the figure. In summary, the epipolar constraint cost for a pair of segment  $(V_i, V'_j)$  is

$$C_e = \begin{cases} d_e & \text{if } V_i \text{ or } V'_j \text{ is close to horizontal lines;} \\ 0 & \text{if } V_i \text{ or } V'_j \text{ overlaps;} \\ T_{highcost} & \text{otherwise.} \end{cases} \quad (4)$$

- **The orientation difference:** It is defined as a power function of the orientation difference between the proposed matching segments  $(V_i, V'_j)$ . Let  $a_i$  and  $a_j$  be orientation of  $V_i$  and  $V'_j$ , the orientation difference is

$$C_a = \left( \frac{|a_i - a_j|}{T_a} \right)^n \quad (5)$$

where  $T_a$  is the angular difference threshold, and  $n$  is the power factor. We use  $T_a = 30^\circ$  and  $n = 2$ .

- **The disparity gradient limit:** It is similar to that used in template matching. However, we do not want to consider feature points in matching contour segments because the contour is on the occluding boundary, where the disparity gradient with respect to the matched feature points is very likely to exceed the limit. On the other hand, it is reasonable to assume that the disparity gradient limit will be upheld *between* the two endpoints of the segment. We adopt the disparity prediction model in [25]. That is, given a pair of matched points  $(\mathbf{m}_i, \mathbf{m}'_i)$ , the disparity of a point  $\mathbf{m}$  is modeled as

$$d = d_i + D_i n_i \quad (6)$$

where  $d_i = \|\mathbf{m}'_i - \mathbf{m}_i\|$ ,  $D_i = \|\mathbf{m} - \mathbf{m}_i\|$ , and  $n_i \sim N(0, \sigma^2 I)$  with  $\sigma = K/(2 - K)$ . A pair of matched segments contains two pairs of matched endpoints  $(\mathbf{m}_s, \mathbf{m}'_s)$  and  $(\mathbf{m}_e, \mathbf{m}'_e)$ . We use  $(\mathbf{m}_s, \mathbf{m}'_s)$  to predict the disparity of  $\mathbf{m}_e$ , and compute the variance of the “real” disparity from the predicted one. Similarly we also compute the variance of the predicted disparity of  $\mathbf{m}_s$  using  $(\mathbf{m}_e, \mathbf{m}'_e)$ . As suggested in [25], the predicted variance should be less restrictive when the point being considered is away from the matched point, which leads to the following formulae:

$$\sigma_i = [\sigma_{max} - \sigma_{min}](1 - \exp(-D_i^2/\tau^2)) + \sigma_{min} \quad (7)$$

where the range  $[\sigma_{min}, \sigma_{max}]$  and  $\tau$  are parameters. We use  $\sigma_{min} = 0.5$ ,  $\sigma_{max} = 1.0$ , and  $\tau = 30$ .

Now we can finally write out the disparity gradient cost. Let  $d_s = \|\mathbf{m}'_s - \mathbf{m}_s\|$ ,  $d_e = \|\mathbf{m}'_e - \mathbf{m}_e\|$ ,  $D_1 = \|\mathbf{m}_e - \mathbf{m}_s\|$ ,  $D_2 = \|\mathbf{m}'_e - \mathbf{m}'_s\|$ , and  $\Delta d = d_e - d_s$ ;  $\sigma_e$  and  $\sigma_s$  are computed by plugging in  $D_1$  and  $D_2$  into (7); the disparity gradient cost is given by

$$C_d = \Delta d^2/\sigma_e^2 + \Delta d^2/\sigma_s^2. \quad (8)$$

Combining all the above three terms, we have the final matching cost as:

$$C = \max(T_{highcost}, C_e + w_a C_a + w_d C_d) \quad (9)$$

where  $w_a$  and  $w_d$  are weighting constants. The match cost is capped by  $T_{highcost}$ . This is necessary to prevent any corrupted segment in the contour from contaminating the entire matching. In our implementation,  $T_{highcost} = 20$ ,  $w_a = 1.0$ , and  $w_d = 1.0$ .

**The Transition Cost.** In contour matching, when two segments are continuous in one image, we would prefer that their matched segments in the other image are continuous too. This is not always possible due to changes in visibility: some part of the contour can only be seen in one image. The transition cost ( $W$ ) is designed to favor smooth matching from one segment to the next, while taking into account discontinuities due to occlusions. The principle we use is again the gradient disparity limit. For two consecutive segments  $V_i$  and  $V_{i+1}$  in  $P$ , the transition cost function is the same as the one used in matching cost – equation (8), except that the two pairs of matched points involved are now the endpoint of  $V_i$  and the starting point of  $V_{i+1}$  and their corresponding points in  $P'$ .

## 6 View Synthesis

From the previous tracking and matching stages, we have obtained a set of point matches and line matches that can be used to synthesize new views. Note that these matches contain not only the modeled face part, but also other foreground part such as hands and shoulders. This is yet another advantage of our stereovision-based approach. We could obtain a more complete description of the scene geometry beyond the limit of the face model. Treating these matches as a whole, our view synthesis methods can create seamless virtual imagery. We implemented and tested two methods for view synthesis. One is based on view morphing [21] and the other uses hardware-assisted multi-texture blending. The view morphing technique allows to synthesize virtual views along the path connecting the optical centers of the two cameras. A view morphing factor  $c_m$  controls the exact view position. It is usually between 0 and 1, whereas a value of 0 corresponds exactly to the first camera view, and a value of 1 corresponds exactly to the second

camera view. Any value in between represents a virtual viewpoint somewhere along the path from the first camera to the second.

In our hardware-assisted rendering method, we first create a 2D triangular mesh using Delaunay triangulation in the first camera's image space. We then offset each vertex's coordinate by its disparity modulated by the view morphing factor  $c_m$ ,  $[u'_i, v'_i] = [u_i + c_m d_i, v_i]$ . The offset mesh is fed to the hardware render with two sets of texture coordinates, one for each camera image. Note that all the images and the mesh are in the rectified coordinate space. We need to set the viewing matrix to the inverse of the rectification matrix to "un-rectify" the resulting image to its normal view position. This is equivalent to the post-warp in view morphing. Thus the hardware can generate the final synthesized view in a single pass. We also use a more elaborate blending scheme, thanks to the powerful graphics hardware. The weight  $W_i$  for the vertex  $V_i$  is based on the product of the total area of adjacent triangles and the view-morphing factor, as

$$W_i = \frac{\sum S_i^1 * (1 - c_m)}{\sum S_i^1 * (1 - c_m) + \sum S_i^2 * c_m}; \quad (10)$$

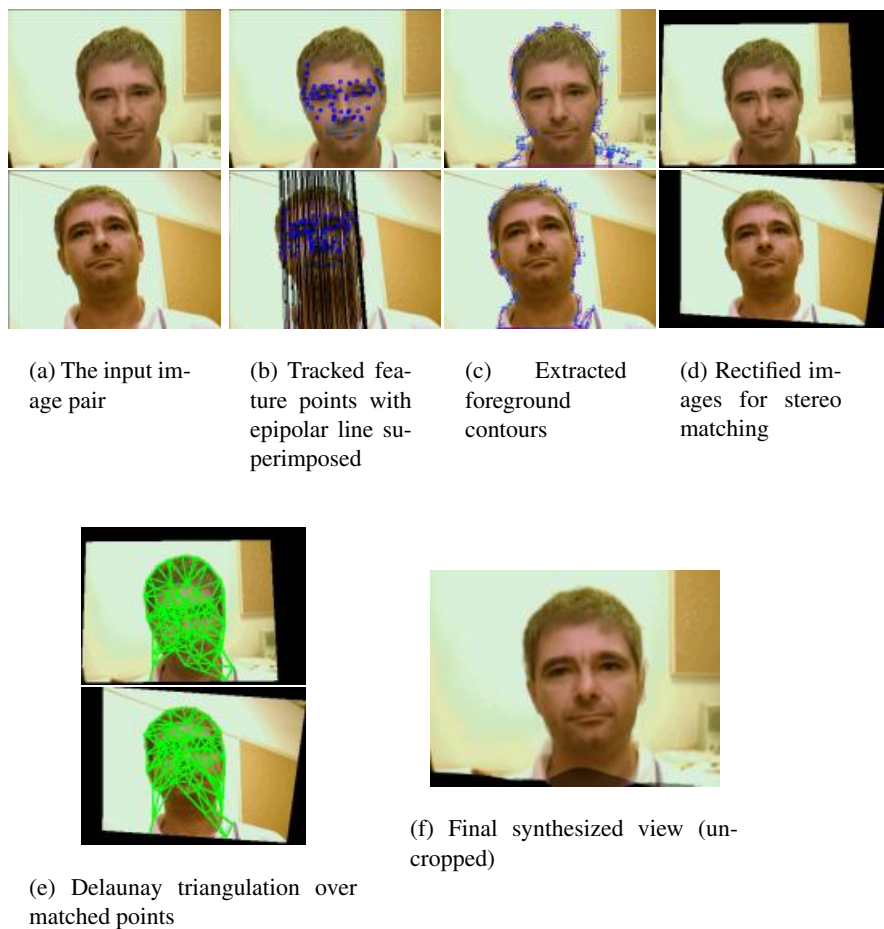
where  $S_i^1$  are the areas of the triangles of which  $V_i$  is a vertex, and  $S_i^2$  are the areas of the corresponding triangles in the other image. By changing the view morphing factor  $c_m$ , we can use the graphics hardware to synthesize correct views with desired eye gaze in real-time. Because the view synthesis process is conducted by hardware, we can spare the CPU for more challenging tracking and matching tasks.

Comparing these two methods, the hardware-assisted method, aside from its blazing speed, generates crisper results if there is no false match in the mesh. On the other hand, the original view morphing method is less susceptible to bad matches, because it essentially uses every matched point or line segment to compute the final coloring of a single pixel, while in the hardware-based method only the three closest neighbors are used.

Regarding the background, it is very difficult to obtain a reliable set of matches since the baseline between the two views is very large, as can be observed in the two original images shown in Fig. 1. In this work, we do not attempt to model the background at all, but we offer two solutions. The first is to treat the background as unstructured, and add image boundary as matches. The result will be ideal if the background has a uniform color; otherwise, it will be fuzzy as shown in the synthesized view shown in Fig. 1. The second solution is to replace the background by anything appropriate. In that case, view synthesis is only performed for the foreground objects. In our implementation, we overlay the synthesized foreground objects on the image from the first camera. The results shown in the following section were produced in this way.

## 7 Experiment Results

We have implemented our proposed approach using C++ and tested with several sets of real data. Very promising results have been obtained. We will first present a set of sample images to further illustrate our algorithm, then we will show some more results from different test users. For each user, we built a personalized face model using a face modelling tool[16]. This process, which takes only a few minutes and requires no additional hardware, only needs to be done once per user. All the parameters in our algorithm are set to be the same for all the tests.



**Fig. 5.** Intermediate results of our eye-gaze correction algorithm

Figure 5 shows the intermediate results at various stages of our algorithm. It starts with a pair of stereo images in Fig. 5(a); Fig. 5(b) shows the matched feature points, the epipolar lines of feature points in the first image are drawn in the second image. Fig. 5(c) shows the extracted foreground contours: the red one (typically a few pixels far away from the "true" contour) is the initial contour after background subtraction while the blue one is the refined contour using the "snake" technique. In Fig. 5(d) we show the rectified images for template matching. All the matched points form a mesh using Delaunay triangulation, as shown in Fig. 5(e). The last image (Fig. 5(f)) shows the synthesized virtual view. We can observe that the person appears to look down and up in the two original image but look forward in this synthesized view.

During our experiments, we captured all test sequences with resolution 320x240 at 30 frames per second. Our current implementation can only run at 4 to 5 frames per second. The results shown here are computed with our system in a "step-through" mode.

Except the manual initialization performed once at the beginning of the test sequences, the results are computed automatically without any human interaction.



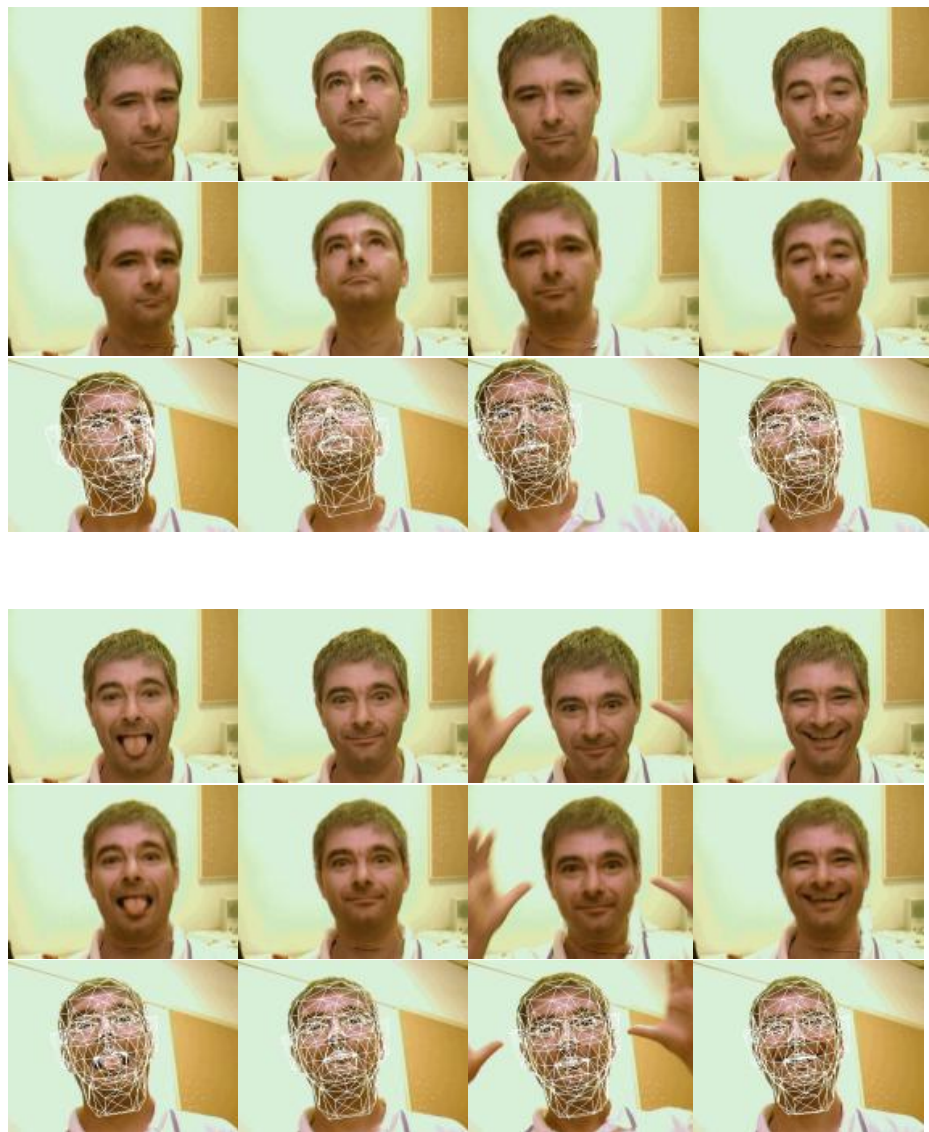
**Fig. 6.** Sample results from the first test sequence. The top and bottom rows show the images from the top and bottom cameras. The middle row displays the synthesized images from a virtual camera located in the middle of the real cameras. The frame numbers from left to right are 108, 144, 167 and 191.

The first sequence (Figure 6) shows the viability of our system. Note the large disparity changes between the upper and lower camera images, making direct template-based stereo matching very difficult. However, our model-based system is able to accurately track and synthesize photo-realistic images under the difficult configuration, even with partial occlusions or oblique viewing angles.

Our second sequence is even more challenging, containing not only large head motions, but also dramatic facial expression changes and even hand waving. Results from this sequence, shown in Figure 7, demonstrated that our system is both effective and robust under these difficult conditions. Non-rigid facial deformations, as well as the subject's torso and hands, are not in the face model, yet we are still able to generate seamless and convincing views, thanks to our view matching algorithm that includes a multitude of stereo matching primitives (features, templates, and curves). Templates matching finds matching points, as many as possible, in regions where the face model does not cover, while contour matching preserves the important visual cue of silhouettes.

## 8 Conclusions

In this paper, we have presented a software scheme for maintaining eye contact during video-teleconferencing. We use model-based stereo tracking and stereo analysis to compute a partial 3D description of the scene. Virtual views that preserve eye contact are then synthesized using graphics hardware. In our system, modeled-based head tracking and



**Fig. 7.** Sample results from the second test sequence. The upper and lower rows are the original stereo images, while the middle rows are the synthesized ones. The triangular face model is overlaid on the bottom images. From left to right and from top to bottom, the frame numbers are 159, 200, 400, 577, 617, 720, 743, and 830.

stereo analysis work hand in hand to provide a new level of accuracy, robustness, and versatility that neither of them alone could provide. Experimental results have demonstrated the viability and effectiveness of our proposed approach.

While we believe that our proposed eye-gaze correction scheme represents a large step towards a viable video-teleconferencing system for the mass market, there are still plenty of rooms for improvements, especially in the stereo view matching stage. We have used several matching techniques and prior domain knowledge to find good matches as many as possible, but we have not exhausted all the possibilities. We believe that the silhouettes in the virtual view could be more clear and consistent across frames if we incorporate temporal information for contour matching. Furthermore, there are still salient curve features, such as hairlines and necklines, that sometimes go unmatched. They are very difficult to match using a correlation-based scheme because of highlights and visibility changes. We are investigating a more advanced curve matching technique.

## References

1. A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive Estimation of Structure and Motion Using the Relative Orientation Constraint. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 70–75, 1993.
2. S. Basu, I. Essa, and A. Pentland. Motion Regularization for Model-based Head Tracking. In *Proceedings of International Conference on Pattern Recognition*, pages 611–616, Vienna, Austria, 1996.
3. R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.
4. M. J. Black and Y. Yacoob. Tracking and Recognizing Rigid and Non-Rigid Facial Motions Using Local Parametric Model of Image Motion. In *Proceedings of International Conference on Computer Vision*, pages 374–381, Cambridge, MA, 1995.
5. P. Burt and B. Julesz. A gradient limit for binocular fusion. *Science*, 208:615–617, 1980.
6. T.J. Cham and M. Jones. Gaze Correction for Video Conferencing. Compaq Cambridge Research Laboratory, <http://www.crl.research.digital.com/vision/interfaces/corga>.
7. D. DeCarlo and D. Metaxas. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2001.
8. D.H. Douglas and T.K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *Canadian Cartographer*, 10(2):112–122, 1973.
9. O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
10. J. Gemmell, C.L. Zitnick, T. Kang, K. Toyama, and S. Seitz. Gaze-awareness for Videoconferencing: A Software Approach. *IEEE Multimedia*, 7(4):26–35, October 2000.
11. T. Horprasert. Computing 3-D Head Orientation from a Monocular Image. In *International Conference of Automatic Face and Gesture Recognition*, pages 242–247, 1996.
12. Michael Jones. Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes. *International Journal of Computer Vision*, 29(2):107–131, August 1998.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snake: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
14. R. Kollarits, C. Woodworth, J. Ribera, and R. Gitlin. An Eye-Contact Camera/Display System for Videophone Applications Using a Conventional Direct-View LCD. *SID Digest*, 1995.

15. J. Liu, I. Beldie, and M. Wopking. A Computational Approach to Establish Eye-contact in Videocommunication. In *the International Workshop on Stereoscopic and Three Dimensional Imaging (IWS3DI)*, pages 229–234, Santorini, Greece, 1995.
16. Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen. Rapid Modeling of Animated Faces From Video. *Journal of Visualization and Compute Animation*, 12(4):227–240, 2001.
17. C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 125–131, June 1999.
18. L. Mhlbach, B. Kellner, A. Prussog, and G. Romahn. The Importance of Eye Contact in a Videotelephone Service. In *11th Interational Symposium on Human Factors in Telecommunications*, Cesson Sevigne, France, 1985.
19. M. Ott, J. Lewis, and I. Cox. Teleconferencing Eye Contact Using a Virtual Camera. In *INTERCHI '93*, pages 119 – 110, 1993.
20. S. Pollard, J. Porrill, J. Mayhew, and J. Frisby. Disparity Gradient, Lipschitz Continuity, and Computing Binocular Correspondance. In O.D. Faugeras and G. Giralt, editors, *Robotics Research: The Third International Symposium*, volume 30, pages 19–26. MIT Press, 1986.
21. S.M. Seitz and C.R. Dyer. View Morphing. In *SIGGRAPH 96 Conference Proceedings*, volume 30 of *Annual Conference Series*, pages 21–30, New Orleans, Louisiana, 1996. ACM SIGGRAPH, Addison Wesley.
22. J. Shi and C. Tomasi. Good Features to Track. In *the IEEE Conferecne on Computer Vision and Pattern Recognition*, pages 593–600, Washington, June 1994.
23. R.R. Stokes. Human Factors and Appearance Design Considerations of the Mod II PICTUREPHONE Station Set. *IEEE Trans. on Communication Technology*, COM-17(2), April 1969.
24. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
25. Z. Zhang and Y. Shan. A Progressive Scheme for Stereo Matching. In M. Pollefeys et al., editor, *Springer LNCS 2018: 3D Structure from Images - SMILE 2000*, pages 68–85. Springer-Verlag, 2001.



# Wavelet-Based Correlation for Stereopsis

Maureen Clerc

CERMICS, INRIA  
BP 92, 06902 Sophia-Antipolis  
France

**Abstract.** Position disparity between two stereoscopic images, combined with camera calibration information, allow depth recovery. The measurement of position disparity is known to be ambiguous when the scene reflectance displays repetitive patterns. This problem is reduced if one analyzes scale disparity, as in shape from texture, which relies on the deformations of repetitive patterns to recover scene geometry from a single view.

These observations lead us to introduce a new correlation measure based not only on position disparity, but on position and scale disparity. Local scale disparity is expressed as a change in the scale of wavelet coefficients. Our work is related to the spatial frequency disparity analysis of Jones and Malik (ECCV92). We introduce a new wavelet-based correlation measure, and we show its application to stereopsis. We demonstrate its ability to reproduce perceptual results which no other method of our knowledge had accounted for.

## Introduction

Position disparity is a well-known cue used for stereoscopic depth reconstruction, but it is not the only one. Jones and Malik have demonstrated the importance of frequency disparity information in stereopsis [4,5], and perceptual results have shown that stereo image pairs created with bandpass filtered noise, and designed to contain frequency disparity information, but no position disparity information, lead to slant perception [12]. In shape from texture, which studies the recovery of scene depth from the monocular image of a homogeneously textured object, the shape cue comes from the scale or the frequency disparity between different positions in the image [3,6].

Many stereoscopic algorithms which produce dense disparity maps require a measure of similarity between regions from the image pair, and define disparity as the local translation maximizing this similarity. In order not to be sensitive to noise, the similarity measure is usually based on averages of image intensities over a region. A geometrical problem arises when the local scene element being viewed is not fronto-parallel: because of projective distortion, the shapes and sizes of corresponding image patches are not identical in the two images. Since the distortion parameters depend on the local orientation of the scene whose shape is being computed, an iterative scheme must be used, in order to improve the similarity measure by using the scene shape calculated at the previous step [2].

Repetitive texture (regular grating, woven fabric), is difficult to handle with classical similarity measures because they produce many local maxima. We introduce a new similarity measure, incorporating a scale disparity constraint, which we observe in practice to lift the repetitive texture ambiguity.

Wavelets, whose parameters are position and scale, are ideally suited to define a correlation measure based on position and scale disparity. Wavelet methods have been proposed for stereopsis in the context of coarse-to-fine disparity measurement [9,10]. The method proposed here is related to [8], in which an affine transformation between image patches is measured at the output of a set of filters.

We demonstrate through numerical examples in 1D and 2D that our wavelet-based correlation is smooth and well-behaved and generally displays only one local maximum in the presence of repetitive texture. This makes it a good ingredient for stereopsis algorithms which rely on a similarity measure [13].

## 1 Wavelet Analysis of Distortion

In this paper, we neglect occlusion effects, which can be taken into account at a higher level, for instance in a cooperative stereopsis algorithm [13]. We suppose the stereoscopic pair of images  $I_l$  and  $I_r$  to satisfy

$$I_l(x) = I_r(d(x)) .$$

where

$$d(x) = d(x_1, x_2) = (d_1(x_1, x_2), d_2(x_1, x_2))$$

is a continuous map, which we call the disparity<sup>1</sup> between  $I_l$  and  $I_r$ . Let  $\psi(x)$  be a wavelet, i.e. a two-dimensional oscillating function whose spatial support is localized around  $(0, 0)$  and whose spatial frequency support concentrates around a frequency  $\xi \neq (0, 0)$ . A Gabor wavelet, which is a Gaussian window modulated to oscillate at a frequency  $\xi$ , satisfies these requirements. Let  $u$  be a position in  $\mathbb{R}^2$  and let  $S$  denote a positive definite  $2 \times 2$  matrix. An affine transformation

$$\psi_{u,S}(x) = (\det S)^{-1} \psi(S^{-1}(x - u)) .$$

modifies the space-scale localization of the wavelet to position  $u$  and frequency  $S^T \xi$ , where  $S^T$  is the matrix transpose of  $S$ .

The wavelet coefficients of  $I$  are defined by

$$W(u, S) = \langle I, \psi_{u,S} \rangle = \int I(x) \psi_{u,S}^*(x) dx . \quad (1)$$

The squared amplitude of wavelet coefficients  $|W(u, S)|^2$  measures the energy contained in a surface patch of the image  $I$  centered at  $u$ , around spatial frequency  $S^T \xi$ .

<sup>1</sup> Disparity is generally defined as  $d(x) - x$ .

Let us compare the wavelet coefficients of  $I_l$  (denoted  $W_l(u, S)$ ) to those of  $I_r$  (denoted  $W_r(u, S)$ ), supposing for the moment that the disparity  $d$  between the two images is an affine transformation

$$d(x) = d(u) + J \times (x - u) ,$$

where  $J$  is a  $2 \times 2$  matrix. A simple change of variable<sup>2</sup> in (1) yields

$$W_l(u, S) = W_r(d(u), J S) .$$

This relationship between wavelet coefficients makes apparent the position and scale disparities between the two images.

In the case of a general, no longer affine, disparity, if  $d$  is differentiable, it can be approximated, locally around a position  $u$ , by its first-order Taylor approximation

$$d(x) \approx d(u) + J(u) \times (x - u) ,$$

where  $J(u)$  is the  $2 \times 2$  Jacobian matrix of  $d$  at position  $u$

$$J(u) = \begin{pmatrix} \frac{\partial d_1}{\partial u_1}(u) & \frac{\partial d_1}{\partial u_2}(u) \\ \frac{\partial d_2}{\partial u_1}(u) & \frac{\partial d_2}{\partial u_2}(u) \end{pmatrix} .$$

If  $I_l$  were a smooth image, it would be possible to extend the Taylor approximation to the image, and obtain

$$I_l(d(x)) \approx I_l(d(u) + J(u) \times (x - u))$$

Unfortunately, even in the absence of sharp discontinuities, an image cannot be assumed to be smooth on account of measurement noise. One can however model the image as the realization of a stochastic process, whose covariance is smooth away from the diagonal. Then one can show [1] that the variances of wavelet coefficients satisfy

$$E[|W_l(u, S)|^2] \approx E[|W_r(d(u), J(u) S)|^2] . \quad (2)$$

## 2 Correlation Measure

We choose a correlation measure between images  $I_l$  and  $I_r$  of the form

$$\rho = \frac{2 (\text{Feat}_l, \text{Feat}_r)}{(\text{Feat}_l, \text{Feat}_l) + (\text{Feat}_r, \text{Feat}_r)} \quad (3)$$

---

<sup>2</sup> The change of variable is the motivation for the  $L^1$  normalization of the wavelet, instead of the more classical  $L^2$  normalization.

where  $\text{Feat}_l$  and  $\text{Feat}_r$  are features relative to the two images, and  $(\cdot, \cdot)$  is an inner product in feature parameter space. Clearly,  $\rho \leq 1$  and  $\rho = 1$  if and only if  $\text{Feat}_l = \text{Feat}_r$ . Section 4.2 comments on this choice of correlation ratio.

Relationship (2) allows to derive a correlation measure which combines position disparity  $d$  and scale disparity  $J$ . Given a collection of scaling matrices  $S_i$  (typically less than 5), let

$$\text{Feat}_l(u, S_i) = E [|W_l(u, S_i)|^2]$$

and for a given position disparity  $d$  and scale disparity  $J$ , let

$$\text{Feat}_r^{(d,J)}(u, S_i) = E [|W_r(d(u), J(u) S_i)|^2] \quad .$$

We define the features at a position  $u$  as the collection of wavelet coefficient variances at preselected scales:

$$\text{Feat}_l(u) = \{\text{Feat}_l(u, S_i)\}_{\{S_i\}} \quad (4)$$

$$\text{Feat}_r^{(d,J)}(u) = \{\text{Feat}_r^{(d,J)}(u, S_i)\}_{\{S_i\}} \quad . \quad (5)$$

The inner product  $(\cdot, \cdot)$  is then simply

$$(\text{Feat}_l, \text{Feat}_r) = \sum_{S_i} \text{Feat}_l(u, S_i) \cdot \text{Feat}_r(u, S_i) \quad .$$

We finally obtain a correlation measure which depends on position disparity  $d$  and scale disparity  $J$ :

$$\rho(u, d, J) = \frac{2 \sum_{S_i} \text{Feat}_l(u, S_i) \cdot \text{Feat}_r^{(d,J)}(u, S_i)}{\sum_{S_i} (\text{Feat}_l(u, S_i))^2 + (\text{Feat}_r^{(d,J)}(u, S_i))^2} \quad . \quad (6)$$

Note that one is in practice limited to a unique realization of the images. In order to estimate the variance of wavelet coefficients, we rely on estimation results from [1]. We estimate  $E[|W(u, S)|^2]$  by averaging  $|W(v, S)|^2$  for  $v$  in a neighborhood  $B(u)$  of  $u$ . This estimation procedure, which is proved to be consistent, justifies the use of wavelet filters rather than Gaussian filters in the correlation measure.

For a given scaling matrix  $S_i$ ,  $\text{Feat}_l(u, S_i)$  is estimated with

$$\widehat{\text{Feat}}_l(u, S_i) = \int_{B(u)} |W_l(v, S_i)|^2 dv$$

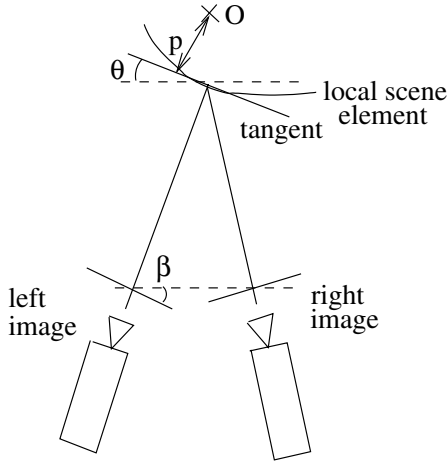
and the corresponding  $\text{Feat}_r^{(d,J)}(u, S_i)$  is estimated with

$$\widehat{\text{Feat}}_r^{(d,J)}(u, S_i) = \int_{B(u)} |W_r(d(v), J(v) S_i)|^2 dv \quad .$$

### 3 Local Shape Measurement

We have relaxed the classical stereopsis problem by introducing a new parameter for image matching: scale disparity  $J$ . At first view, the problem may appear more difficult to solve, because finding the best local match between  $I_l$  and  $I_r$  now requires to maximize  $\rho$  (defined in (6)) over  $d$  and  $J$ , instead of  $d$  only.

This difficulty disappears when recalling the objective, which is to measure scene depth. At a given position  $u$  in image  $I_l$ , position disparity  $d(u)$  and scale disparity  $J(u)$  are both a function of the local position and orientation of the scene element being viewed. Consider a simplified 2D geometry displayed in Figure 3, where local orientation is defined by a unique angle  $\theta$ , and position is determined by the signed distance  $p$  from the surface tangent to a reference point  $O$ . Appendix A gives the expressions of position disparity  $d_{p,\theta}$  and scale disparity  $J_{p,\theta}$ . In 3D, local geometry would be expressed by three scalar parameters  $(p, \theta_1, \theta_2)$ .



**Fig. 1.** Simplified 2D geometry.

The correlation measure  $\rho$  can then be reformulated as a function of  $p$  and  $\theta$  instead of  $d$  and  $J$ :

$$\rho(u, p, \theta) = \frac{2 \sum_{S_i} \text{Feat}_l(u, S_i) \cdot \text{Feat}_r^{(d_{p,\theta}, J_{p,\theta})}(u, S_i)}{\sum_{S_i} (\text{Feat}_l(u, S_i))^2 + (\text{Feat}_r^{(d_{p,\theta}, J_{p,\theta})}(u, S_i))^2} . \quad (7)$$

## 4 Numerical Results

In this section, we compare the correlation measure  $\rho(u, p, \theta)$  defined in (7) to the classical area correlation  $\rho_0(u, p, \theta)$  defined by

$$\rho_0(u, p, \theta) = \frac{\int_{B(u)} \tilde{I}_l(v) \tilde{I}_r(d_{p,\theta}(v)) dv}{\left( \int_{B(u)} \tilde{I}_l(v)^2 dv \int_{B(u)} \tilde{I}_r(d_{p,\theta}(v))^2 dv \right)^{1/2}} ,$$

where

$$\tilde{I}_l(v) = I_l(v) - \frac{1}{\text{area}(B(u))} \int_{B(u)} I_l(v) dv$$

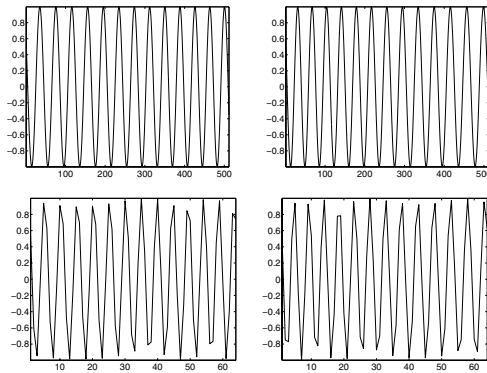
and

$$\tilde{I}_r(d_{p,\theta}(v)) = I_r(d_{p,\theta}(v)) - \frac{1}{\text{area}(B(u))} \int_{B(u)} I_r(d_{p,\theta}(v)) dv .$$

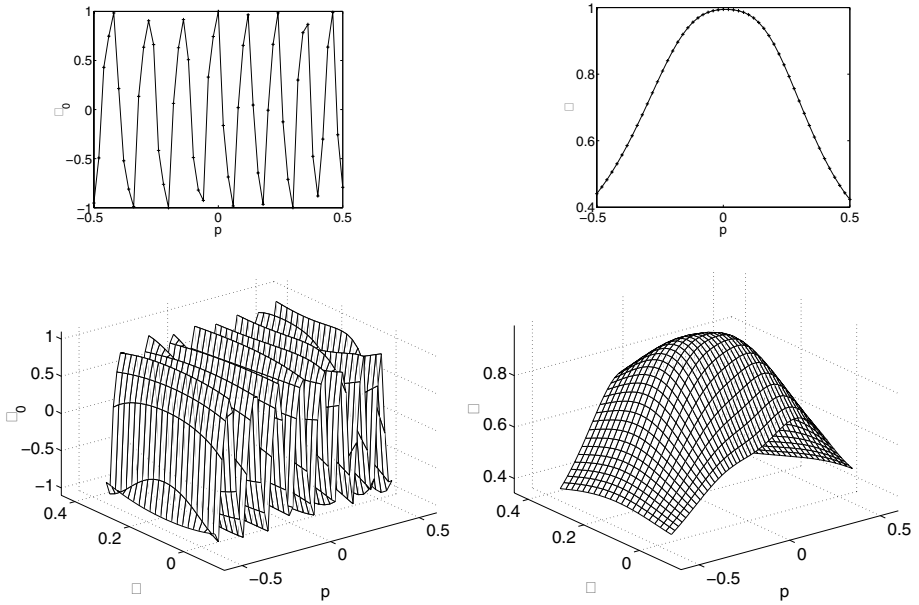
### 4.1 One-Dimensional Results

Consider the following synthetic example: a one-dimensional signal covering a straight line with position and orientation parameters  $p_0$  and  $\theta_0$  is viewed by two cameras, leading to two distorted one-dimensional signals  $I_l$  and  $I_r$  (Figure 2). The local scene geometry is specified by  $p_0 = 0$  and  $\theta_0 = .15$  radians. The width of the portion of image being viewed is .69 units in the real scene. The widths of the perspective projections are .15 units in the left image, and .14 units in the right image.

The following three examples compare  $\rho$  and  $\rho_0$  for repetitive texture, for the extreme case with only scale disparity, and for non-repetitive texture.



**Fig. 2.** Two stereo “images”  $I_l$  (left) and  $I_r$  (right), high resolution (top), and low resolution (bottom).



**Fig. 3.** Correlation for the high-resolution stereo pair of Figure 2 (Example 1, modality 1). Left: classical correlation measure  $\rho_0(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho_0(u_0, p, \theta)$  (bottom). Right: new correlation measure  $\rho(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho(u_0, p, \theta)$  (bottom). The new correlation  $\rho$  is smoother than  $\rho_0$  and displays a unique local maximum at the correct value  $(p, \theta) = (p_0, \theta_0)$ .

**Example 1: repetitive texture.** A 1D repetitive texture stereo pair is displayed in Figure 2 both with a high resolution (512 pixels) and with a low resolution (64 pixels).

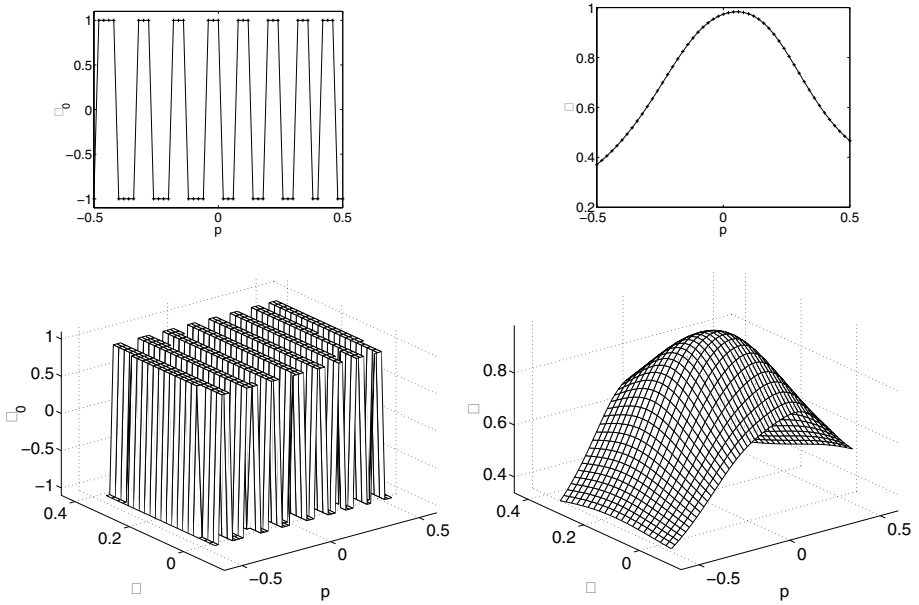
We display  $\rho$  and  $\rho_0$  measured at a fixed position  $u_0$  in the middle of the left image for  $-0.05 \leq \theta \leq 0.35$  and  $-0.5 \leq p \leq 0.5$  in three different modalities:

1. High-resolution images  $I_l$  and  $I_r$  with 512 pixels (Figure 3);
2. Low-resolution images  $I_l$  and  $I_r$  with 64 pixels (Figure 4);
3. High-resolution images  $I_l$  and  $I_r$  corrupted by two distinct realizations of an additive white Gaussian noise with variance equal to  $1/15$  of the standard deviation of  $I_l$  (Figure 6).

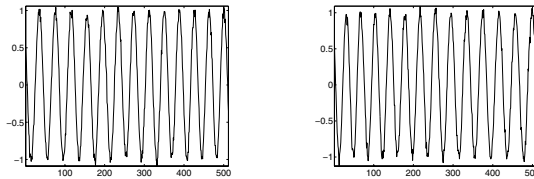
The wavelet used is the one-dimensional Gabor wavelet

$$\psi(x) = \exp(-x^2) \exp(-i\xi x) .$$

We select 5 scales  $s_i = 0.05 \times (1.1)^i$  for  $i = 0, \dots, 4$ , which are relatively coarse compared to the width of image  $I_l$  which is .15. The width of the averaging window  $B(u_0)$  is .015.



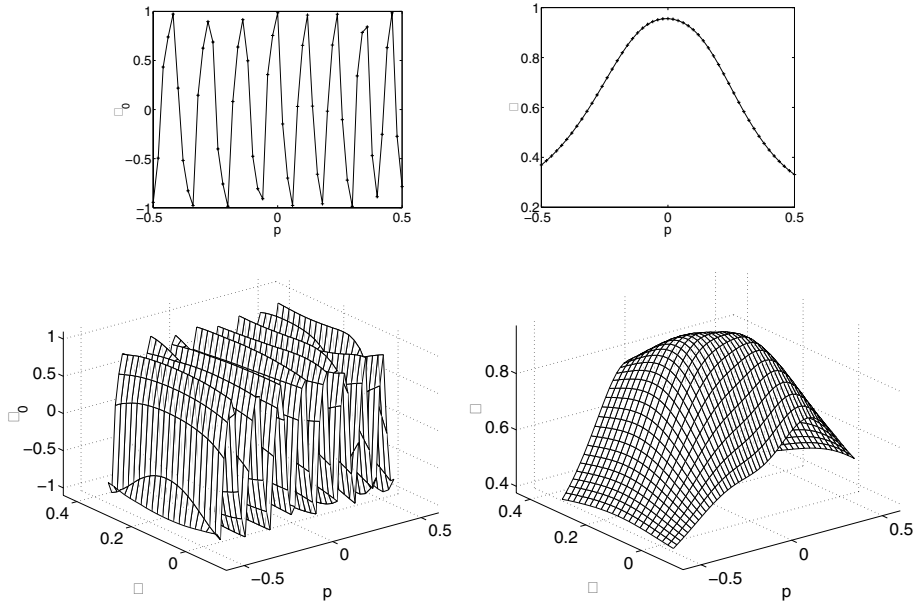
**Fig. 4.** Correlation for the low-resolution stereo pair of Figure 2 (Example 1, modality 2). Left: classical correlation measure  $\rho_0(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho_0(u_0, p, \theta)$  (bottom). Right: new correlation measure  $\rho(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho(u_0, p, \theta)$  (bottom). The classical correlation  $\rho_0$  is blocky because of the low resolution, whereas  $\rho$  is smooth, with a unique local maximum (which is not as precise as in the first modality because of the low resolution).



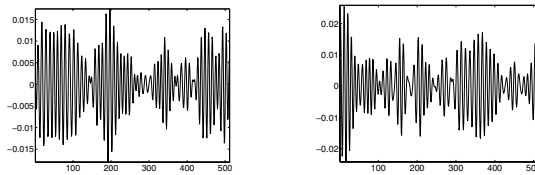
**Fig. 5.** Stereo pair  $I_l$  (left) and  $I_r$  (right) corrupted by additive Gaussian noise.

**Example 2: scale disparity only.** We demonstrate that our correlation measure accounts for a stereoscopic depth perception experiment [12], in which there is a spatial frequency disparity between images  $I_l$  and  $I_r$ , but no position disparity. Images  $I_l$  and  $I_r$  in Figure 7 are created by projecting two independent realizations of a colored noise according to a stereoscopic geometry with parameters  $(p_0, \theta_0) = (0, .15)$ . The numerical results in Figure 8 show that the classical correlation measure  $\rho_0$ , based on position disparity alone, has many lo-





**Fig. 6.** Correlation for the noisy stereo pair of Figure 5 (Example 1, modality 3). Left: classical correlation measure  $\rho_0(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho_0(u_0, p, \theta)$  (bottom). Right: new correlation measure  $\rho(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho(u_0, p, \theta)$  (bottom). Observe the smoothness of  $\rho$  compared to  $\rho_0$ .

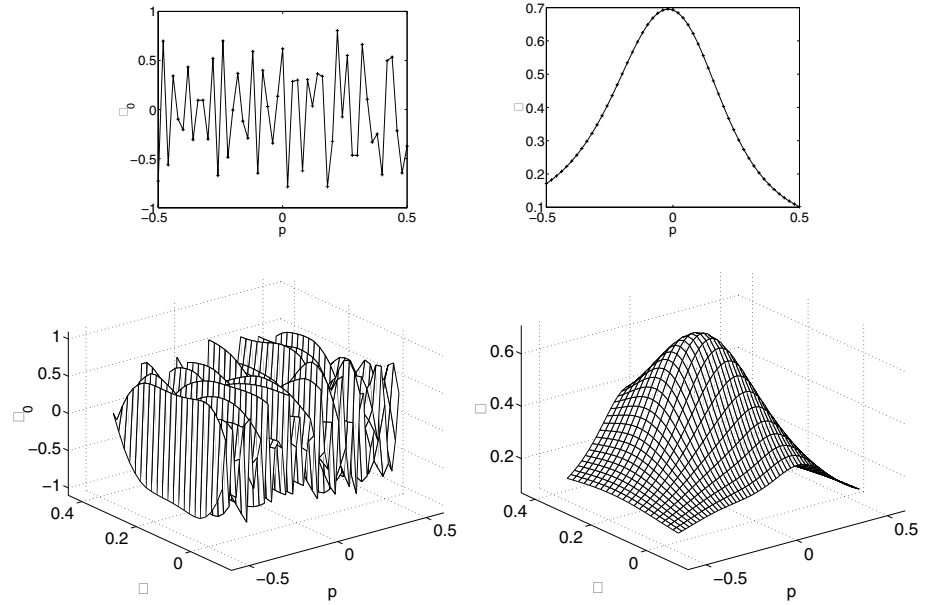


**Fig. 7.** Two stereo “images”  $I_l$  (left) and  $I_r$  (right) containing scale disparity information, but no position disparity information.

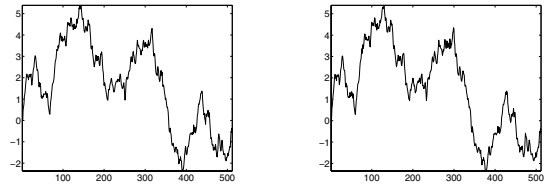
cal maxima, whereas the correlation measure  $\rho$  which is based on scale disparity information displays a unique local maximum at the correct position  $p_0$ .

The wavelets and scales used in this example are the same as in Example 1.

**Example 3: non-repetitive texture.** Finally, we show the advantage of the new correlation measure  $\rho$  over the classical one  $\rho_0$  in the case of a stereo pair with no repetitive texture, but high-frequency oscillations, displayed in Figure 9. Because of the band-pass filtering performed by the Gabor wavelets, the correlation measure  $\rho$  is smoother than the classical one, as displayed in Figure



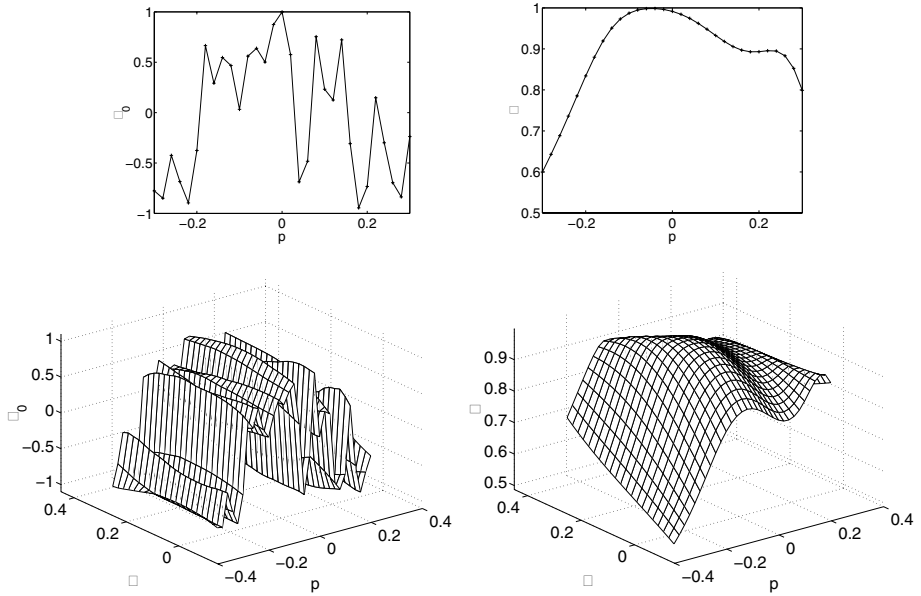
**Fig. 8.** Left: with only scale disparity information (stereo pair of Figure 7), the classical correlation measure  $\rho_0$  fails. Right: on the other hand, the new correlation measure  $\rho$  displays a unique local maximum at  $p = p_0$  and  $\theta = \theta_0$ .



**Fig. 9.** A stereo pair  $I_l$  (left),  $I_r$  (right) with non-repetitive texture.

10. Note that this improvement could also have been obtained by applying a low-pass filter to the stereo image pair, before computing the classical correlation. However, this example is noteworthy because it shows the consistency of the wavelet-based correlation measure, which can be applied successfully to different types of images.

In this example we used a set of finer scales than in the two previous examples:  $s_i = 0.015 \times (1.1)^i$ , for  $i = 0, \dots, 4$ .



**Fig. 10.** Correlation for the non-repetitive stereo pair of Figure 9. Left: classical correlation measure  $\rho_0(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho_0(u_0, p, \theta)$  (bottom). Right: new correlation measure  $\rho(u_0, p, \theta_0)$  for fixed  $\theta = \theta_0$  (top) and  $\rho(u_0, p, \theta)$  (bottom). Right:  $\rho(u_0, p, \theta_0)$ . Even in the absence of repetitive texture,  $\rho$  is smoother than  $\rho_0$  and does not display spurious local maxima.

## 4.2 Choice of Correlation Ratio

We found the correlation measure (3) to be more discriminant than the widespread correlation measure

$$\frac{(\text{Feat}_l, \text{Feat}_r)}{(\text{Feat}_l, \text{Feat}_l)^{1/2} (\text{Feat}_r, \text{Feat}_r)^{1/2}}. \quad (8)$$

The above ratio (8) is equal to one as soon as  $\text{Feat}_l$  and  $\text{Feat}_r$  are *collinear* in feature space, whereas (3) is not equal to one unless  $\text{Feat}_l$  and  $\text{Feat}_r$  are *equal* in feature space. The possible advantage of (8) over (3) could be its immunity to shading variations between the images, but we observed that shading variations between images bring the maximum value of (3) down from one, without reducing its high contrast.

## 4.3 Two-Dimensional Results

Two-dimensional stereoscopic pairs created synthetically by projecting a planar image at position  $p_0$  and with orientation  $\theta_0$  onto two cameras in the simplified

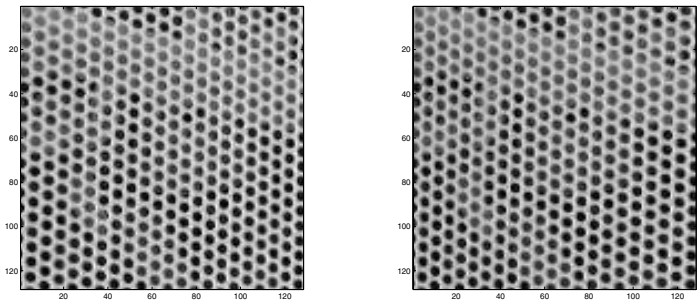
geometry of Figure 3. We compare the classical correlation measure  $\rho_0$  to the wavelet-based correlation measure  $\rho$ , for repetitive texture (the metallic panel in Figure 11), and non-repetitive texture (the dog hair in Figure 13).

The correlation measure  $\rho$  is computed using two-dimensional Gabor wavelets

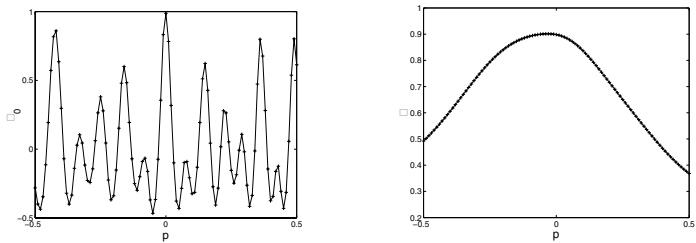
$$\psi_{u,s}(x) = \det S^{-1} \exp \left( -\|S^{-1}(x - u)\|^2 \right) \exp \left( -i\xi \cdot (S^{-1}(x - u)) \right) .$$

The selected scaling matrices are of the form

$$S_i = (1.05)^i \times \begin{pmatrix} 0.05 & 0 \\ 0 & 0.05 \end{pmatrix} \quad \text{for } i = 0, \dots, 4.$$

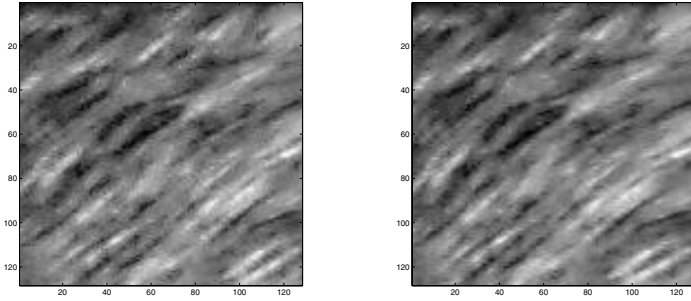


**Fig. 11.** A synthetic stereo pair, created from a photograph of a metallic panel.

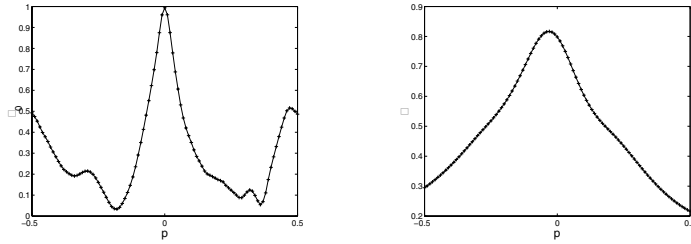


**Fig. 12.** Left:  $\rho_0(u_0, p, \theta_0)$  computed with the stereo pair of Figure 11. Right:  $\rho(u_0, p, \theta_0)$ . The correct value for  $p$  is  $p_0 = 0$ .

The numerical results displayed in Figures 12 and 14 for fixed  $\theta = \theta_0$  show that the new correlation measure is smoother than the classical area-based correlation, and displays a unique local maximum at the correct position  $p = p_0$ , for repetitive texture and non-repetitive texture alike.



**Fig. 13.** Synthetic stereo pair, created from a photograph of dog hair.



**Fig. 14.** Left:  $\rho_0(u_0, p, \theta_0)$  computed with the stereo pair of Figure 13. Right:  $\rho(u_0, p, \theta_0)$ . The correct value for  $p$  is  $p_0 = 0$ .

## Conclusion

We have introduced a new correlation measure for stereopsis based on wavelet coefficients of images, which has several interesting properties: it lifts the ambiguity on disparity measurement due to the presence of repetitive texture, it shows good performance at low resolution, in the presence of noise, as well as for non-repetitive texture with high-frequency components. Moreover, it is able to reproduce results on stereoscopic depth perception from scale disparity, in the absence of any position disparity. These promising results indicate that, if incorporated in a stereoscopy algorithm which deals with occlusions, this correlation measure could significantly improve its performance. Further work includes testing the algorithm on real data, and investigating the influence of surface curvature.

## A Disparity as a Function of Local Surface Position and Orientation

We give the expressions of  $d(u)$  and  $J(u)$  as a function of  $p$  and  $\theta$  in the case of a locally planar surface element. Let the origin of the Cartesian coordinates

be at the midpoint between the two optical centers, and let the line joining the two optical centers define the  $x$ -axis. We assume both cameras to have the same focal length, and we denote by  $c$  the half-length between the two optical centers. We suppose that the viewing angles of the cameras with respect to the  $x$ -axis are  $\beta$  and  $-\beta$ . In the simplified geometry of Figure 3, the position and orientation of a local scene element are described by

- $p$ , the distance between the surface tangent and a reference point  $O$  with coordinates  $(x_O, y_O)$ ,
- $\theta$ , the angle between the surface tangent and the  $x$ -axis.

If the scene element is locally flat (i.e. neglecting its curvature), the distortion map  $d$  is a homography

$$d(u) = \frac{A + B u}{C + D u} \quad (9)$$

where

$$\begin{aligned} A &= -\cos(\theta - \beta) X' + \cos(\theta + \beta) X \\ B &= -\sin(\theta - \beta) X' - \cos(\theta + \beta) Y' \\ C &= -\sin(\theta + \beta) X - \cos(\theta - \beta) Y' \\ D &= -\sin(\theta - \beta) Y' + \sin(\theta + \beta) Y \end{aligned}$$

with

$$\begin{aligned} X &= \cos \beta (c + x_O - p \sin \theta) - \sin \beta (y_O - p \cos \theta) \\ Y &= \sin \beta (c + x_O - p \sin \theta) + \cos \beta (y_O - p \cos \theta) \end{aligned}$$

and  $X', Y'$  are obtained by replacing  $c$  by  $-c$  and  $\beta$  by  $-\beta$  in the expressions of  $X, Y$  above. The Jacobian  $J(u)$  is calculated by differentiating (9) with respect to  $u$ :

$$J(u) = \frac{B C - A D}{(C + D u)^2} . \quad (10)$$

## References

1. Clerc, M. and Mallat, S. (2000) Estimating Deformations of Stationary Processes. Research Report no. 192, CERMICS, ENPC.
2. Devernay, F. and Faugeras, O. (1994) Computing Differential Properties of 3D Shapes from Stereoscopic Images without 3D Models. Research Report No. 2304, INRIA, July 1994.
3. Gårding, J. (1992). Shape from Texture for Smooth Surfaces under Perspective Projection. *Journal of Mathematical Imaging and Vision* **2**, pp. 327-350.
4. Jones, D.G. and Malik, J. (1992). A Computational Framework for Determining Stereo Correspondence from a Set of Linear Spatial Filters. Proc. ECCV'92, pp. 395-410.
5. Jones, D.G. and Malik, J. (1992). Determining Three-Dimensional Shape from Orientation and Spatial Frequency Disparities. Proc. ECCV'92, pp. 661-669.
6. Malik, J. and Rosenholtz, R. (1997). Computing Local Surface Orientation and Shape From Texture for Curved Surfaces. *Int. J. of Computer Vision* **23-2**, pp. 149-168.

7. Mallat, S. (1997). *A wavelet tour of signal processing*. Academic Press.
8. Manmatha, R. (1994). Measuring the Affine Transform using Gaussian Filters. *Proc. 3rd European Conf. on Computer Vision*, pp. 159-164, Stockholm, Sweden.
9. Pan, H.P. (1996). General Stereo Image Matching using Symmetric Complex Wavelets. *Proceedings of SPIE Wavelet Applications in Signal and Image Processing IV*.
10. Perrin, J., Torr  sani, B., and Fuchs, P. (1999). A Localized Correlation Function for Stereoscopic Image Matching. *Traitement du Signal* **16-1**.
11. Schaffalitzky, F., Zisserman, A. (2001). Viewpoint invariant Texture Matching and Wide Baseline Stereo. *Proceedings of ICCV*.
12. Tyler, C.W. and Sutter, E.E. (1979). Depth from spatial frequency difference: an old kind of stereopsis? *Vision Research* 19:859-865.
13. Zitnick, C.L. and Kanade, T. (2000). A Cooperative Algorithm for Stereo Matching and Occlusion Detection. *IEEE Trans. Pat. Anal. and Mach. Intell.* **22-7**, pp. 675-684.

# Stereo Matching Using Belief Propagation

Jian Sun<sup>\*1,2</sup>, Heung-Yeung Shum<sup>2</sup>, and Nan-Ning Zheng<sup>1</sup>

<sup>1</sup> Artificial Intelligence and Robotics Lab, Xi'an Jiaotong University, China  
{sj,nnzheng}@aiar.xjtu.edu.cn

<sup>2</sup> Visual Computing Group, Microsoft Research Asia, Beijing  
hshum@microsoft.com

**Abstract.** In this paper, we formulate the stereo matching problem as a Markov network consisting of three coupled Markov random fields (MRF's). These three MRF's model a smooth field for depth/disparity, a line process for depth discontinuity and a binary process for occlusion, respectively. After eliminating the line process and the binary process by introducing two robust functions, we obtain the maximum a posteriori (MAP) estimation in the Markov network by applying a Bayesian belief propagation (BP) algorithm. Furthermore, we extend our basic stereo model to incorporate other visual cues (e.g., image segmentation) that are not modeled in the three MRF's, and again obtain the MAP solution. Experimental results demonstrate that our method outperforms the state-of-art stereo algorithms for most test cases.

## 1 Introduction

Stereo vision infers scene geometry from two images with different viewpoints. Classical dense two-frame stereo matching computes a dense disparity or depth map from a pair of images under a known camera configuration. In general, the scene is assumed Lambertian or intensity-consistent from different viewpoints, without specularities, reflection, or transparency.

Stereo matching is difficult because of the following reasons.

- Noise: There are always unavoidable light variations, image blurring, and sensor noise in image formation.
- Textureless region: Information from highly textured regions needs to be propagated into textureless regions for stereo matching.
- Depth discontinuity: Information propagation should stop at object boundaries.
- Occlusion: Those occluded pixels in the reference view cannot be matched with the other view.

Therefore, stereo matching is an ill-posed problem with inherent ambiguities. Obviously, some constraints are needed to get a good “guess” of scene structure. Many methods have been proposed to encode various constraints,

---

<sup>\*</sup> This work was performed while the first author was visiting Microsoft Research Asia



e.g., intensity-consistency, local smoothness constraints, generalized order constraints, and uniqueness constraints. It has been shown that these constraints can be modeled well as priors in the Bayesian approach to stereo matching.

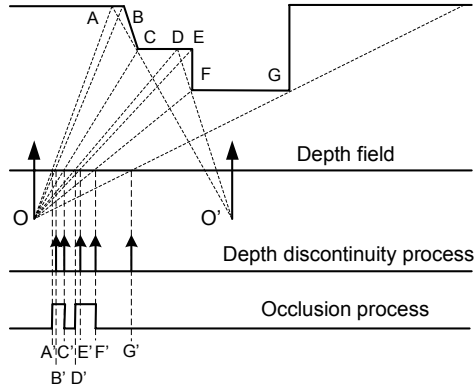
In this paper, after reviewing related works in Section 2 on stereo matching, and especially on the Bayesian approaches, we propose in Section 3 a Bayesian stereo matching approach to explicitly model discontinuities, occlusion and the disparity field in the Bayesian framework. In Section 4, Bayesian Belief Propagation is used to infer the stereo matching. The basic stereo model is then extended in Section 5 to integrate multiple cues, e.g., region similarity. The experimental results shown in Section 6 demonstrate that our model is effective and efficient. Finally, we discuss in Section 7 why our stereo matching with belief propagation outperforms the state-of-art stereo algorithms.

## 2 Related Works

In this section, we review related stereo algorithms and especially those using the Bayesian approach. We refer the reader to a detailed and updated taxonomy of dense, two-frame stereo correspondence algorithms by Scharstein and Szeliski [21]. It also provides a testbed for quantitative evaluation of stereo algorithms.

A stereo algorithm is called a global method if there is a global objective function to be optimized. Otherwise it is called a local method. The central problem of local or window-based stereo matching methods is to determine the optimal support window for each pixel. An ideal support region should be bigger in textureless regions and should be suspended at depth discontinuities. The fixed window is obviously invalid at depth discontinuities. Some improved window-based methods, such as adaptive windows [16], shiftable windows [5] and compact windows [23] try to avoid the windows that span depth discontinuities.

Bayesian methods (e.g., [11,1,5,8,15]) are global methods that model discontinuities and occlusion. Geiger et al. [11] derived an occlusion process and a disparity field from a matching process. Assuming an “order constraint” and “uniqueness constraint”, the matching process becomes a “path-finding” problem what the global optimum is obtained by dynamic programming. Belhumeur [1] defined a set of priors from a simple scene to a complex scene. A simplified relationship between disparity and occlusion is used to solve scan line matching by dynamic programming. Unlike Geiger and Belhumeur who enforced a piecewise smooth constraint, Cox et al. [8] and Bobick & Intille [5] did not require the smoothing prior. Assuming corresponding features are normally distributed and a fixed cost for occlusion, Cox also proposed a dynamic programming solution using only the occlusion constraint and ordering constraints. Bobick & Intille incorporated Ground Control Points constraint to reduce the sensitivity to occlusion cost and complexity of Cox’s dynamic programming. These methods use dynamic programming and assume that the occlusion cost is the same in each scanline. Ignorance of dependence between scanlines results in the characteristic “streaking” in the disparity maps.



**Fig. 1.** A scene illustrates the geometry relationship among depth, discontinuities and occlusions.  $O$  and  $O'$  are optical centers of two cameras. Discontinuities occur at  $B', C', E', F'$  and  $G'$ . Occlusion occur in  $[A, C]$  and  $[D, F]$

In general, Bayesian stereo matching can be formulated as a maximum a posteriori MRF (MAP-MRF) problem. There are several methods to solve the MAP-MRF problem: simulated annealing [12], Mean-Field annealing [10], the Graduated Non-Convexity algorithm (GNC) [4], and Variational approximation [14]. Finding a solution by simulated annealing can often take an unacceptably long time although global optimization is achieved in theory. Mean-Field annealing is a deterministic approximation to simulated annealing by attempting to average over the statistics of the annealing process. It reduces execution time at the expense of solution quality. GNC can only be applied to some special energy functions. Variational approximation converges to a local minimum. Graph Cut (GC) [6] is a fast efficient algorithm to solve a MAP-MRF whose energy function is Potts or Generalized Potts.

### 3 Basic Stereo Model

In our work, to handle occlusion and depth discontinuity explicitly, we model stereo vision by three coupled MRF's:  $D$  is the smooth disparity field of the reference view,  $L$  is a spatial line process located on the dual lattice and representing explicitly the presence or absence of depth discontinuities in the reference view, and  $O$  is a spatial binary process to indicate occlusion regions in the reference view. Figure 1 illustrates these processes in the 1D case. By using Bayes' rule, the joint posterior probability over  $D$ ,  $L$  and  $O$  given a pair of stereo images ( $I = (I_L, I_R)$  where  $I_L$  is the left and reference image) is:

$$P(D, L, O|I) = \frac{P(I|D, L, O)P(D, L, O)}{P(I)}. \quad (1)$$

Without occlusion,  $\{D, L\}$  are coupled MRF's that model a piece-wise smooth surface by two random fields: one represents the variable required to

estimate, the other represents its discontinuities. This model was proposed by [12]. However, the occlusion problem in stereo vision is not included in this kind of model explicitly. In image formation, the piece-wise smooth scene is projected on a pair of stereo images. Some regions are only visible in one image. There is no matching pixel in the other view for each pixel in the occlusion region. We assume that likelihood  $P(I|D, O, L)$  is independent of  $L$  because the observation is pixel-based, and ignore the statistical dependence between  $O$  and  $\{D, L\}$ :

$$P(I|D, O, L) = P(I|D, O), \quad (2)$$

$$P(D, O, L) = P(D, L)P(O). \quad (3)$$

The basic stereo model now becomes

$$P(D, O, L|I) = \frac{P(I|D, O)P(D, L)P(O)}{P(I)}. \quad (4)$$

### 3.1 Likelihood

Assuming observation noises follow an independent identical distribution(i.i.d), we can define the likelihood  $P(I|D, O)$  as:

$$P(I|D, O) \propto \prod_{s \notin O} \exp(-F(s, d_s, I)) \quad (5)$$

where  $F(s, d_s, I)$  is the matching cost function of pixel  $s$  with disparity  $d_s$  given observation  $I$ . Our likelihood considers the pixels only in non-occluded areas  $s \notin O$  because likelihood of the pixels in occluded areas can not be well defined. We use the pixel dissimilarity that is provably insensitive to sampling [2]:

$$F(s, d_s, I) = \min\{\bar{d}(s, s', I)/\sigma_f, \bar{d}(s', s, I)/\sigma_f\}$$

where  $\bar{d}(s, s', I) = \min\{|I_L(s) - I_R^-(s')|, |I_L(s) - I_R(s')|, |I_L(s) - I_R^+(s')|\}$ ,  $s'$  is the matching pixel in right view of  $s$  with disparity  $d_s$ ,  $I_R^-(s')$  is the linearly interpolated intensity halfway between  $s'$  and its neighboring pixel to the left,  $I_R^+(s')$  is to the right,  $\bar{d}(s', s, I)$  is the symmetric version of  $\bar{d}(s, s', I)$  and  $\sigma_f$  is the variance to be estimated.

### 3.2 Prior

Deriving appropriate priors to encode constraints directly is not only hard but may also result in too many annoying hyper parameters to find the solution easily. The Markov property asserts that the probability of a site in the field depends only on its neighboring sites. By specifying the first order neighborhood  $G(s)$  and  $N(s) = \{t|t > s, t \in G(s)\}$  of site  $s$ , the prior 3 can be expanded as:

$$P(D, L, O) \propto \prod_s \prod_{t \in N(s)} \exp(-\varphi_c(d_s, d_t, l_{s,t})) \prod_s \exp(-\eta_c(o_s)) \quad (6)$$

where  $\varphi_c(d_s, d_t, l_{s,t})$  is the joint clique potential function of  $d_s$ ,  $d_t$  and  $l_{s,t}$ , and  $\eta_c(o_s)$  is the clique potential function of  $o_s$ .  $\varphi_c(d_s, d_t, l_{s,t})$  and  $\eta_c(o_s)$  are user-customized functions to force the constraints for stereo matching.  $\varphi_c(d_s, d_t, l_{s,t})$  and  $\eta_c(o_s)$  also determine the distributions of  $\{D, L, O\}$ . To enforce spatial interactions between  $d_s$  and  $l_s$ , we define  $\varphi_c(d_s, l_s)$  as follows:

$$\varphi_c(d_s, d_t, l_{s,t}) = \varphi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}) \quad (7)$$

where  $\varphi(d_s, d_t)$  penalizes the different assignments of neighbor sites when no discontinuity exists between them, and  $\gamma(l_{s,t})$  penalizes the occurrence of a discontinuity between site  $s$  and  $t$ .

Combining (5), (6) and (7), our basic stereo model becomes:

$$P(D, O, L|I) \propto \prod_{s \notin O} \exp(-F(s, d_s, I)) \prod_s \exp(-\eta_c(o_s)) \prod_s \prod_{t \in N(s)} \exp(-(\varphi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}))). \quad (8)$$

## 4 Approximate Inference by Belief Propagation

In the last section, we model stereo matching by three coupled MRFs. After converting MRFs to the corresponding Markov network, the approximate inference algorithm Loopy Belief Propagation can be used to approximate the posterior probability for stereo matching.

### 4.1 From Line Process to Outlier Process

It is hard not only to specify appropriate forms of  $\varphi(d_s, d_t)$ ,  $\gamma(l_{s,t})$  and  $\eta_c(o_s)$ , but also to do inference in a continuous MRF and two binary MRFs. Fortunately, the unification of line process and robust statistics [3] provides us a way to eliminate the binary random variable from our MAP problem. If we simplify  $\eta_c(o_s)$  by ignoring the spatial interaction of occlusion sites<sup>1</sup>

$$\eta_c(o_s) = \eta(o_s) \quad (10)$$

we can rewrite our MAP problem as:

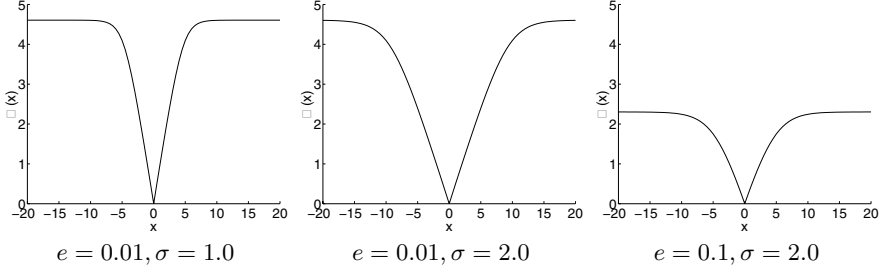
$$\max_{D, L, O} P(D, L, O|I) = \max_D \left\{ \max_O \prod_s \exp(-(F(s, d_s, I)(1 - o_s) + \eta(o_s))) \right. \\ \left. \max_L \prod_s \prod_{t \in N(s)} \exp(-(\varphi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}))) \right\} \quad (11)$$

---

<sup>1</sup> The complete form of  $\eta_c(o_s)$  should be:

$$\eta_c(o_s) = \eta(o_s) + \sum_{t \in N(s)} \eta'(o_s, o_t) \quad (9)$$

where  $\eta(o_s)$  is a single-site clique potential function that penalizes the occurrence of occlusion, and  $\eta'(o_s, o_t)$  is a pair-site cliques potential function that penalizes the different assignments of  $o_s$  and  $o_t$ .



**Fig. 2.** Robust function  $\rho(x) = -\ln((1 - e) \exp(-\frac{|x|}{\sigma}) + e)$  derived from TV model.

Now, we upgrade the binary process  $l_{st}$  and  $o_s$  to analog process  $l_{st}^a$  and  $o_s^a$  (“outlier process” [3]) by allowing  $0 \leq l_{st}^a \leq 1$  and  $0 \leq o_s^a \leq 1$ . For the first term,

$$\begin{aligned} & \max_O \prod_s \exp(-(F(s, d_s, I)(1 - o_s^a) + \eta(o_s^a))) \\ &= \exp(-\min_O \sum_s (F(s, d_s, I)(1 - o_s^a) + \eta(o_s^a))) \end{aligned} \quad (12)$$

where  $\min_O \sum_s (F(s, d_s, I)(1 - o_s^a) + \eta(o_s^a))$  is the objective function of a robust estimator. The robust function of this robust estimator is

$$\rho_d(d_s) = \min_{o_s^a} (F(s, d_s, I)(1 - o_s^a) + \eta(o_s^a)) \quad (13)$$

and for the second term, we also have a robust function  $\rho_p(d_s, d_t)$ :

$$\rho_p(d_s, d_t) = \min_{l_{s,t}^a} (\varphi(d_s, d_t)(1 - l_{s,t}^a) + \gamma(l_{s,t}^a)). \quad (14)$$

We get the posterior probability over  $D$  defined by two robust functions:

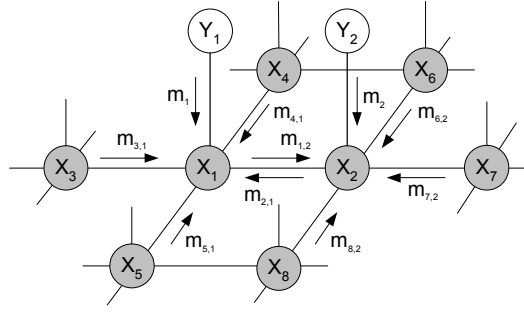
$$P(D|I) \propto \prod_s \exp(-\rho_d(F(s, d_s, I))) \prod_s \prod_{t \in N(s)} \exp(-\rho_p(d_s, d_t)). \quad (15)$$

Thus, we not only eliminate two analog line processes via an outlier process but also model outliers in measurements. We convert the task of modelling the priors of the occlusion process and depth discontinuity process explicitly into defining two robust functions that model occlusion and discontinuity implicitly.

In this paper, our robust functions are derived from the Total Variance(TV) model [18] with the potential function  $\rho(x) = |x|$  because of its discontinuity preserving property. We truncate this potential function as our robust function:

$$\begin{aligned} \rho_d(d_s) &= -\ln((1 - e_d) \exp(-\frac{|F(s, d_s, I)|}{\sigma_d}) + e_d) \\ \rho_p(d_s, d_t) &= -\ln((1 - e_p) \exp(-\frac{|d_s - d_t|}{\sigma_p}) + e_p) \end{aligned}$$

Figure 2 shows different shapes of our robust functions. By varying parameters  $e$  and  $\sigma$ , we control the shape of the robust function, and therefore the posterior probability.



**Fig. 3.** Local message passing in Markov Network. In "max-product" algorithm, the new message sent from node  $x_1$  to  $x_2$  is:  $m_{1,2}^{new} \leftarrow \kappa \max_{x_1} \psi_{12}(x_1, x_2) m_1 m_{3,1} m_{4,1} m_{5,1}$ . The belief at node  $x_1$  is computed as:  $b_1 \leftarrow \kappa m_1 m_{2,1} m_{3,1} m_{4,1} m_{5,1}$ .

## 4.2 Belief Propagation

The model that is most similar to our posterior probability (15) is Scharstein & Szeliski's [20]. Unlike Scharstein & Szeliski, where a nonlinear diffusion algorithm is used, we address this MAP problem by Belief Propagation. Belief Propagation is an exact inference method proposed by Pearl[19] in the belief network without loops. Loopy Belief Propagation is just Belief Propagation that ignores the existence of loops in the networks. It has been applied successfully to some vision [9] and communication [24] problems despite the presence of network loops.

The posterior probability (15) over  $D$  is exactly a Markov Network in the literature of probabilistic graph models as shown in Figure 3. In the Markov Network, random variable  $d_s$  in our stereo model is represented by a hidden node  $x_s$ . A "private" observation node  $y_s$  is connected to each  $x_s$ . Each  $y_s$  is a vector where each element is the matching cost given different assignments of node  $x_s$ . By denoting  $X = \{x_s\}$  and  $Y = \{y_s\}$ , (15) can be represented with  $x_s$  and  $y_s$ :

$$P(X|Y) \propto \prod_{s,t:s>t,t \in N(s)} \psi_{st}(x_s, x_t) \prod_s \psi_s(x_s, y_s) \quad (16)$$

where

$$\psi_{st}(x_s, x_t) = \exp(-\rho_p(x_s, x_t)) \quad (17)$$

$$\psi_s(x_s, y_s) \propto \exp(-\rho_d(F(s, x_s, I))) \quad (18)$$

$\psi_{st}(x_s, x_t)$  is called compatibility matrix between node  $x_s$  and  $x_t$ , and  $\psi_s(x_s, y_s)$  is called the local evidence for node  $x_s$ . If the disparity level is  $L$ ,  $\psi_{st}(x_s, x_t)$  is a  $L \times L$  matrix and  $\psi_s(x_s, y_s)$  is a  $L$ -length vector.

Belief Propagation is an iterative inference algorithm that propagates messages in the network. Let  $m_{st}(x_s, x_t)$  be the message that node  $x_s$  sends to  $x_t$ ,  $m_s(x_s, y_s)$  be the message that observed node  $y_s$  sends to node  $x_s$ ,  $b_s(x_s)$  be the belief at node  $x_s$ . Note that  $m_{st}(x_s, x_t)$ ,  $m_s(x_s, y_s)$  and  $b_s(x_s)$  are all 1D vectors. We simplify  $m_{st}(x_s, x_t)$  as  $m_{st}(x_t)$ , and  $m_s(x_s, y_s)$  as  $m_s(x_s)$ . There are

two kinds of BP algorithms with different message update rules: “max-product” and “sum-product”, which maximize the joint posterior of the network, and the marginal posterior of each node, respectively. The standard “max-product” algorithm is shown below.

1. Initialize all messages as uniform distributions
2. Update messages iteratively for  $i=1:T$

$$m_{st}^{i+1}(x_t) \leftarrow \kappa \max_{x_s} \psi_{st}(x_s, x_t) m_s^i(x_s) \prod_{x_k \in N(x_s) \setminus x_s} m_{ks}^i(x_s)$$

3. Compute beliefs

$$b_s(x_s) \leftarrow \kappa m_s(x_s) \prod_{x_k \in N(x_s)} m_{ks}(x_s)$$

$$x_s^{MAP} = \arg \max_{x_k} b_s(x_k)$$

For example, in Figure 3, the new message sent from node  $x_1$  to  $x_2$  is updated as:  $m_{1,2}^{new} \leftarrow \kappa \max_{x_1} \psi_{12}(x_1, x_2) m_1 m_{3,1} m_{4,1} m_{5,1}$ . The belief at node  $x_1$  is computed as:  $b_1 \leftarrow \kappa m_1 m_{2,1} m_{3,1} m_{4,1} m_{5,1}$  (the product of two messages is component-wise product). And  $\kappa$  is the normalization constant.

## 5 Integrating Multiple Cues

More constraints and priors (e.g., edges, corners, junctions, segmentation, visibility) can be incorporated to improve stereo matching. For instance, a segmentation-based stereo algorithm [22] has been recently proposed based on the assumption that the depth discontinuities occur on the boundary of the segmented regions. In [22], the segmentation results are used as hard constraints. In our work, we make use of image segmentation but incorporate segmentation results into our basic stereo model as soft constraints (prior) under a probabilistic framework.

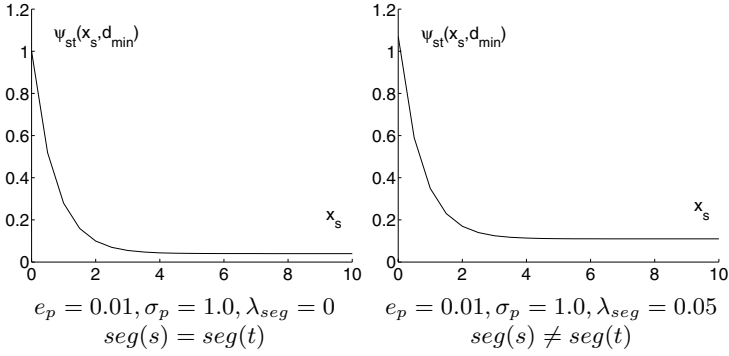
With additional cues, we extend the basic stereo model (15):

$$P(D, O, L|I) \propto \prod_s \exp(-\rho_d(F(s, d_s, I))) \prod_s \prod_{t \in N(s)} \exp(-\varphi_c(d_s, d_t, l_{s,t})) \exp(-\rho_{pcue}(d_s, d_t)) \quad (19)$$

where  $\rho_{pcue}(d_s, d_t)$  encodes some constraints between sites. To integrate region similarities from image segmentation, we define  $\rho_{pcue}(d_s, d_t)$  as:

$$\rho_{pcue}(d_s, d_t) = \rho_{seg}(d_s, d_t) = \begin{cases} 0 & seg(s) = seg(t) \\ \lambda_{seg} & seg(s) \neq seg(t) \end{cases} \quad (20)$$

where  $seg(s)$  is the label of the segmentation result at site  $s$ . The larger the  $\lambda_{seg}$ , the more difficulty in passing the message between neighbor sites. In other



**Fig. 4.** Left image is the first row of  $\psi_{st}(x_s, x_t)$  when node  $x_s$  and  $x_t$  are in same region. Right image is the first row of  $\psi_{st}(x_s, x_t)$  when node  $x_s$  and  $x_t$  are in different regions

words, the influence from neighbors becomes smaller as  $\lambda_{seg}$  increases. In our experiments, the segmentation labels are produced by the Mean-Shift algorithm [7]. The execution time is usually just a few seconds in all images used in our experiments.

According to (15), the compatibility matrix  $\psi_{st}(x_s, x_t)$  can be rewritten as:

$$\psi_{st}(x_s, x_t) = \exp(-\rho_p(x_s, x_t)) \exp(-\rho_{pcue}(x_s, x_t)) \quad (21)$$

Figure 4 shows the first rows of  $\psi_{st}(x_s, x_t)$  when  $x_s$  and  $x_t$  are in same region and in different regions.

## 6 Experimental Results

In this paper, we evaluate the performance of our stereo algorithm using the quality measures proposed in [21] with those measures based on known ground truth data listed in Table 1. In particular,  $B_{\bar{\mathbf{O}}}$  represents the overall performance of a stereo algorithm.

**Table 1.** Quality measures based on known ground truth data

Percentage of bad matching pixels in non-occlusion regions $\bar{\mathbf{O}}$	$B_{\bar{\mathbf{O}}} = \frac{1}{N} \sum_{s \in \bar{\mathbf{O}}} ( d(s) - d_T(s)  > \delta_d)$
Percentage of bad matching pixels in textureless regions $\bar{\mathbf{T}}$	$B_{\bar{\mathbf{T}}} = \frac{1}{N} \sum_{s \in \bar{\mathbf{T}}} ( d(s) - d_T(s)  > \delta_d)$
Percentage of bad matching pixels in depth discontinuity regions $\mathbf{D}$	$B_{\mathbf{D}} = \frac{1}{N} \sum_{s \notin \mathbf{D}} ( d(s) - d_T(s)  > \delta_d)$

The test data set consists of four pairs of images: “Map”, “Tsukuba”, “Sawtooth” and “Venus” [21]. “Tsukuba” is a complicated indoor environment with



slanted surfaces and contains a number of integer valued disparities. Other pairs consist of mainly slanted planes.

Table 2 shows the results of applying our BP algorithm to all four pairs of images. It also lists the results of other stereo algorithms. This table is courtesy of Scharstein and Szeliski (see <http://www.middlebury.edu/stereo/results.html> for details). Our results with and without image segmentation incorporated into stereo matching are shown in the first and the second row, respectively.

For a complicated environment like “Tsukuba”, incorporating image segmentation improves stereo matching significantly, with 40% error reduction in  $B_{\bar{0}}$ . In fact, our algorithm ranks as the best for “Tsukuba” and outperforms Graph Cut (with occlusion) [17] which was widely considered the state-of-art stereo matching algorithm. Our algorithm competes well with other stereo algorithms for the three other data sets, “Sawtooth”, “Venus” and “Map”. It is interesting to note that for these three data sets with simple slanted surfaces, incorporating image segmentation does not necessarily improve stereo matching, as seen from the first and second rows.

Figures 5 and 6 show the results obtained by our algorithm. The segmentation map is obtained by the Mean-Shift algorithm with default parameters suggested by [7]. Note that a fixed set of parameters  $\{e_d = 0.05, \sigma_d = 0.6, e_p = 0.01, \sigma_p = 8\}$  are used in our BP algorithm for all image pairs. Obviously, this set of parameters is not the optimal for “Map” data because the disparity range of this data is almost twice that of “Tsukuba” data’s disparity range.

The complexity of our BP algorithm is  $O(L^2NT)$  where  $N$  is the number of pixels,  $L$  is the number of disparities, and  $T$  is the number of iterations. For the “Tsukuba” data, it took 288 seconds on a Pentium III 500 MHz PC. It is comparable or slightly better than the graph cut algorithm reported in [21].

The local oscillation phenomena of the BP algorithm also occurred in our experiments. A time average operation is executed after a fixed number of iterations:  $m_{st}^t(x_t) = m_{st}^{t-1}(x_t) + m_{st}^t(x_t)$ . This heuristic worked well in our experiments.

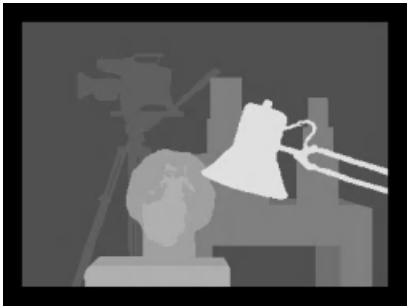
## 7 Discussion

**Why BP works?** The magic of the BP algorithm lies in its powerful message passing. A message presents the probability that the receiver should be at a disparity according to all information from the sender up to the current iteration. Message passing has two important properties. First, it is asymmetric. The entropy of the messages from high-confidence nodes to low-confidence nodes is smaller than the entropy of the messages from low-confidence nodes to high-confidence nodes. Second, it is adaptive. The influence of a message between a pair of nodes with larger divergence would be weakened more.

Therefore, BP’s message passing provides a time-varying adaptive smoothing mechanism for stereo matching to deal with textureless regions and depth discontinuities naturally. In textureless regions, for example, the influence of a



(a) Left (reference) Image



(b) Ground Truth



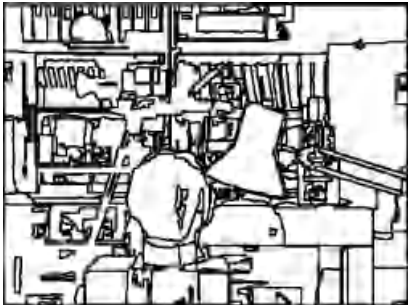
(c) Textureless regions



(d) Depth discontinuity regions



(e) Occlusion regions



(f) Image segmentation

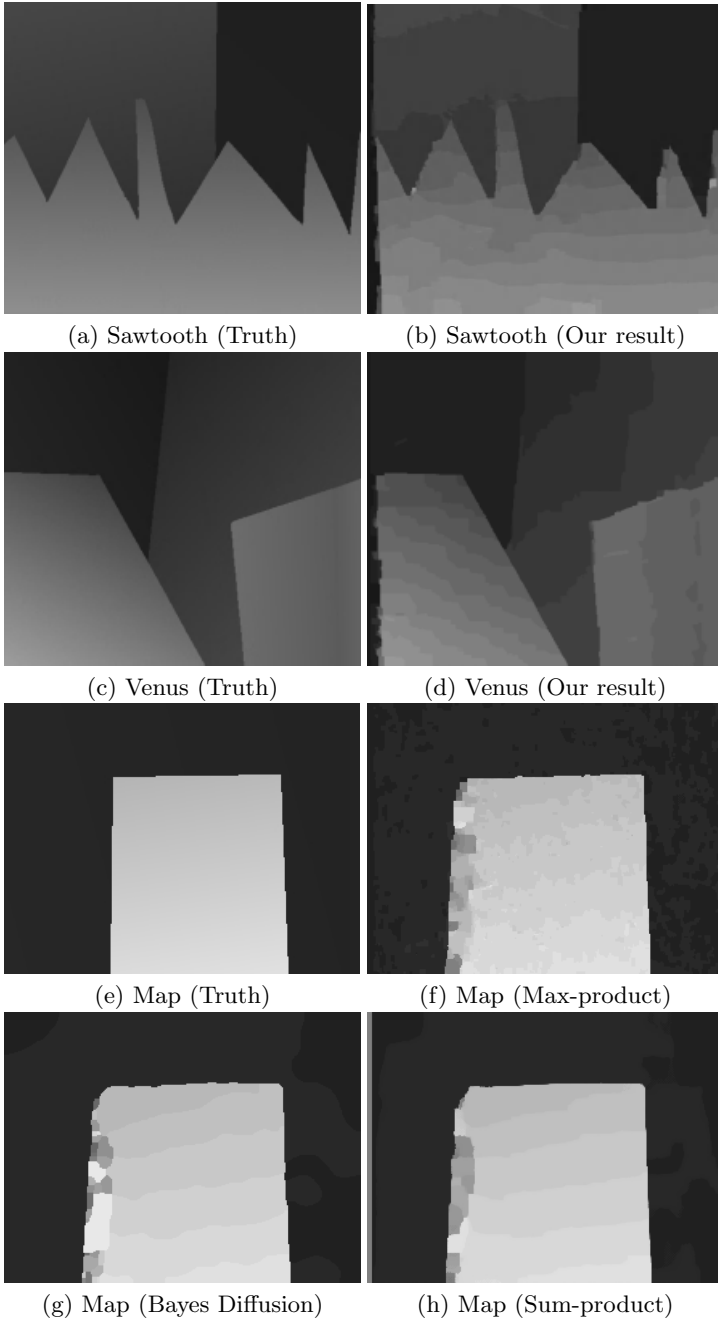


(g) “max-product” result



(h) “max-product” result with segmentation

**Fig. 5.** The results on Tsukuba dataset. (a)-(e) are given.



**Fig. 6.** The results of Sawtooth and Venus based on the “max-product” algorithm are shown in (b) and (d). For the Map data, the “max-product” result is shown in (f). Bayesian diffusion results with  $B_{\bar{O}} = 0.20$ ,  $B_{DD} = 2.49$  are shown in (g), while “sum-product” results with  $B_{\bar{O}} = 0.16$ ,  $B_{DD} = 2.11$  are shown in (h).

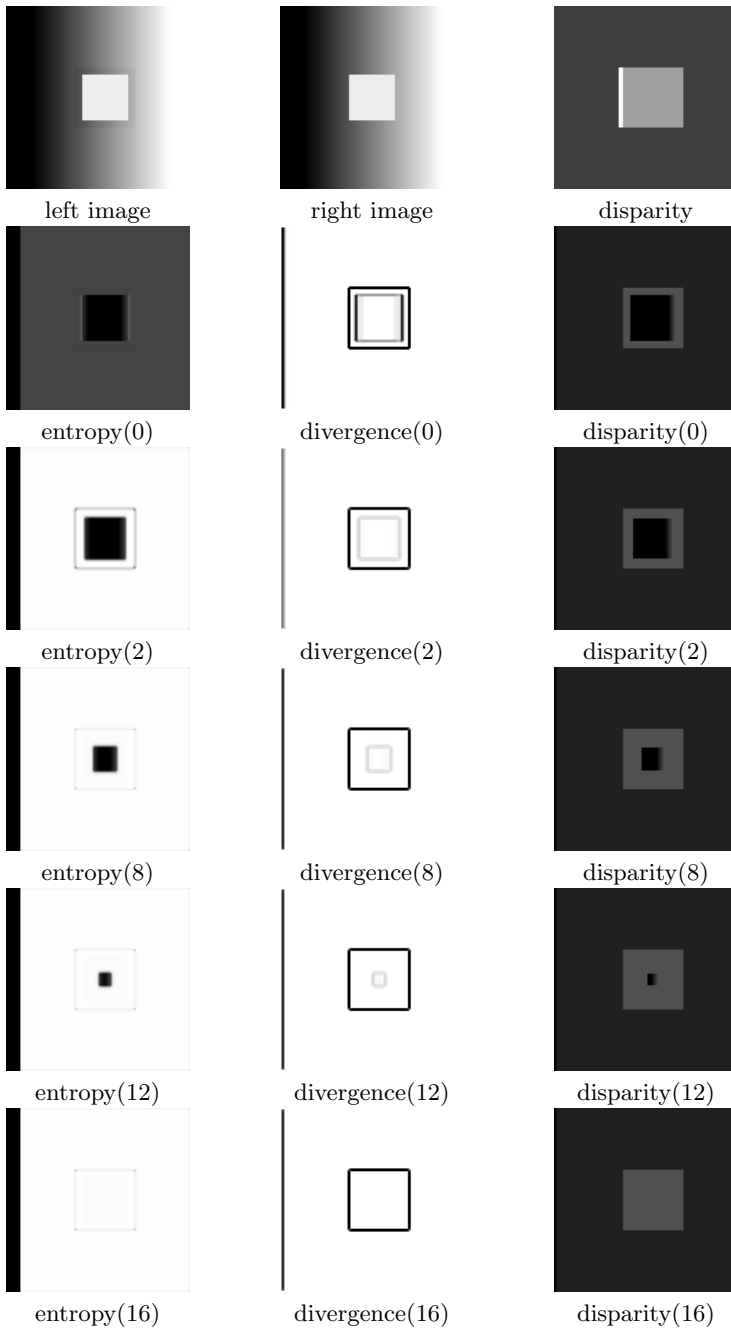
**Table 2.** The performance of different stereo algorithms with fixed parameters on four test image pairs. An underlined number is the best in its category.

Algorithms	Tsukuba			Sawtooth			Venus			Map	
	$B_{\bar{0}}$	$B_{\bar{T}}$	$B_{\bar{D}}$	$B_{\bar{0}}$	$B_{\bar{T}}$	$B_{\bar{D}}$	$B_{\bar{0}}$	$B_{\bar{T}}$	$B_{\bar{D}}$	$B_{\bar{0}}$	$B_{\bar{D}}$
Belief prop. (seg)	<u>1.15</u>	<u>0.42</u>	<u>6.31</u>	0.98	0.30	4.83	<u>1.00</u>	<u>0.76</u>	9.13	0.84	5.27
Belief prop.	1.61	0.66	9.17	0.85	0.37	7.92	1.17	1.00	12.87	0.67	3.42
Graph cuts [21]	1.96	1.06	9.41	1.36	0.23	6.57	1.36	1.75	6.63	0.33	4.40
GC+occl. [17]	1.27	0.43	6.90	<u>0.36</u>	<u>0.00</u>	<u>3.65</u>	2.79	5.39	<u>2.54</u>	1.79	10.08
Graph cuts [6]	1.86	1.00	9.35	0.42	0.14	3.76	1.69	2.30	5.40	2.39	9.35
Realtime SAD [13]	4.25	4.47	15.05	1.32	0.35	9.21	1.53	1.80	12.33	0.81	11.35
Bay. diff. [21]	6.49	11.62	12.29	1.43	0.69	9.29	3.89	7.15	18.17	<u>0.20</u>	<u>2.49</u>
SSD+MF [21]	5.26	3.86	24.65	2.14	0.72	13.08	3.81	6.93	12.94	0.66	9.35
Dyn. prog. [21]	3.43	3.22	12.34	4.54	3.59	13.11	8.47	12.76	17.61	3.77	13.93
Scanl. opt. [21]	4.94	6.50	11.94	4.19	2.95	12.14	9.71	14.98	18.20	4.61	10.22

message can be passed far away. On the other hand, the influence in discontinuous regions will fall off quickly. Figure 7 shows this adaptive smoothing procedure in an example. In Figure 7, the image pair is modified from that used in [16] and [20]. A linear ramp in the direction of the baseline is used as the underlying intensity pattern. The disparity of background and foreground is 2 and 5, respectively. Unlike [16] or [20], a smaller pure textureless square is overlapped in the center of the foreground in the ramp1 pair.

We use entropy  $H(b) = -\sum_i b_i \log b_i$  to measure the confidence of the belief, and the symmetric version of the Kullback-Leiber(KL) divergence  $KL_s(b^1 \| b^2) = \sum_i (b_i^1 - b_i^2) \log(\frac{b_i^1}{b_i^2})$  to measure the difference between belief  $b^1$  and  $b^2$ . Smaller entropy represents higher confidence of a belief. Larger divergence represents larger dissimilarity between beliefs. As shown in the figure, the entropy map of a belief represents the confidence of disparity estimation for each node. Clearly, the confidence of each node increases with each iteration. Note that the confidence in occlusion regions and corners is lower than that in other regions. This shows that the probabilistic method outputs not only a solution, but also its certainty. The divergence map of a belief shows where message-passing is stopped. The divergence map after convergence illustrates the ideal support regions.

**Assumptions and future work.** The Bayesian approaches have the advantage over energy minimization techniques that all assumptions need to be made explicitly. In fact, three important assumptions (2,3,10) are made in our model in order to apply BP. Although good experimental results are obtained with our model, it is worth investigating when these assumptions break. Many other future directions can also be pursued. Naturally, we plan to extend our work to multi-baseline stereo. We are also investigating how to improve stereo matching with other Bayesian inference techniques based on Markov networks such as generalized BP.



**Fig. 7.** Time-varying adaptive smoothing mechanism of the BP algorithm in stereo matching is illustrated from row 2 to row 6. The input image pair and the ground truth are shown in the first row. The number in the braces shows the iteration step.

## References

1. P.N. Belhumeur. A bayesian-approach to binocular stereopsis. *IJCV*, 19(3):237–260, 1996.
2. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *PAMI*, 20(4):401–406, 1998.
3. M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.
4. A. Blake and A. Zisserman. *Visual reconstruction*. MIT Press, 1987.
5. A.F. Bobick and S.S. Intille. Large occlusion stereo. *IJCV*, 33(3):1–20, 1999.
6. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *ICCV*, 1999.
7. D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. *CVPR*, 1997.
8. I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs. A maximum-likelihood stereo algorithm. *CVIU*, 63(3):542–567, 1996.
9. W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.
10. D. Geiger and F. Girosi. Parallel and deterministic algorithms from mrfs: Surface reconstruction. *PAMI*, 13(5):401–412, 1991.
11. D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *IJCV*, 14(3):211–226, 1995.
12. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 6(6):721–741, 1984.
13. H. Hirschmueller. Improvements in real-time correlation-based stereo vision. *IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.
14. B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *CVGIP*, 33(2):174–208, 1986.
15. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. *ECCV*, 1998.
16. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, 1994.
17. V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. *ICCV*, 2001.
18. S. Osher L.I. Rudin and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 27(60):259–268, 1992.
19. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988.
20. D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *IJCV*, 28(2):155–174, 1998.
21. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.
22. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. *ICCV*, 2001.
23. O. Veksler. Stereo matching by compact windows via minimum ratio cycle. *ICCV*, 2001.
24. W. T. Yedidia, J. S. Freeman and Weiss Y. Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR-2001-16, Mitsubishi Electric Research, 2001.

# Symmetric Sub-pixel Stereo Matching

Richard Szeliski<sup>1</sup> and Daniel Scharstein<sup>2</sup>

<sup>1</sup> Microsoft Research, Redmond, WA 98052, USA

<sup>2</sup> Middlebury College, Middlebury, VT 05753, USA

**Abstract.** Two central issues in stereo algorithm design are the matching criterion and the underlying smoothness assumptions. In this paper we propose a new stereo algorithm with novel approaches to both issues. We start with a careful analysis of the properties of the continuous *disparity space image* (DSI), and derive a new matching cost based on the reconstructed image signals. We then use a symmetric matching process that employs visibility constraints to assign disparities to a large fraction of pixels with minimal smoothness assumptions. While the matching operates on integer disparities, sub-pixel information is maintained throughout the process. Global smoothness assumptions are delayed until a later stage in which disparities are assigned in textureless and occluded areas. We validate our approach with experimental results on stereo images with ground truth.

## 1 Introduction

The last few years have seen a dramatic improvement in the quality of dense stereo matching algorithms [14]. A lot of this improvement can be attributed to better optimization algorithms and better smoothness constraints [6,4,16]. However, a remarkable amount of the improvement has also come from better matching metrics at the input [3]. In fact, Birchfield and Tomasi's sampling-insensitive dissimilarity measure is used by a number of today's best performing algorithms [6,4].

Using something better than just pixel-sampled intensity differences is not a new idea. For example, Matthies *et al.* interpolated scanlines by a factor of 4 using a cubic interpolant before computing the SSD score [11]. Tian and Huhns wrote an even earlier survey paper comparing various algorithms for sub-pixel registration [17]. In fact, some stereo and motion algorithms have always evaluated displacements on a half-pixel grid, but never mentioned this fact explicitly (P. Anandan, personal communication).

The set of initial matching costs that are fed into a stereo matcher's optimization stage is often called the *disparity space image* (DSI) [19,5]. However, while the concept of stereo matching as finding an optimal surface through this space has been around for a while [19,2,5], relatively little attention has been paid to the proper sampling and treatment of the DSI.

In this paper, we take a more careful look at the structure of the DSI, including its frequency characteristics and the effects of using different interpolators in sub-pixel registration. Among the questions we ask are: What does the DSI look like? How finely do we need to sample it? Does it matter what interpolator we use?

We also propose a number of novel modifications to the matching cost that produce a better set of initial high-quality matches, at least in textured, unoccluded areas. It is

our contention that filling in textureless and occluded areas is best left to a later stage of processing [5,6,4,16].

In the second half of the paper, we show how the local structure of the DSI can be used to select *certain* matches, i.e., matches that are correct with high probability. (Intille and Bobick [5] call such points *ground control points*.) We also develop an iterative algorithm that adds more matches, using a combination of uniqueness enforcement (filling the DSI with high costs in already matched columns), and doing more aggregation for pixels with multiple possible matches. We present the final dense matching results of our approach, which are comparable in quality to recent stereo matching algorithms, but which do not require any global optimization algorithm.

The remainder of the paper is structured as follow. Section 2 presents our analysis of the DSI and discusses minimal sampling requirements. Section 3 develops some novel matching costs based on our analysis. The utility of these novel costs is validated experimentally in Section 4. Section 5 presents our algorithm for establishing certain matches and for iteratively adding more matches to this set. We conclude with some ideas for future research.

## 2 Matching Costs

In this section, we look at how matching costs are formulated. In particular, we analyze the structure of the DSI and its sampling properties and propose some improvements to commonly used matching costs.

Given two input images,  $I_L(x, y)$  and  $I_R(x, y)$ , we wish to find a disparity map  $d_L(x, y)$  such that the two images match as closely as possible

$$I_L(x, y) \approx I_R(x - d_L(x, y), y). \quad (1)$$

(We assume that the images have been rectified to have a horizontal epipolar geometry [12,9]. We will also want to impose some smoothness or other prior constraints on the function  $d_L(x, y)$ .)

Define the 3D *signed difference image* (SDI) as the intensity (or color) difference between the shifted left and right images,

$$SDI_L(x, y, d) = I_L(x, y) - I_R(x - d, y). \quad (2)$$

Let the raw *disparity space image* (DSI) be the squared difference (summed over all the color bands),

$$DSI_L(x, y, d) = \|SDI_L(x, y, d)\|^2. \quad (3)$$

Why do we use squared differences? The analysis for this case is simpler, and it also has some other advantages we will discuss shortly. In the ideal (continuous, noise-free) case with no occlusions, we expect  $DSI_L(x, y, d_L(x, y))$  to be zero.

Unfortunately, we do not actually have access to continuous, noise-free versions of  $I_L(x, y)$  and  $I_R(x, y)$ . Instead, we have sampled noisy versions,  $\hat{I}_L(x_i, y_i)$  and  $\hat{I}_R(x_i, y_i)$ ,

$$\hat{I}_L(x_i, y_i) = [I_L * h](x_i, y_i) + n_L(x_i, y_i) \quad (4)$$

$$\hat{I}_R(x_i, y_i) = [I_R * h](x_i, y_i) + n_R(x_i, y_i), \quad (5)$$



where  $h(x, y)$  is the combined point-spread-function of the optics and sampling sensor (e.g., it incorporates the CCD fill factor [18]), and  $n_L$  is the (integrated) imaging noise.

Given that we can only evaluate the DSI at the integral grid positions  $(x_i, y_i)$ , we have to ask whether this sampling of the DSI is adequate, or whether there is severe aliasing in the resulting signal. We cannot, of course, reconstruct the true DSI since we have already band-limited, corrupted, and sampled the original images. However, we can (in principle) reconstruct continuous signals from the noisy samples, and then compute their continuous DSI. The reconstructed signal can be written as

$$\bar{I}_L(x, y) = \sum_i \hat{I}_L(x_i, y_i) g(x - x_i, y - y_i) \quad (6)$$

$$= \tilde{I}_L(x, y) + \tilde{n}_L(x, y), \quad (7)$$

where  $g(x, y)$  is a reconstruction filter,  $\tilde{I}_L(x, y)$  is the sampled and reconstructed version of the *clean* (original) signal, and  $\tilde{n}_L(x, y)$  is an interpolated version of the noise. This latter signal is a band-limited version of continuous Gaussian noise (assuming that the discrete noise is i.i.d. Gaussian).

We can then write the interpolated SDI and DSI as

$$\overline{SDI}_L(x, y, d) = \bar{I}_L(x, y) - \bar{I}_L(x - d, y) \quad \text{and} \quad (8)$$

$$\overline{DSI}_L(x, y, d) = \|\overline{SDI}_L(x, y, d)\|^2. \quad (9)$$

What can we say about the structure of these signals?

The answer can be found by taking a Fourier transform of the SDI. Let us fix  $y$  for now and just look at a single scanline,

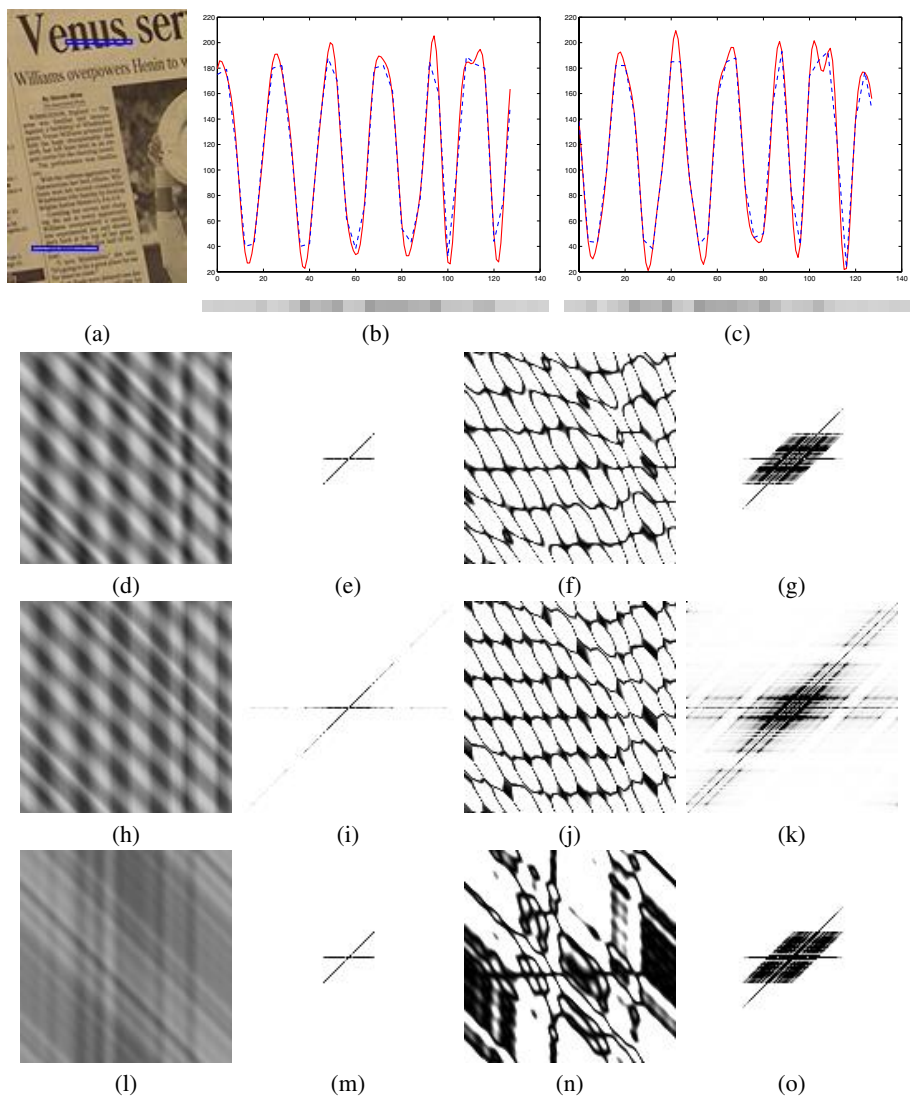
$$\mathcal{F}\{\overline{SDI}\} = H_L(f_x) - H_R(f_x) e^{j2\pi(f_x - f_d)}, \quad (10)$$

where  $H_L$  and  $H_R$  are the Fourier transforms of  $\bar{I}_L$  and  $\bar{I}_R$ .

Figure 1 shows the SDIs and DSIs for two scanlines taken from the 38th and 148th row of the image in Figure 1a, along with their Fourier transforms. The first term in (10) corresponds to the horizontal line in the SDI's Fourier transform (second column of Figure 1), while the second term, which involves the disparity, is the slanted line.

Squaring the SDI to obtain the DSI (third column in Figure 1) is equivalent to convolving the Fourier transform with itself (fourth column in Figure 1). The resulting signal has twice the bandwidth in  $x$  and  $d$  as the original SDI (which has the same bandwidth as the interpolated signal). It is also interesting to look at the structure of the DSI itself. The thin diagonal stripes are spurious bad matches (dark-light transitions matching light-dark transitions), while the horizontal stripes are good matching regions (the straighter and darker the better).

What can we infer from this analysis? First, the continuous DSI has significant frequency content above the frequencies present in the original intensity signal. Second, the amount of additional content depends on the quality of the interpolator applied to the signal. Thus, when perfect band-limited reconstruction (a sinc filter) is used, the resulting DSI signal only has twice the frequency of the image. It is therefore adequate (in theory) to sample the DSI at 1/2 pixel intervals in  $x$  and  $d$ . When a poorer interpolant



**Fig. 1.** Sample SDIs and DSIs and their Fourier transforms. (a) Original color image with two selected scanlines; (b–c) profiles of second selected scanline (L148); notice how the sinc-interpolated signals (red, solid) are more similar than the linearly interpolated ones (blue, dashed). (d–g) Signed Difference Image (SDI) and its transform, and Disparity Space Image (DSI) and its transform for L38, using perfect (sinc) interpolation; (h–k) same images using piecewise linear interpolation; (l–o) same images for L148 and perfect interpolation. (See the electronic version of this paper for color images.)

such as piecewise linear interpolation is used, the sampling may have to be much higher. The same is true when a different non-linearity is used to go from the SDI to the DSI,

e.g., when absolute differences or robust measures are used. This is one of the reasons we prefer to use squared difference measures. Other reasons include the statistical optimality of the DSI as the log likelihood measure under Gaussian noise, and the ability to fit quadratics to the locally linearized expansion of the DSI.

We can summarize these observations in the following Lemma:

**Lemma 1:** *To properly reconstruct a Disparity Space Image (DSI), it must be sampled at at least twice the horizontal and disparity frequency as the original image (i.e., we must use at least 1/2 pixel samples and disparity steps).*

It is interesting to note that if a piecewise linear interpolant is applied between image samples before differencing and squaring, the resulting DSI is piecewise quadratic. Therefore, it suffices in principle to simply compute one additional squared difference between pixels, and to then fit a piecewise quadratic model. While this does reconstruct a continuous DSI, there is no guarantee that this DSI will have the same behavior near true matches as a better reconstructed DSI. Also, the resulting minima will be sensitive to the original placement of samples, i.e., a significant bias towards integral disparities will exist [15].

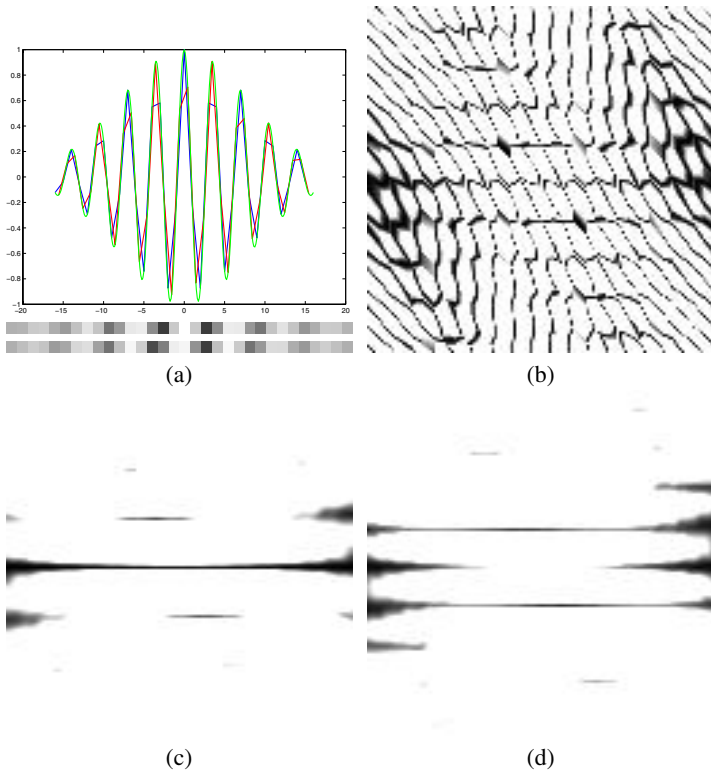
For example, if the original signal is a fairly high-frequency chirp (Figure 2a), then applying a piecewise linear interpolant will fail to correctly match the signal with a fractionally shifted version. Figure 2b and c show the results of aggregating the original raw DSIs with a 7-pixel filter (see Section 3). Clearly, using the linear interpolant will result in the wrong disparity minimum being selected in the central portion (where the central horizontal line is weak). One might ask whether such high-frequency signals really exist in practice, but it should be clear from Figure 1b and c that they do.

### 3 Improved Matching Costs

Given the above analysis, how can we design a better initial matching cost? Birchfield and Tomasi [3] and Shimizu and Okutomi [15] have both observed problems with integral DSI sampling, and have proposed different methods to overcome this problem.

Birchfield and Tomasi's *sampling-insensitive dissimilarity measure* compares each pixel in the reference image against the linearly interpolated signal in the matching image, and takes the minimum squared error as the matching cost. It then reverses the role of the reference and matching images, and takes the minimum of the resulting two cost measures. In terms of our continuous DSI analysis, this is equivalent to sampling the DSI at integral  $x$  locations, and computing the minimum value vertically and diagonally around each integral  $d$  value, based on a piecewise linear reconstruction of the DSI from integral samples. We generalize Birchfield and Tomasi's matching measure using the following two ideas:

**Symmetric Matching of Interpolated Signals** First of all, we interpolate both signals up by a factor  $s$  using an arbitrary interpolation filter. In this paper, we study linear ( $o = 1$ ) and cubic ( $o = 3$ ) interpolants. We then compute the squared differences between *all* of the interpolated and shifted samples, as opposed to just between the original left (reference) image pixels and the interpolated and shifted right (matching) image samples. This difference signal is then reduced back to the original horizontal

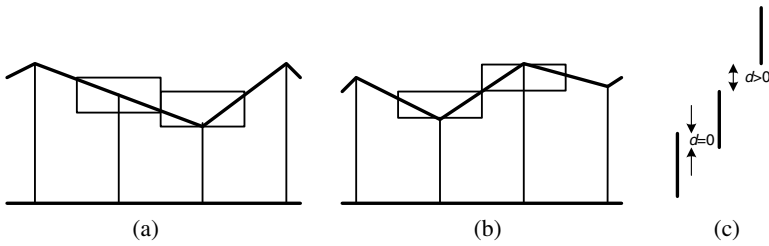


**Fig. 2.** Chirp signal matching: (a) the continuous green signal is sampled discretely to obtain the red and blue signals; (b) Disparity Space Image (DSI) for linear interpolation; (c) horizontally aggregated DSI for sinc interpolation, showing correct minimum; (d) horizontally aggregated DSI for linear interpolation, with incorrect minima near the center.

image sampling using a symmetric filter of width  $s$  and then downsampling. A higher-order filter could potentially be used, but we wish to keep discontinuities in depth sharp in the DSI, so we prefer a simple box filter.

**Interval Matching** If we wish to still apply the idea of a sampling-insensitive dissimilarity measure [3], we can still do this on the interpolated signals before downsampling. However, rather than treating the reference and matching images asymmetrically and then reversing the roles of reference and matching (as in [3]), we have developed the following related variant that is based on interval analysis.

Figure 3 shows two signals that have been interpolated to yield the set of discrete intensity samples shown as vertical lines. The original Birchfield-Tomasi measure compares a pixel in the reference image with the interval in the matching image defined by the center pixel and its two  $1/2$ -sample interpolated values (rectangular boxes in Figure 3b). It then performs this same computation switching the reference and match images, and takes the minimum of the resulting two costs. Our version of the algorithm



**Fig. 3.** Interval analysis: (a–b) two signals with their corresponding half-sample intervals; (c) three intervals being compared (differenced).

simply compares the two *intervals*, one from the left scanline, the other from the right, rather than comparing values against intervals. The unsigned difference between two intervals is trivial to compute: it is 0 if the intervals overlap (Figure 3c), else it is the gap between the two intervals. A signed difference could also be obtained by keeping track of which interval is higher, but in our case this is unnecessary since we square the differences after computing them.

When working with color images, we currently apply this interval analysis to each color band separately. In principle, the same sub-pixel offset should be used for all three channels, but the problem then becomes a more complicated quadratic minimization problem instead of simple interval analysis.

**Local Minimum Finding (Quadratic Fit)** An alternative to doing such interval analysis is to directly compute the squared differences, and to then fit a parabola to the resulting sampled DSI. This is a classic approach to obtaining sub-pixel disparity estimates [17, 1, 11], although applying it directly to integer-valued displacements (disparities) can lead to severe biases [15].

When the DSI has been adequately sampled, however, this is a very useful alternative for estimating the analytic minimum from the (fractionally) sampled DSI. In order to reduce the noise in the DSI before fitting, we apply spatial aggregation first. In this paper, we study both fixed and shiftable square windows, as these perform quite well [14], at least in textured areas. (We will deal with untextured areas in Section 5.)

**Collapsing the DSI** Finally, once the local minima in the DSI at all pixels have been adequately modeled, we can collapse the DSI back to an integral sampling of disparities. This step is often not necessary, as many stereo matchers do their optimization at sub-pixel disparities. It does, however, have several potential advantages:

- For optimization algorithms like graph cuts [6] where the computation complexity is proportional to the square of the number of disparity level, this can lead to significant performance improvements.
- Certain symmetric matching algorithm such as dynamic programming and the technique developed in Section 5 require an integral sampling of disparity to establish two-way optima.

To collapse the DSI, we find the lowest matching score within a  $\frac{1}{2}$  disparity from each integral disparity, using the results of the parabolic fitting, if it was used. We also store the relative offset of this minimum from the integral disparity for future processing and for outputting a final high-accuracy disparity map. Alternately, sub-pixel estimates could be recomputed at the end around each winning disparity using one of the techniques described in [17], e.g., using a Lucas-Kanade gradient-based fit [10] to nearby pixels at the same disparity.

## 4 Experimental Evaluation of Matching Costs

Since there are so many alternatives possible for computing the DSI, how do we choose among them? From theoretical arguments, we know that it is better to sample the DSI at fractional disparities and to interpolate the resulting surface when looking for local minima. However, real images have noise and other artifacts such as aliasing and depth discontinuities. So, how do we choose?

In order to answer this question, we apply the techniques introduced in this section to the four test sequences described in [14], which are also available on the Web (<http://www.middlebury.edu/stereo/>). Two of these sequences are shown in Figure 4a. (The other two are omitted due to space limitations.) At this point, we are only interested in the accuracy of these techniques in unoccluded textured areas; techniques for estimating the disparities of the remaining pixels will be presented in the next section.

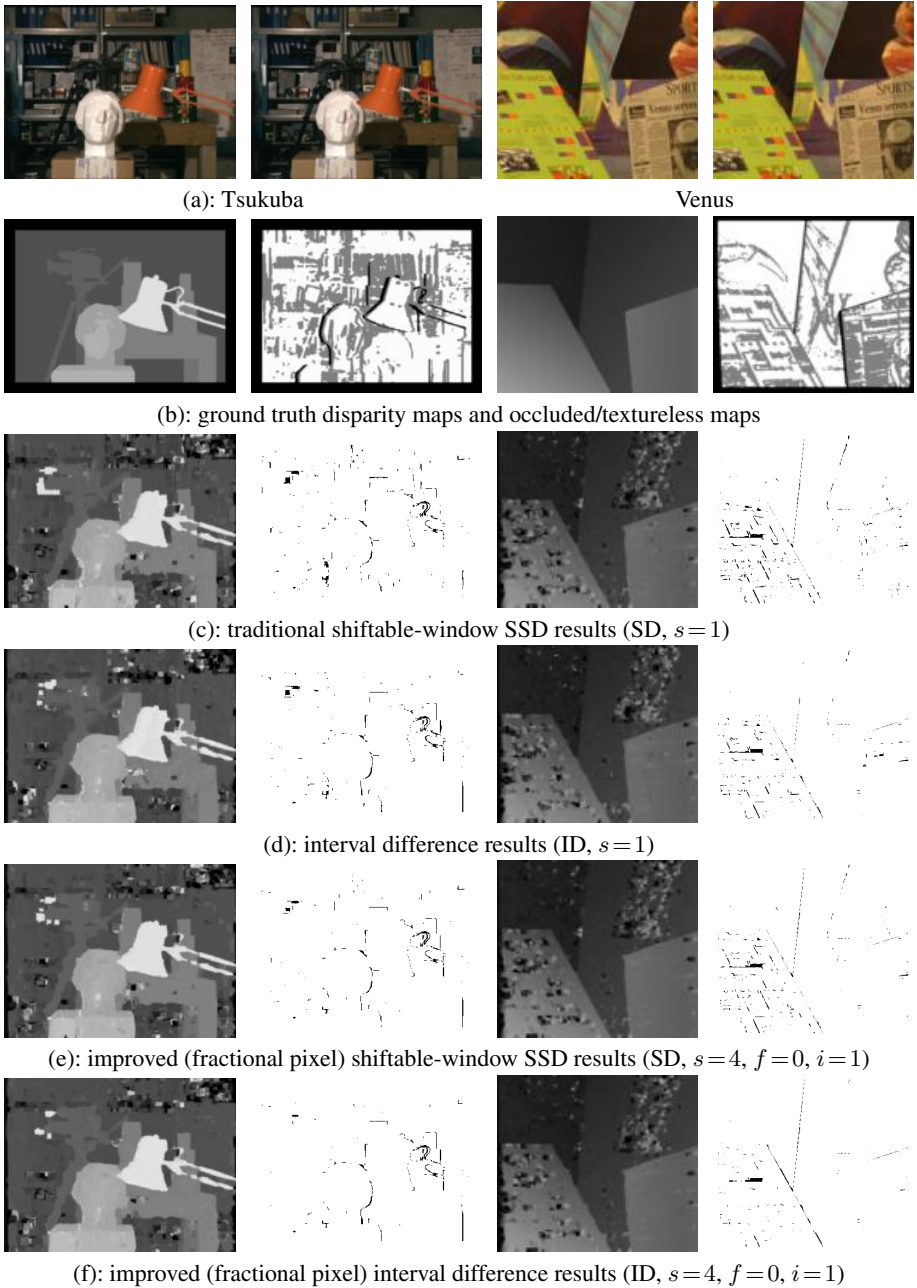
For the analysis in this section, we select textured pixels as follows: compute the squared horizontal gradient at each pixel (averaging the left and right values to remain symmetrical), and average these values in a  $3 \times 3$  neighborhood. Then, threshold the averaged squared value to obtain the textureless masks shown in Figure 4c. We currently use a threshold of 9 gray levels squared.

The parameters that we vary in our experiments are as follows:

- $s = 1, 2, 4$  interpolation rate (inverse of fractional disparity)
- $o = 1, 3$  interpolation order (linear or cubic)
- $i$ : symmetric matching of interpolated scanlines (on or off)
- $d$ : dissimilarity metric (squared differences SD or interval differences ID)
- $f$ : sub-pixel fit of disparities after aggregation (on or off)

The parameters that we hold fixed in the algorithm are the matching criterion (squared differences), the window size ( $7 \times 7$ ), and the fact that the windows are shiftable. We also collapse the DSI to an integer-valued sampling, so that the final winner-take-all is performed on integer disparities (with the stored sub-pixel minimum estimates used to compute the final sub-pixel disparity).

The statistics that we gather are the number of pixels that are “bad” matches, i.e., whose floating point disparity differs from the ground truth by more than 1.5 pixels. (We use 1.5 instead of 1 in order to tolerate small disparity deviations due to possible vertical misregistration.) Table 1 shows the percentage of bad matches for each of the four data sets as a function of our variable parameters. We only show results using cubic interpolation ( $o = 3$ ); linear interpolation gives comparable, but (on average) slightly inferior results.



**Fig. 4.** Test images and associated maps: (a) input images; (b) true disparity maps and occluded (black) and textureless (white) regions (the gray regions are the ones for which we collect statistics); (c) traditional shiftable SSD results (disparity map and error map); (d) interval difference (asymmetric Birchfield-Tomasi dissimilarity); (e) fractional disparities with symmetric matching; (f) fractional disparities with symmetric matching and interval difference.

**Table 1.** Percentage of bad matching pixels for various matching cost options. The numbers highlighted in boldface are the best matching variants for each dataset.

Image	$d$	$s = 1$	$s = 2$				$s = 4$			
		$f :$	0		1		0		1	
		$i :$	0	1	0	1	0	1	0	1
Sawtooth	SD	0.37	0.36	0.30	0.38	0.39	0.35	0.29	0.37	0.42
	ID	0.22	0.20	<b>0.12</b>	0.18	0.16	0.22	0.20	0.25	0.26
Venus	SD	1.33	1.10	1.23	1.21	1.34	<b>1.08</b>	1.19	1.16	1.29
	ID	4.04	1.33	4.68	1.43	4.82	1.26	1.52	1.38	1.65
Tsukuba	SD	4.38	3.69	3.51	3.73	6.84	3.72	3.43	3.77	3.37
	ID	6.54	3.25	<b>3.15</b>	3.62	3.22	3.67	3.17	3.65	3.26
Map	SD	4.72	3.85	3.25	3.84	3.24	3.66	3.20	3.64	3.22
	ID	3.15	5.77	2.38	3.05	2.55	2.92	<b>2.19</b>	2.95	2.28

In Table 1, we have highlighted in boldface the lowest score for each of the four data sets. As one can see, there is no single setting that consistently outperforms the others, although sub-pixel fitting usually leads to slightly worse results, probably because it is sensitive to noise in the DSI. Figure 4 shows the results corresponding to the first ( $s = 1$ ) and seventh ( $s = 4$ ,  $f = 0$ ,  $i = 1$ ) columns. Note how the seventh column consistently outperforms both the original SD (squared difference) and ID (asymmetric interval difference) algorithms. Using interval analysis instead of sub-pixel fitting seems to usually result in lower errors, although it can lead to problems both during min-finding (this Section) and when establishing certain matches (Section 5) because many reasonable matches can yield a matching cost of 0.

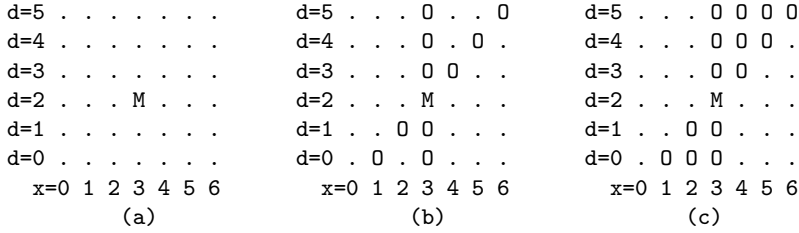
Carefully examining the error maps in Figure 4 shows that the main effects of using the fractional disparity matching scores seems to be getting better results at strong intensity discontinuities. The remaining errors seem to be a combination of the classic “fattening” effect seen near disparity discontinuities (which is characteristic of local analysis), and some errors in repeated textures such as the video tapes in the Tsukuba image, which could be ameliorated with more aggregation or global optimization.

In summary, while there are no clear winners among the different cost variants, it can be seen that symmetric interpolated matching ( $i = 1$  and  $s = 2$  or  $s = 4$ ) usually outperforms traditional, integer-based matching. The benefit of interval matching depends on the winner selection strategy.

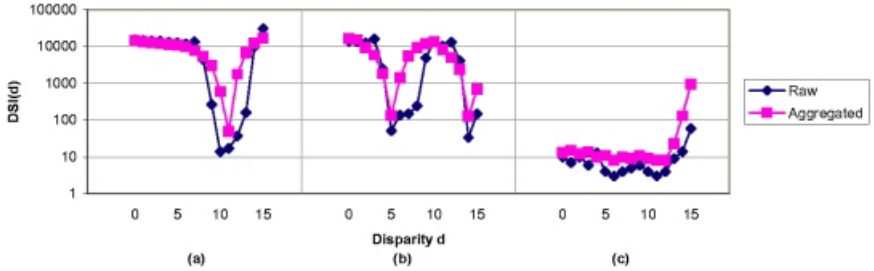
## 5 Symmetric Matching Process

We now turn to the second part of our method: the symmetric matching process. Using an interpolated matching cost that is insensitive to aliasing, we can determine the quality of matches based on the cost distribution in the DSI after a small amount of initial aggregation. Throughout this section we use symmetric matching with  $\frac{1}{2}$  pixel interpolation (SD,  $s = 2$ ,  $f = 0$ ,  $i = 1$ ). We are not using interval matching (ID) since this tends to “round down” good matching costs to 0, making it more difficult to draw conclusions from the cost value distributions.





**Fig. 5.** Illustration of uniqueness and ordering constraints. The figures symbolize part of an  $x$ - $d$  slice through the DSI. (a) A proposed match M for  $x = 3$  and  $d = 2$ . (b) Other matches O ruled out by the uniqueness constraint. The vertical line eliminates other matches for the reference (left) pixel; the diagonal line eliminates other matches for the matching (right) pixel. Many asymmetric algorithms only enforce the former. (c) Some algorithms (not ours) enforce the ordering constraint, and disallow all matches in the two triangular regions.



**Fig. 6.** Example cost value distributions (vertical DSI columns) for three locations in the Tsukuba images. We show both initial (raw) values and after aggregation with a  $5 \times 5$  window. (a) Locally constant image region near the nose of the bust. Aggregation recovers the correct minimum. (b) Repetitive texture (video tapes on shelf), yielding two local minima. (c) Textureless region (shadow under the table), resulting in many low-cost values.

Our algorithm starts by selecting a subset of high-confidence matches, and then aggregates the DSI with increasingly larger windows to disambiguate matches in untextured areas. (A related technique that starts with high-confidence corner matches has been proposed by Zhang and Yang [20].) For easier reasoning about visibility, we collapse the DSI to integer sampling, as discussed in Section 3.

## 5.1 Selecting Certain Matches

There are two basic tests to determine whether a match is *certain*, i.e., correct with high probability. First, there should be a clear minimum among all candidate cost values. Second, the minima for both left-to-right and right-to-left matching should agree [7,5], which can be checked by examining the vertical and diagonal columns in disparity space (Figure 5b).

While the second test is easy to implement, the definition of a clear minimum is less straightforward. Using one of the improved matching costs developed in the previous

section ensures that a correct match yields a low matching cost, even in high-frequency image regions. Thus, a minimum corresponding to a correct match should have a low cost value. Conversely, a large minimum value indicates an occluded pixel that has no correct match. At many pixels, however, there will be multiple low cost values that cannot always be disambiguated. Figure 6 illustrates three different cost distributions, which help motivate the following definition:

We define a match  $(x, y, d)$  with cost  $C = DSI(x, y, d)$  to be *certain* if

1.  $C$  is the minimum cost in both columns, i.e.,

$$\begin{aligned} C &\leq DSI(x, y, d') && \forall d' \neq d, \text{ and} \\ C &\leq DSI(x - d + d', y, d') && \forall d' \neq d; \end{aligned} \quad (11)$$

2.  $C$  is a *strong* minimum in at least one column, i.e.,

$$\begin{aligned} C &\leq m DSI(x, y, d') && \forall d' \neq d, \text{ or} \\ C &\leq m DSI(x - d + d', y, d') && \forall d' \neq d \end{aligned} \quad (12)$$

for a given *winner margin*  $m < 1$ .

The winner margin  $m$  needs to be chosen low enough to avoid false positives (i.e., incorrect matches labeled certain). On the other hand, a higher margin results in a higher fraction of pixels being matched. Table 2a demonstrates this trade-off using the Tsukuba images from Figure 4. It shows the error rate among the certain matches and the total fraction of pixels matched as a function of winner margin  $m$ .

For lowest error rates, a small amount of aggregation is necessary. Here we aggregate the initial cost values with a  $5 \times 5$  window. Note that unlike in Section 4, we do not need to explicitly label pixels as textureless, since this is subsumed by our test for match certainty.

## 5.2 Reasoning about Visibility and Occlusions

Before discussing how we can propagate certain matches to ambiguous regions, we briefly address how visibility constraints enter the matching process. Since we eventually need to assign matches even where there is no clear minimum among the cost values, we need to ensure that the uniqueness constraint is enforced. We can achieve this by altering the cost values “shadowed” by certain matches. Every time a new certain match has been assigned, we set the cost values for other matches eliminated by the new match (i.e., the O’s in Figure 5b) to a large value  $C_{\max}$ . This prohibits the future assignment of low-cost matches in the diagonal DSI column. Altering the cost values can also help disambiguate multiple good matches, especially on the perimeter of textureless regions.

After the certain matches have been found and the costs have been altered as described, new certain matches may emerge where competing low cost values have been changed to  $C_{\max}$ . The process is thus repeated until no new certain matches can be found. Typically, this process yields an increase in certain matches of about 5–10%.

We have also experimented with enforcing the ordering constraint by assigning  $C_{\max}$  to all O’s in Figure 5c. We found, however, that this yields too few additional

**Table 2.** (a) Percent of bad certain matches (disparity error  $> 1$ ) and fraction of pixels matched as a function of winner margin  $m$  for the Tsukuba image pair. Lower margins result in fewer errors but leave more pixels unmatched. (b) Performance of our matching algorithm while aggregating with increasing window size  $w$  (see Section 5.3 and Figure 7) using a constant margin  $m = 0.5$ . The last row shows the percentage of bad pixels in unoccluded regions after the remaining unmatched regions have been filled in.

Margin $m$	Bad	Matched	Final $w$	Pass	Bad	Matched
1.0	12.1%	96%	5	1	2.8%	59%
0.9	9.5%	90%	9	2	3.8%	83%
0.8	6.0%	81%	13	3	4.0%	90%
0.7	4.0%	73%	17	4	4.0%	91%
0.6	3.2%	66%	21	5	4.0%	91%
0.5	2.8%	59%				
0.4	2.4%	53%	21	5*	4.9%	100%
0.3	2.1%	45%				
(a)			(b)			

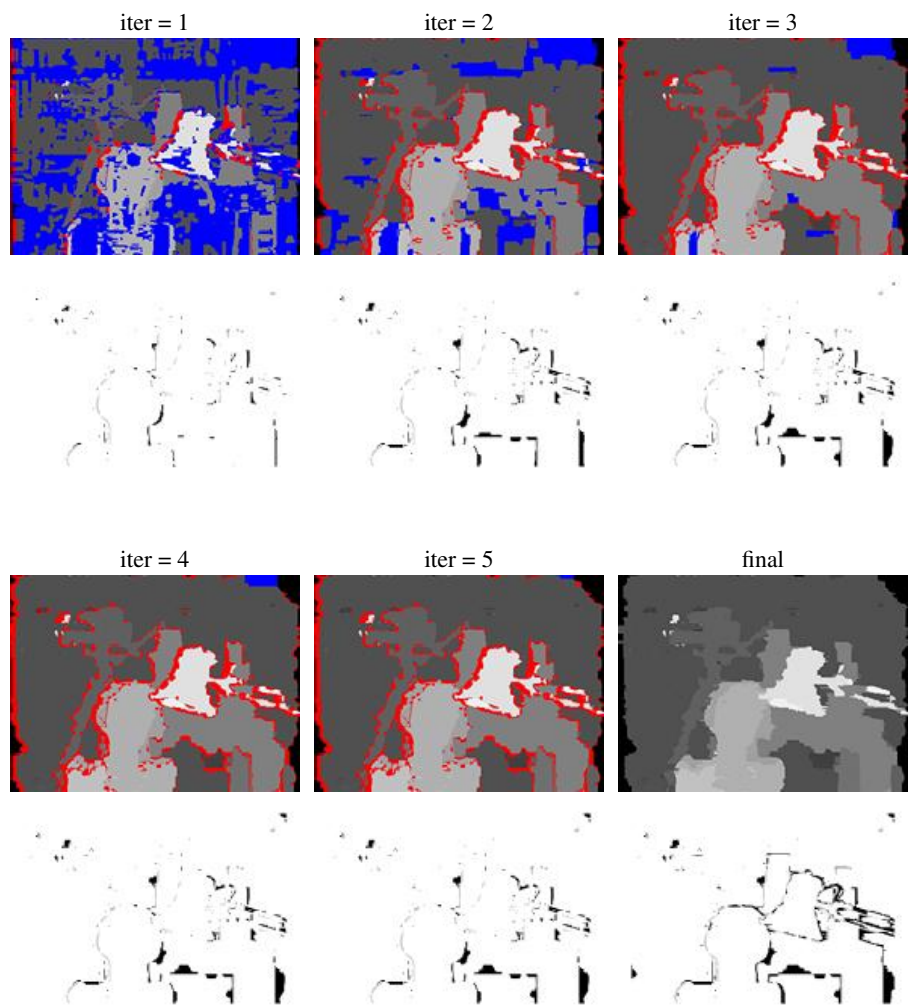
certain matches to warrant the inability to deal with narrow occluding objects imposed by the ordering constraint.

The second advantage of eliminating matches by setting their costs to  $C_{\max}$  is that occluded pixels can be detected more easily. We currently label all pixels as occluded when their minimum cost value is more than 10 times the average cost value of all certain matches. This works very well in textured areas with many certain matches, whose  $C_{\max}$  values effectively rule out *all* disparities for pixels that must be occluded. The disparities of occluded areas can be estimated fairly reliably by filling them with the nearest background disparity on each scanline. This is the last stage in our matching algorithm. First, however, the disparities in ambiguous regions need to be recovered, which is discussed next.

### 5.3 Propagating Matches

To propagate certain matches into ambiguous matching regions, we propose an iterative algorithm similar in spirit to adaptive-windows techniques [8,13,20]. Our algorithm interleaves the selection of certain matches with further aggregation of the DSI, using successively larger windows. After each aggregation step, new certain matches are selected among the previously uncommitted pixels. The algorithm converges quickly because the large cost values  $C_{\max}$  are aggregated as well, which helps to rule out many ambiguous matches.

Figure 7 shows the results of our algorithm on the Tsukuba image pair. We start by aggregating with a  $5 \times 5$  window, and then increase the window size by 4 after each iteration. We use a constant winner margin  $m = 0.5$ . After 5 iterations, disparities have been assigned to virtually all uncertain regions. The remaining occluded (high-cost) regions are filled in as described in the previous section. Finally, we restore the sub-pixel estimates computed before collapsing the DSI to integer disparities.



**Fig. 7.** Resolving unmatched areas using increasing amounts of aggregation. The figure shows a sequence of six (color) disparity maps as the matching process aggregates the DSI with windows of size 5, 9, 13, 17, and 21, and selects certain matches after each pass. Uncertain matches are shown in blue; high-cost (occluded) matches are shown in red. After the 5th pass, the remaining unmatched areas are filled in. Underneath each disparity map is the corresponding disparity error map (for certain matches only). Table 2b lists the statistics for each of the six disparity maps.

Table 2b lists the statistics for each of the five iterations. Note that the number of bad matched pixels increases only slightly, and the final numbers are quite good. The overall performance (4.9% bad unoccluded pixels) is comparable to most methods evaluated in [14], except for the graph-cut method [6]. The overall running time for this experiment is 4.7 seconds on a 750 MHz Pentium III machine.

## 6 Conclusion

In this paper we have presented both novel matching costs and a new symmetric matching algorithm. Our matching costs are based on interpolated image signals, and are motivated by a frequency analysis of the continuous disparity space image (DSI). We have explored several symmetric cost variants, including a generalized version of Birchfield and Tomasi's matching criterion [3]. While there is no clear winner among the different variants, we have shown that our new costs result in improved matching performance, in particular in high-frequency image regions. An added benefit is that the sub-pixel information derived during the initial cost computation can be restored at the end for the winning disparities, even if the intermediate matching process operates on integer disparities.

Our second contribution, the symmetric matching algorithm, utilizes visibility constraints to find an initial set of high-confidence matches, and then propagates disparity estimates into ambiguous image regions using successive aggregation of the DSI. Our initial experiments show competitive performance for a method that does not perform global optimization.

There are several issues we plan to address in future work. Relating to matching costs, we have started to explore the effect of certain asymmetries that occur when collapsing the subsampled DSI to an integer grid. We also want to evaluate our matching costs with further experiments, and compare them with the method developed by Shimizu and Okutomi [15]. Relating to disparity estimation, we plan to improve our current algorithm and to test its performance on image pairs with narrow occluding objects that violate the ordering constraint. In the longer term, we hope to achieve results competitive with the graph-cut algorithm [6] by exploring new ways of imposing global smoothness constraints in textureless areas.

## References

1. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.
2. P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996.
3. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.
4. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Seventh International Conference on Computer Vision (ICCV'99)*, pages 489–495, Kerkira, Greece, September 1999.
5. A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, September 1999.
6. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.
7. P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, Winter 1993.

8. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.
9. C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, volume I, pages 125–131, Fort Collins, June 1999.
10. B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679, Vancouver, 1981.
11. L. H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.
12. M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
13. D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, July 1998.
14. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.
15. M. Shimizu and M. Okutomi. Precise sub-pixel estimation on area-based matching. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 90–97, Vancouver, Canada, July 2001.
16. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 532–539, Vancouver, Canada, July 2001.
17. Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986.
18. Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of CCD imaging process. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 480–487, Vancouver, Canada, July 2001.
19. Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 274–279, New York, New York, June 1993. IEEE Computer Society.
20. Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In M. Pollefeys et al., editors, *Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE 2000)*, pages 68–85, Dublin, Ireland, July 2000.

# New Techniques for Automated Architectural Reconstruction from Photographs

Tomas Werner and Andrew Zisserman

Robotics Research Group  
Department of Engineering Science  
University of Oxford  
Oxford, OX1 3PJ  
`{werner,az}@robots.ox.ac.uk`

**Abstract.** We investigate a strategy for reconstructing of buildings from multiple (uncalibrated) images. In a similar manner to the Facade approach we first generate a coarse piecewise planar model of the principal scene planes and their delineations, and then use these facets to guide the search for indentations and protrusions such as windows and doors. However, unlike the Facade approach which involves manual selection and alignment of the geometric primitives, the strategy here is fully automatic.

There are several points of novelty: first we demonstrate that the use of quite generic models together with particular scene constraints (the availability of several principal directions) is sufficiently powerful to enable successful reconstruction of the targeted scenes. Second, we develop and refine a technique for piecewise planar model fitting involving sweeping polygonal primitives, and assess the performance of this technique. Third, lines at infinity are constructed from image correspondences and used to sweep planes in the principal directions.

The strategy is illustrated on several image triplets of College buildings. It is demonstrated that convincing texture mapped models are generated which include the main walls and roofs, together with inset windows and also protruding (dormer) roof windows.

## 1 Introduction

The objective of this work is the automated construction of 3D texture mapped models of buildings from a number of close-range photographs of the scene. Much research into automation in the photogrammetry community has concentrated on reconstruction from aerial, i.e. long-range, views (e.g. [3,5,9,15,19,20]). For close-range photographs there are several manual systems available (e.g. Facade [27], Canoma [1], PhotoModeller [2]) which may be used to produce excellent piecewise polyhedral and piecewise quadric scene models of buildings.

Automated close-range reconstruction of buildings [11,12,21,26], is less thoroughly explored than long-range. Two recent papers represent well the spectrum of approaches currently available. Dick *et al* [12] cast the problem as model based recognition, and attempt to fit very strong models using a Bayesian framework. The models are highly tuned to the scene (e.g. a gothic window for a chapel)

and the strong priors (e.g. on height and width) considerably narrow down the search. The method recognises these models in a single image, and other images may then be used to verify these recognition hypotheses. At the other end of the spectrum Tao *et al* [26] cast the problem as that of fitting piecewise planar patches over multiple views. The patches are determined by (over-) segmenting a reference image using colour, and subsequently larger regions are assembled by merging 3D patches. In contrast to [12] the method cannot proceed if only a single view is available as it is completely rooted in multiple view stereo. Both methods produce excellent results.

In this paper we explore a strategy that is located in the ‘middle’ of this spectrum: we fit quite generic models (planes, polyhedra). These are weaker than the scene specific models of [12] but stronger than the planar patches employed in [26]. Our approach is inspired by that of Facade [27] and proceeds in two stages: first, a coarse piecewise planar model of the principal scene planes and their delineations is generated; and second, this coarse model is used to guide the search for fitting more refined polyhedral models of the indentations and protrusions from the coarse model, such as windows and doors.

The method is targeted on architectural scenes which typically contain planes orientated in three dominant directions which are perpendicular to each other, for example the vertical sides of a building and the horizontal ground plane. The strategy assumes that the scene contains three such principal directions and that the images contain sufficient information to obtain the vanishing points of these directions.

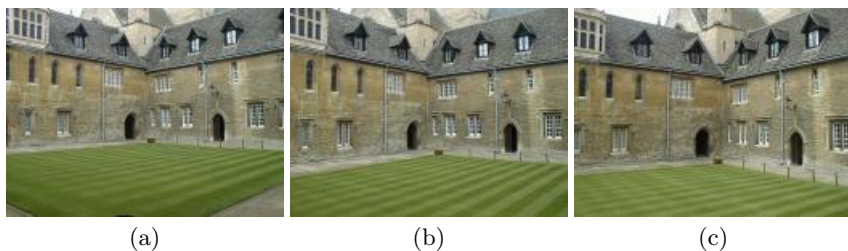
At the technological level we considerably extend the one parameter ‘plane-sweep’ method that has been used by several previous researchers for plane fitting [4,8,11,16]. The extensions are of two types: previously the method was employed with calibrated cameras with known external orientation. Here we are in an uncalibrated setting, and the directions for the sweep are computed from vanishing points in the images. The second extension is to sweep not only planes, but also polyhedral models.

## 2 Preliminaries

In this section we describe the stages of camera computation, vanishing point detection, image point and line matching, and the estimation of the 3D counterparts of the points and lines. This is the necessary starting point in preparation for the ‘clothing’ of the scene by planar primitives. These stages do contain novel elements, but they are not the main issue of this paper so will only be described briefly in the following paragraphs. They will be illustrated for the image triplet shown in figure 1.

**Projective Reconstruction:** Interest points are matched between the image triplet and a trifocal tensor estimated using standard robust methods (see [17] for full details and citations). The result is a camera corresponding to each image and a set of 3D points, defined up to an unknown projective transformation of

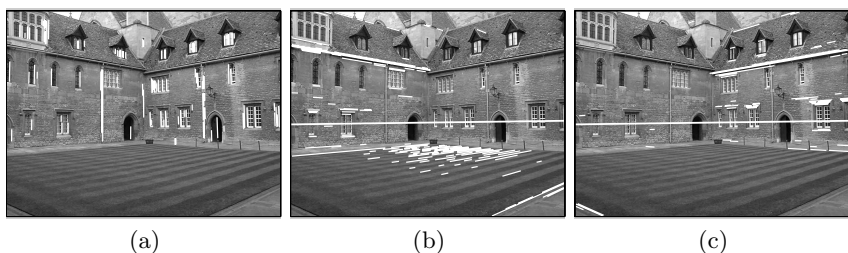




**Fig. 1.** Three images of a Merton College court acquired with a hand held low cost Olympus digital camera. The image size is  $1024 \times 768$  pixels.

3-space. The projective reconstruction computed from the images of figure 1 has 684 3D points, which after bundle adjustment achieve a RMS reprojection accuracy of 0.14 pixels, with a maximum error of 1.01 pixels.

**Vanishing points:** Much previous work has been devoted to automatic vanishing point detection (e.g. [6,7,10,18,22,23,25,28]), and we employ a standard method where lines are computed independently in each image, and the vanishing points corresponding to the three principal directions estimated using a RANSAC method. The typical line sets used to compute the vanishing points are shown in figure 2. The vanishing points are then matched across the images by a simple combinatorial algorithm. The principal direction (which is a point in 3D, the pre-image of a vanishing point) is estimated by minimising the reprojection residuals from the lines supporting each vanishing point.



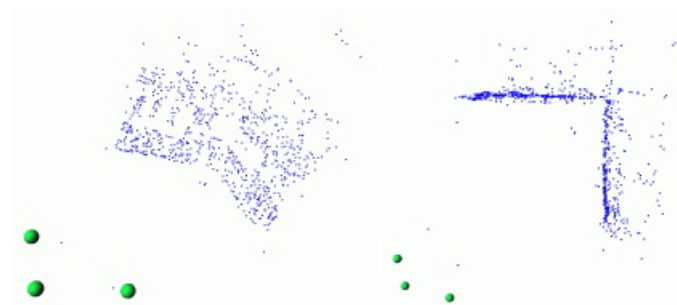
**Fig. 2.** Vanishing point computation. (a), (b) and (c) shows the lines (in white) supporting the vanishing points corresponding to the three principal directions for the first image of figure 1. The thick line in (b) and (c) is the horizon computed as the line through the vanishing points.

These principal directions (which are points at infinity) then play an important role throughout the reconstruction strategy. For example, in determining a vertical facade plane by a RANSAC search on 3D point triplets, the directions

provide two of the points with only one point required to be sampled from the matched scene points. This is described in section 3.

**Metric rectification:** The approach used here to upgrade the projective reconstruction to metric is based on two metric constraints: the three principal and orthogonal scene directions; and that the cameras have square pixels. The method proceeds in three steps. First, the three principal directions (points at infinity) are computed from their images (the vanishing points). This determines the plane at infinity as the plane defined by the three directions, and consequently the projective reconstruction is upgraded to affine. Second, the principal directions are forced to be mutually orthogonal. This determines a reconstruction which differs from metric only by a scaling in each of the principal directions. In the last step, these scalings are determined (up to an overall scale) using the constraint that the camera pixels are square by linearly minimizing an algebraic residual.

This linear, non-iterative algorithm yields a very good initial estimate of the metric cameras, i.e. both the internal and exterior orientation. For example the computed internal parameters for the first image are: principal point (580.5, 349.0), aspect ratio 1.00055, angle between image axes  $89.8944^\circ$ , and focal length 1085.4 pixels. Their further improvement is possible by a bundle adjustment constrained by orthogonality and square pixels assumptions. The reconstructed 3D point cloud after the metric rectification is shown in figure 3.

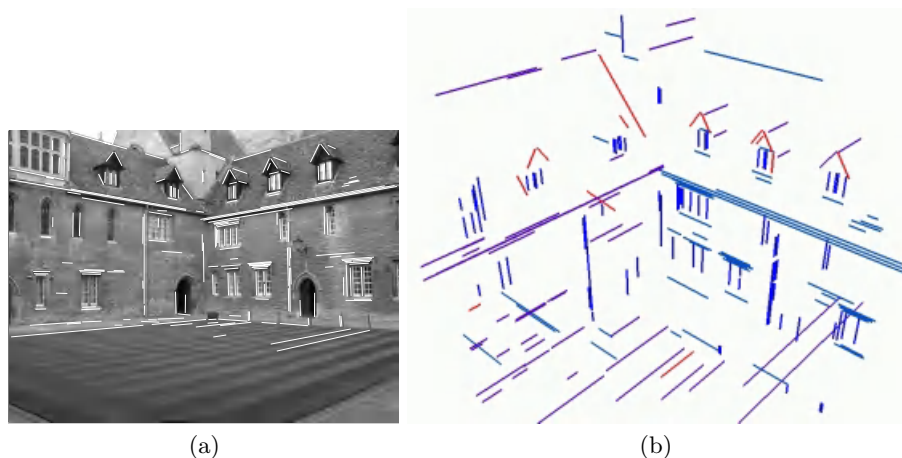


**Fig. 3.** Two views of the cloud of 3D points after metric rectification. The spheres depict the camera centers.

**Line matching:** We use a local implementation of the line matching method of Schmid and Zisserman [24]. Line segments are detected independently in each image, and then matched across the triplet using a combination of photometric constraints and trifocal geometry. In total 434 line segments are matched. Special attention is paid to horizontal lines as these tend to lie in epipolar planes for

typical acquisition geometries. The resulting ambiguity is resolved by a search to register the photometric neighbourhood – a ‘line sweep’.

We then carry out another stage of processing where the lines are classified into disjoint direction sets and re-estimated with this additional constraint. This is crucial for (i) accuracy of the estimated 3D lines, and (ii) organizing the subsequent search for higher level building primitives (planes, polygons). In more detail, from the intersection point of the line with the plane at infinity, the lines are labeled (using  $\chi^2$  statistics on reprojection residuals) as belonging to one of seven classes: principal direction (3), principal line (3), other (1). For example if a line is vertical, then it intersects the plane at infinity at the vertical direction. When it is re-estimated under this (hard) constraint, the minimization of reprojection error is only over the two remaining parameters of the line. Figure 4 shows the resulting matched lines in 3D.



**Fig. 4.** Reconstructed 3D lines, shown reprojected onto the first image (a) and in 3D (b). Of 193 matched lines, there are 7 mismatches (mostly due to repeated structure on the lawn). Of the rest, 168 lines go through a principal direction (e.g., vertical), 19 go through a principal line (e.g., are parallel to a wall), and none is unconstrained.

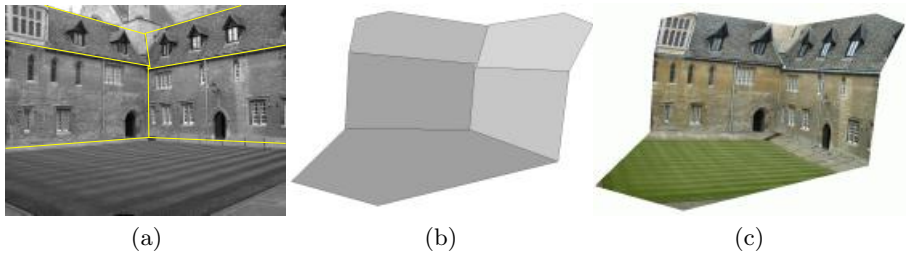
### 3 Coarse Model Fitting

The objective of this section is to automatically construct a coarse polyhedral model of the scene, given the cameras, 3D points, directions and lines, whose computation was described in the previous section. Numerous methods have been given for plane and polyhedral fitting and we employ these with the addition of direction constraints.

In overview we organize the search in the following stages: first we search for planes intersecting two principal directions (typically walls). The intersections of these planes give a partial delineation, and visibility is then used to determine

which parts of the delineation are included. Second, we search for planes incident with the selected wall planes and intersecting one principal direction (roofs). The vertical walls are consequently completely delineated: above by the roof plane, below by the ground plane, and laterally by other planes or the image boundary.

To give an example of a plane search, suppose we are searching for the vertical wall. Two types of methods are employed. One is the standard RANSAC [13] robust estimation. The novelty here is that samples are not simply based on three random points, but on one of the following particular samples: one point and a 3D line of the appropriate class (in this case not oblique); one point and the appropriate line at infinity (the one corresponding to vertical planes); two points and the appropriate point at infinity (e.g. the vertical direction). This ensures that only planes of the required type are estimated, and has a considerable advantage in reducing complexity – e.g. one finite 3D point match is sufficient to instantiate a plane. The second type of method is plane sweeping which is described in more detail below. Both types of method are attempted and the result of the most successful is used. The other planes (e.g. the roof planes) are searched for in a similar manner.

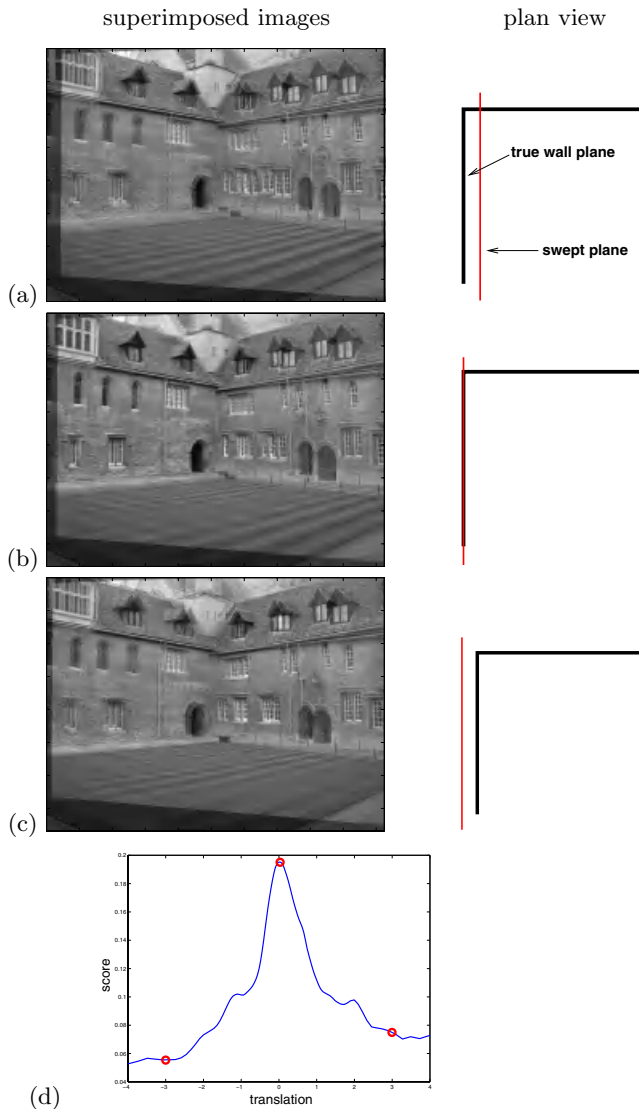


**Fig. 5.** Coarse model: wire frame projected to image 1 (a), shaded (b) and texture mapped (c) 3D model.

In most cases to date finding the principal vertical planes is straightforward. The ground plane often contains fewer point and line matches, and is found by plane sweeping. Once the vertical and ground planes are known, the search for roof planes is simpler because much of the erroneous data can now be ignored. The coarse model is shown in figure 5.

### 3.1 Plane Sweeping

The objective here is to determine scene planes by a one parameter search (the sweep) of a virtual plane. The support for the virtual plane as it is swept is measured using the planar homography induced by the plane to define a point-to-point mapping between the images, and aggregating cross-correlation scores directly from the images. The novelty of the application here is that we define the sweep about a line at infinity. A line determines a plane up to one parameter,



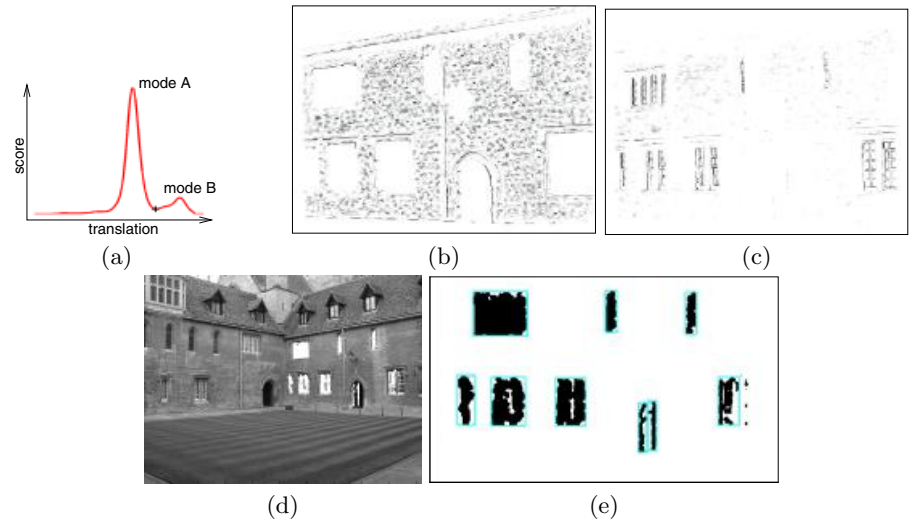
**Fig. 6.** Plane sweep under translations along a principal scene direction. (a,b,c) show images in figure 1a and 1b superimposed by a homography map corresponding to a translating a virtual scene plane. This scene plane is parallel with the left wall. The right-hand figures illustrates position of the wall and swept plane from a plan view. (d) shows a plot of the score function against translation. The circles correspond to the translations in (a,b,c), respectively. The middle translation is the one which best registers the planes, and this is visible in (b) where the plane of interest is most “focussed”.

and in this case the parameter defines a translation of the plane since planes through a line at infinity are parallel.

For example, to search for the ground plane, the horizontal line at infinity is used, and the resulting sweep is in the vertical direction across the scene. When the swept plane corresponds to the actual ground plane, then image regions mapped by the homography will be highly correlated in the region corresponding to the ground. By determining the swept plane which produces the highest correlation (measured by an appropriate robust score function described below) the true position of the ground plane is determined. The sweeping is illustrated for one of the principal directions in figure 6.

Plane sweeping is able to determine scene planes even in the absence of any point or line correspondences for that plane (as are required in RANSAC fitting).

The cross-correlation is measured as follows: first image points at significant image gradients are determined (e.g., [14]); second, a sample of these points are selected as the key points at which to measure the cross-correlation. Only these points are used in order to improve the signal to noise ratio by discarding uninformative points in homogeneous image regions. Approximately 10% of image pixels are selected as the keypoints. At each of the key points the cross-correlation is



**Fig. 7.** (a) Aggregated similarity score in the neighbourhood of the wall plane computed over three views. Mode A corresponds to the wall plane, and B to the plane of the windows. The magnitude of the similarity scores for individual key points are represented by pixel darkness in (b) for the swept plane at the position of mode A, and (c) for the position of mode B. The key points belonging mostly to the wall plane score highly (are dark) in (b) and those belonging to the window plane score highly in (c). Points belonging to windows, (d), are obtained by thresholding depths at the value denoted by the cross in plot (a). (e) Rectangles robustly fitted to clustered points shown on the rectified facade.

computed between the point’s neighbourhood and the corresponding neighbourhood in the other image mapped by the homography induced by the sweeping plane. A neighbourhood size of  $7 \times 7$  pixels is used here. The cross-correlation is computed between all pairs of images, and the result averaged. A score function consists of the cross-correlations averaged over all points as a function of plane translation. Typical score functions are shown in figure 6.

**Performance assessment.** In practice plane sweeping is a very powerful fitting method, and we may ask why it is so successful. There are two main reasons: (i) it correctly models and corrects for distortion of the correlation region between the image, in a similar manner to wide baseline stereo matching (*cf.* stereo rectification which doesn’t correct) – this is the reason that the matching of individual key points is so precise (see figure 7b and c), and (ii) the scoring function aggregates over a large number of points in order to estimate depth (*cf.* traditional stereo where the depth of each point is estimated independently).

## 4 Refinement

In this section we refine the coarse polyhedral model computed in the previous section by attempting to fit two types of polyhedral model to account for deviations from the plane. The models are a rectangular block (this fits to doors and windows, described in section 4.1), and a wedge block (this fits to dormer windows (in roofs) and wall protrusions, described in section 4.2).

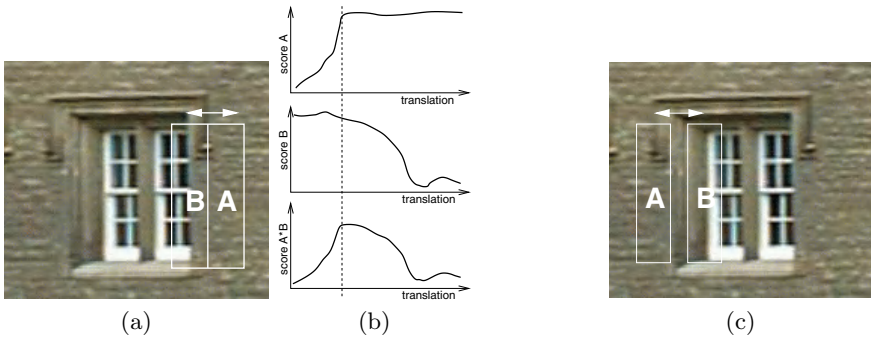
These models are quite generic, and in contrast to the approach of [12] there is no use of strong priors to define (for example) their aspect ratio. Fitting is initiated in regions on either side of the fitted plane where there is evidence of a perturbation from the plane. All models are fitted by variations on sweeping as described below.

### 4.1 Fitting Rectangular Block Models

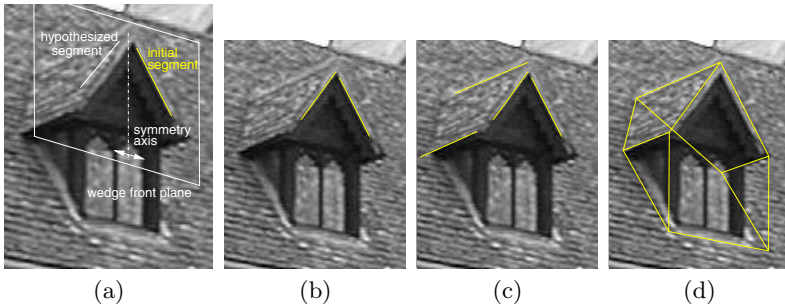
The idea is to determine regions of the plane which do not coincide with the coarsely fitted scene plane, and then to model these regions as rectangles aligned with the principal scene directions. Note, each region is modelled independently and from depth information alone.

Points which lie behind the fitted plane (indentations) are determined by thresholding depths of individual keypoints. The threshold value is obtained from the score function recomputed for the image region corresponding to the current facade, see figure 7a. Two modes are clearly discernable in the function – one corresponding to the coarse facade plane, and the other to the window plane.

The keypoints labeled by thresholding as being behind the wall are shown in figure 7d. Contiguous regions (corresponding to each of the windows and doors) are then computed by robustly clustering these points. The fitted plane has essentially simplified this task to that of clustering a set of pixels in a 2D



**Fig. 8.** Refining the position of the vertical window boundaries. (a) Search regions consisting of two adjacent rectangles which are translated horizontally. In rectangle A the similarity score is based on the wall plane. In rectangle B the window plane is used. (b) plots the two scores and their product, which peaks at the actual window boundary location (the dashed line). (c) For non-occluding window edges (the left one in this case), a gap is inserted between the rectangles. The width of the gap is the distance between the wall and window planes projected to the image.



**Fig. 9.** Fitting a wedge model (to reconstruct a dormer window). (a) Fitting is initiated if a single oblique 3D line segment (depicted in yellow) is detected. A symmetry related line segment is determined by searching for intensity gradient maximum, the gradient is summed along the hypothesized line segment. (b) The detected symmetric segment. (c) The top and lower line segments are determined by a similar search. (d) The model is then completed in a straightforward manner, using the known roof plane.

image as the analysis can be carried out on a rectified version of the image (where the principal scene directions are orthogonal in the image). Standard image processing methods, such as simple operations of binary morphology, are used. Rectangular boxes are then fitted to the clusters. The resulting windows boundaries are shown in figure 7e.

The window boundaries are then refined using a further correlation based search, but now concentrated in the vicinity of the putative window boundary. For each boundary a one-dimensional search is carried out perpendicular to that boundary. For example for a vertical boundary the search is horizontal. Two score functions are computed, one based on the homography induced by the



wall plane, the other based on the homography induced by the window plane, as illustrated in figure 8. For the wall plane the score is high when the pixels in the rectangle belong to the wall and small otherwise. Conversely, for the window plane homography the similarity score is high for window pixels, and low for wall ones. The product of these two scores peaks at the actual window boundary, as shown in the figure. In essence this is another sweep-like fitting of a model.

## 4.2 Fitting Wedge Models

Wedge shaped protrusions are frequently found in architecture, for example they might form part of the roof or a bay windows. We illustrate here how a wedge model is instantiated and fitted automatically in order to reconstruct dormer windows in the vicinity of a roof plane.

The method is demonstrated in figure 9. Initially, all reconstructed scene lines that intersect exactly one principal line are considered as belonging to a potential wedge shape. The principal line of the segment defines the wedge front plane. If the other wedge edges have been matched and are available in 3D, finishing the wedge is straightforward. However, usually these lines are not available – for example in figure 1 many dormer edges are not even detected by the Canny edge detector due to the complicated roof texture. In this case a search for a line segment symmetrical to the initial segment is performed. The search is parametrized by a single parameter (the horizontal position of the symmetry axis) and the true location maximizes the intensity gradient along the segment. In a similar manner, the remaining wedge edges are localized. All dormer windows in figure 1 are successfully reconstructed.

## 4.3 Results

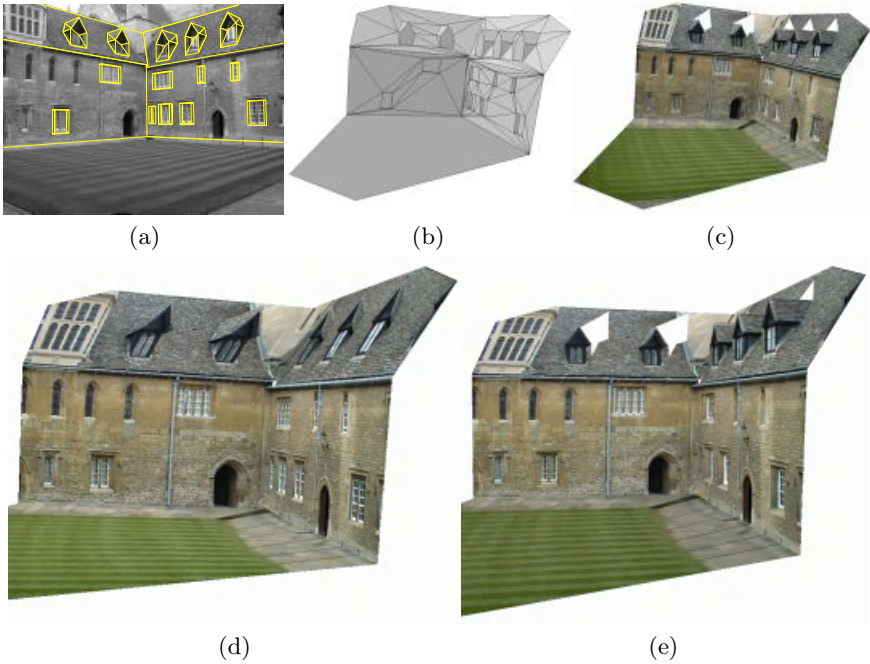
The facade model updated with models found in sections 4.1 and 4.2, and texture mapped from the appropriate images is shown in figure 10.

The results of applying the same algorithms (coarse then refinements) to an another image triplet are shown in figure 11. In this case, a fitting algorithm similar to one described in Section 4.2 automatically finds the protrusion in the left wall. The algorithm works on the same principle, but uses a different protrusion model. It is instantiated by the horizontal oblique edge at the top of the protrusion. The dimensions of the protrusion are then found by sweeping hypothesized edges in directions determined by the initial edge and the principal directions. High gradient along the swept edge is used as evidence for the true edge. Again, the classification of principal directions is crucial here.

## 5 Discussion

In this work we have systematically explored a particular strategy for scene reconstruction.

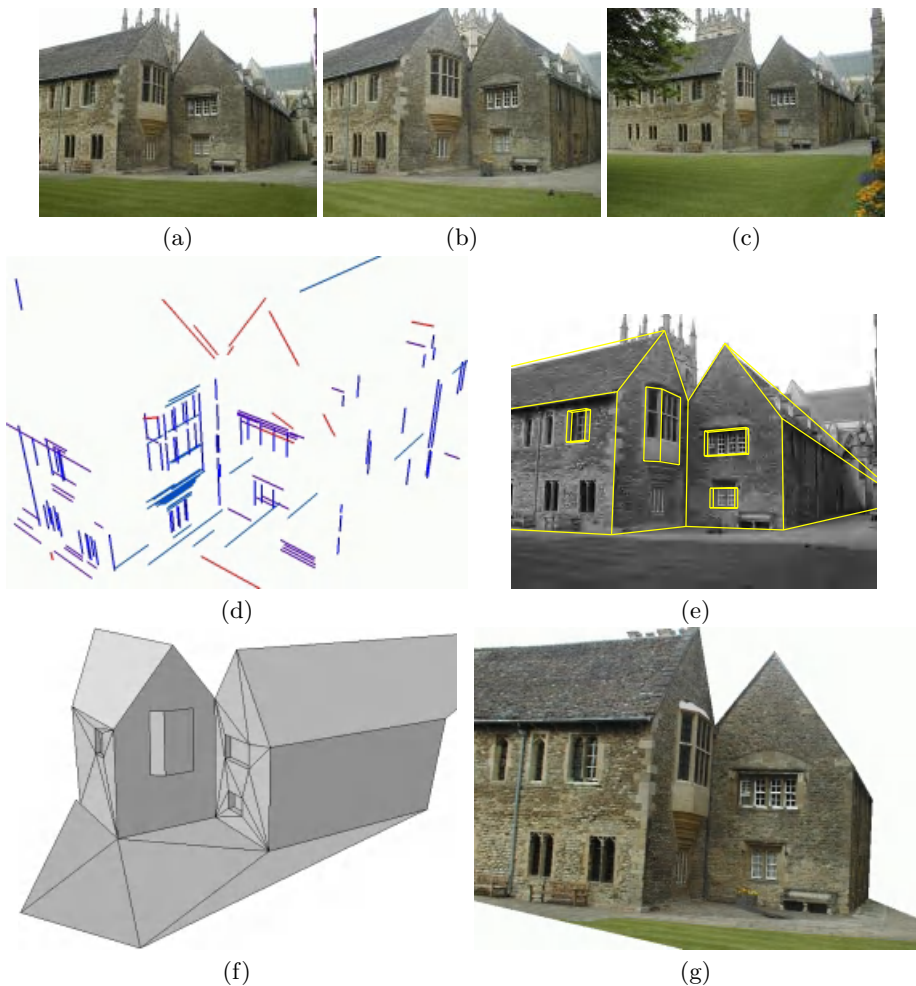
Clearly a strategy of this type is not applicable to non-urban scenes (trees, mountains) or cluttered interiors (rooms with tables, table legs, piles of books).



**Fig. 10.** The automatically computed 3D model augmented with indented windows in the vertical planes, and dormer windows in the roof. (a) wire frame projected onto the image, (b) shaded wireframe, (c) textured mapped 3D model. Compare the same view of the coarse model (d) and the augmented final model (e) – explicit modelling of dormer windows and indented windows on the walls is clearly an improvement. Note, in textured models the occluded parts of the roof and wall planes (e.g. behind the dormer windows) are removed and appear as white.

A more interesting question is where it will fail in its targetted domain of architectural scenes. By and large, and for obvious reasons, many buildings have many horizontal and vertical parts (bricks, edges, windows, doors) so that a vertical direction and line at infinity for horizontal plane may be estimated from images. The weakest point of the current strategy is that two orthogonal horizontal directions are not always available in the scene, e.g., on an octagonal building, or a circular church. The lack of these directions affects the strategy at several points. Fortunately, alternatives are available and these are currently being incorporated so that the strategy is not so limited. For example, when plane sweeping for perturbations the direction of the sweep can simply be chosen as perpendicular to the planes of the coarse geometric model.

One of the important contributions of this work is the development of fitting geometric models by sweeping, and three new techniques are described in the previous sections, namely sweeping scene planes about a line at infinity (used for obtaining or refining positions of the main walls in section 3.1), correlation



**Fig. 11.** (a,b,c) Input images of another court of Merton College, Oxford, acquired with the same camera as figure 1. Reconstruction results: (d) reconstructed and classified line segments, (e) model wire frame projected onto image 1, (f,g) final model with indented windows and protrusion reconstructed.

based search for building edges using translating rectangles and inter image homographies (used for refining window boundaries in section 4.1), and search for local gradient maxima along translating line segments (used for dormer window reconstruction in section 4.2). They have much in common: (i) A single unknown parameter, making the search fast and simple. (ii) Evidence is aggregated over relatively large region of interest, enabling features to be found that would not be detected by local operators such as Canny edge detectors or correlation based stereo on individual pixels. (iii) Results of the previous reconstruction stages

allow sweep algorithms to focus only on relevant image parts, achieving a high signal to noise ratio.

**Acknowledgements.** We are very grateful to Frederick Schaffalitzky for computing the projective reconstructions for the triplets. Funding was provided by a Marie Curie fellowship and EC project Vibes.

## References

1. <http://www.canoma.org>.
2. <http://www.photomodeler.org>.
3. C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5*, pages 69–80, September 1999.
4. C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 559–565, June 1999.
5. F. Bignone, O. Henricsson, P. Fua, and M. Stricker. Automatic extraction of generic house roofs from high resolution aerial imagery. In *Proc. 4th European Conference on Computer Vision, Cambridge*, pages 85–96, 1996.
6. B. Brillault-O’Mahony. New method for vanishing point detection. *CVGIP: Image Understanding*, 54(2):289–300, 1991.
7. C. Coelho, M. Straforini, and M. Campani. Using geometrical rules and a priori knowledge for the understanding of indoor scenes. In *Proc. British Machine Vision Conference*, pages 229–234, 1990.
8. R. T. Collins. A space-sweep approach to true multi-image matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.
9. R. T. Collins, C.O. Jaynes, , Y-Q Cheng, X. Wang, F. Stolle, E. M. Riseman, and A. R. Hanson. The ascender system: Automated site modeling from multiple images. *Computer Vision and Image Understanding*, 72(2):143–162, 1998.
10. R. T. Collins and R. S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Proc. 3rd International Conference on Computer Vision, Osaka*, pages 400–403, December 1990.
11. S. Coorg and S. Teller. Extracting textured vertical facades from controlled close-range imagery. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado*, pages 625–632, 1999.
12. A. R. Dick, P. H. S. Torr, Ruffe S. J., and R. Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Int. Conf. Computer Vision*, pages 268–280. IEEE Computer Society, 2001.
13. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.*, 24(6):381–395, 1981.
14. W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and center of circular features. In *Proc. of ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland*, pages 281–305, June 2-4 1987.

15. M. Fradkin, M. Roux, and H. Maître. Building detection from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, September 1999.
16. A.W. Gruen. Adaptative least squares correlation: a powerful image matching technique. *S. Afr. Journal of Photogrammetry, Remote Sensing and Cartography*, 3(14):175–187, 1985.
17. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
18. G. F. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095, 1995.
19. T. Moons, D. Frère, J. Vandekerckhove, and L. Van Gool. Automatic modelling and 3D reconstruction of urban house roofs from high resolution aerial imagery. In *Proc. 5th European Conference on Computer Vision, Freiburg, Germany*, pages 410–425, 1998.
20. S. Noronha and R. Nevatia. Detection and description of buildings from multiple images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 588–594, 1997.
21. M. Pollefeys, R. Koch, and L. Van Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 90–96, 1998.
22. C. Rother. A new approach for vanishing point detection in architectural environments. In *Proc. 11th British Machine Vision Conference, Bristol*, pages 382–391, UK, September 2000.
23. F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18:647–658, 2000.
24. C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.
25. J. A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, March 1999.
26. Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. A global matching framework for stereo computation. In *Int. Conf. Computer Vision*, pages 532–539. IEEE Computer Society, 2001.
27. C. Taylor, P. Debevec, and J. Malik. Reconstructing polyhedral models of architectural scenes from photographs. In *Proc. 4th European Conference on Computer Vision, Cambridge*. Springer-Verlag, 1996.
28. T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. The cascaded Hough transform as an aid in aerial image interpretation. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 67–72, January 1998.

# Stereo Matching with Segmentation-Based Cooperation

Ye Zhang and Chandra Kambhamettu

Video/Image Modeling and Synthesis Lab  
University of Delaware, Newark DE 19716, USA,  
{zhangye,chandra}@cis.udel.edu,  
<http://www.cis.udel.edu/~vims>

**Abstract.** In this paper we present a new stereo matching algorithm that produces accurate dense disparity maps and explicitly detects occluded areas. This algorithm extends the original cooperative algorithms in two ways. First, we design a method of adjusting the initial matching score volume to guarantee that correct matches have high matching scores. This method propagates “good” disparity information within or among image segments based on certain disparity confidence measurement criterion, thus improving the robustness of the algorithm. Second, we develop a scheme of choosing local support areas by enforcing the image segmentation information. This scheme sees that the depth discontinuities coincide with the color or intensity boundaries. As a result, the foreground fattening errors are drastically reduced. Extensive experimental results demonstrate the effectiveness of our algorithm, both quantitatively and qualitatively. Comparison between our algorithm and some other representative algorithms is also reported.

**Keywords.** Stereoscopic Vision, Occlusion Detection, Cooperative Algorithm.

## 1 Introduction

Stereo matching has long been one of the central research problems and thus one of the most heavily studied areas in computer vision. Traditional stereo matching algorithms, also known as *feature-based* methods, only match points with a certain amount of local information (such as zero-crossings or intensity edges), with the disadvantage of producing only sparse disparity maps. However, most modern applications (such as view synthesis, image-based rendering, z-keying, and virtual reality) require dense, accurate disparity maps. Therefore, we focus on dense stereo matching approaches in this paper.

Numerous dense stereo matching algorithms, including local matching (e.g., [16,10]), global optimization (e.g., [23,7]), dynamic programming (e.g., [2,14]), and cooperative algorithms (e.g., [18,28]), have been proposed over the past decades. An excellent taxonomy and evaluation of dense stereo algorithms can be

found in [25]. According to the requirements of modern applications in computer graphics and virtual reality, the disparity maps recovered by a stereo matching algorithm should be smooth and detailed, i.e., continuous and even surfaces should produce a region of smooth disparities with their boundaries precisely delineated. Unfortunately, the disparity maps produced by most stereo matching algorithms have *foreground fattening* errors due to disparity discontinuities. Adaptive window [16] and iterative evidence aggregation [24] may sometimes mitigate these errors to some extent. But they do not explicitly handle depth discontinuities and are computationally expensive. The segmentation based method proposed in [27] assumes that the disparities are piecewise smooth and embed this assumption into the planar representation of the disparities within individual image segments. This method is able to produce results with less fattening errors. However, the planar assumption may be an oversimplification of a real scene. Another difficult but critical problem in stereo matching is the handling of occlusion. Some algorithms [3,15,5] have been proposed to use the *ordering constraint* to detect occlusions. However, this constraint may not be valid in a real scene containing thin vertical foreground objects. Textureless areas pose another challenge to stereo matching. Without enough local color/intensity variations, local matching methods tend to generate arbitrarily wrong results. In this case, global optimization is preferred since there are chances that the information from correct matches can be propagated to the textureless areas.

Generally speaking, accurate stereo matching remains difficult due to depth discontinuities, occluded and textureless areas, to name a few. In this paper, we propose a new global stereo algorithm, segmentation-based cooperation, that produces accurate dense disparity maps and explicitly detects occluded areas. In our algorithm, the reference image is first segmented into homogeneous regions and each image segment is labeled with a confidence level by using cross validation. We then extend the original cooperative algorithms [18,28] in two ways. First, we design a method of adjusting the initial matching score volume. This method introduces a new concept, “feature disparity”, of a local patch within an image segment. The feature disparity can be thought of as “good” disparity information propagated within/among the image segments based on certain confidence measurement criterion. The initial matching scores of the feature disparity are set to a relatively large value so as to guarantee that correct matches have high initial matching scores. This technique raises the chances for the following update (inhibition) process to locate the correct matches. Second, we develop a scheme of choosing local support areas by enforcing image segmentation information. In this scheme, two different kinds of local support areas, matching support area and smoothing support area, are clearly distinguished. This scheme sees that the depth discontinuities coincide with the color/intensity boundaries. As a result, the foreground fattening errors are drastically reduced.

The rest of this paper is organized as follows. Section 2 discusses the general assumptions in our algorithm. Section 3 presents the segmentation-based cooperative algorithm. Extensive experimental results are reported in Section 4. Section 5 concludes this paper.

## 2 General Assumptions

All vision algorithms, explicitly or implicitly, embrace certain assumptions. For stereo matching problem, the most widely adopted assumptions include *uniqueness* and *smoothness*, i.e., the disparity maps have unique values and are continuous. In fact, these two assumptions are made not only in stereo matching but also in motion analysis, where the motion displacements are assumed to be unique and smooth. Some attempts [1,26] have been made to relax the uniqueness assumption when transparent surfaces exist in the scene. However, dealing with transparency is very difficult and the proposed methods [1,26] only work in some simple situations. In this paper, we only consider the more usual case: opaque scenes. Therefore, we still make the uniqueness assumption. The concept of “inhibition area” proposed in cooperative algorithms [18] explicitly reflects the uniqueness assumption. For two stereo images that are horizontally rectified, Fig. 1 [28] illustrates the inhibition area of a point  $(x, y)$  on the reference image when assigned a disparity  $d$ . It is easy to see that the inhibition area consists of all the possible 3D points that are projected to  $(x, y)$  on the reference image and to  $(x + d, y)$  on the other view. Since the inhibition area is explicitly considered in the update functions, cooperative algorithms possess a global optimization behavior.

Generally, the smoothness assumption is valid for the projected image areas of continuous and even surfaces. But at surface or object boundaries, this assumption is often broken. If the matching algorithm is not aware of this, the resultant disparity maps tend to be oversmooth, i.e., the details may be lost. A lot of efforts [10,13,5,27] have been made to intelligently enforce the smoothness constraint so that the disparity discontinuities can be well preserved. Inspired by [27], we assume that the disparities vary smoothly within a homogeneous image segment. However, unlike [27], we do not assume image segments as the projected areas of planes, thus making our approach more general.

Finally, without loss of generality, we assume that the input images are well rectified, i.e., the disparities are purely along one dimension.

## 3 Algorithm and Implementation

### 3.1 Initial Matching Score Volume

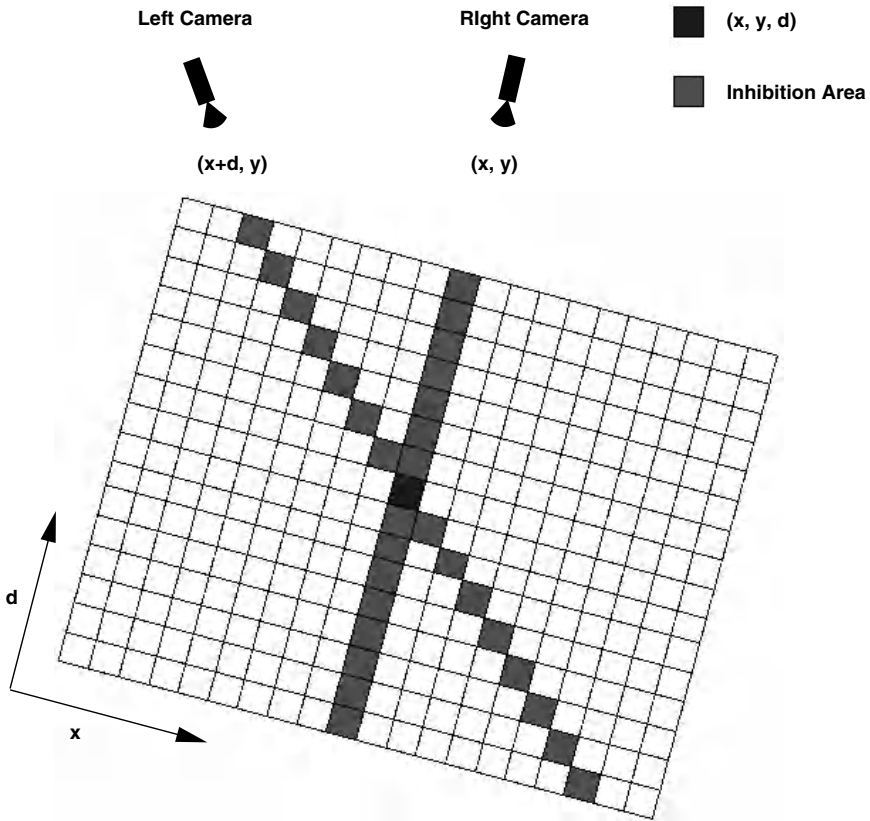
The image coordinate  $x, y$  and the disparity  $d$  defines a 3D disparity space. To compute the stereo match for a point  $p(x, y)$  on the reference image, we need to compute the matching score  $E_0(x, y, d)$  at each disparity level. Therefore all the matching scores for different points at different disparity levels form a 3D volume<sup>1</sup> in this disparity space. This volume can be defined as

$$E_0(x, y, d) = \rho(I_r, I_l, x, y, d), \quad (1)$$

---

<sup>1</sup> This volume is also called Disparity-Space Image (DSI) in [25] and [5].





**Fig. 1.** The inhibition area illustrated on a slice of matching volume ( $y$  coordinate is held constant). This illustration is based on a well-rectified image pair where the disparity is purely along  $x$  dimension.

where  $I_l, I_r$  are the intensity functions of the left and right images, respectively, and  $\rho$  is the similarity measurement function (e.g., sum-of-squared-difference (SSD), sum-of-absolute-difference (SAD), or normalized correlation). Although we only discuss the two-frame case, it is straightforward to extend the matching score volume to the multiple-frame case by simply summing up the scores from other views since the matching score volume is associated with a fixed reference image. For example, [20], [17] and [19] exploited this idea to compute multiple baseline stereo by using sum-of-SSD (SSSD) or sum-of-SAD (SSAD).

### 3.2 Initial Matching Score Adjustment

Cooperative algorithms require that the correct matches produce high initial matching scores. However, the opposite does not need to be true [28]. This is because cooperative algorithms make decisions “globally” through inhibition

and local support, thus possessing a good tolerance to false high-score matches. However, due to projective distortion and inappropriate window sizes or image noise, some correct matches may produce low initial matching scores. Therefore, we need to adjust the initial matching score volume so that we can make sure that correct matches are indeed labeled with high initial scores.

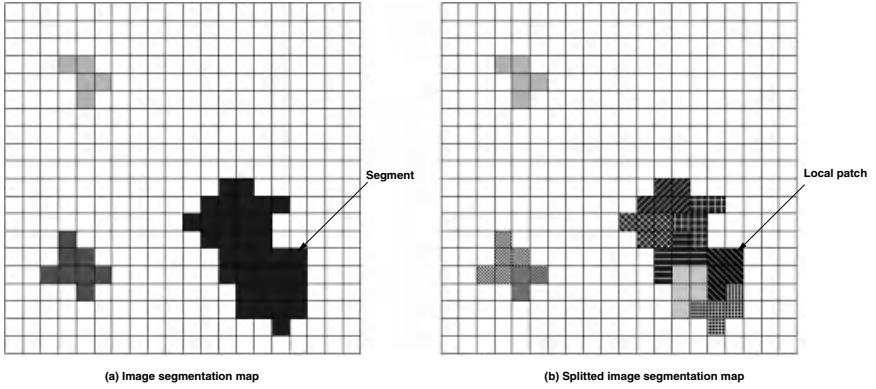
We exploit the ideas of *cross validation* and *image segmentation* to adjust the initial matching scores. In this paper, we adopt the image segmentation algorithm proposed in [8]. According to our assumptions, the disparities should change smoothly within an image segment. In [27], these disparities have been modeled as a plane, or a plane-plus-parallax. More advanced models for image segments, such as a variable-order parametric model [4], have also been proposed to represent the displacements during motion analysis. However, by using model-based representation, the accuracy of the recovered disparities is limited by the model's ability to approximate the real surfaces. For example, if the scene contains a spherical surface that is approximated by a plane model, the results will not be accurate. In our work, we do not use any *a priori* model to represent the disparities in an image segment. Instead, we adopt a "multiresolution" strategy to adjust the initial matching scores so that the update process can be attracted towards the correct matches.

Image segmentation, in fact, is an information-based scale-space filtering. In the segmentation map, the image portions with similar color or intensity, and normally within the same neighborhood, are aggregated as a segment, thus reducing the resolution. However, similar colors do not always mean similar depths. For example, the projected image area of a very slanted Lambertian surface with uniform texture tends to be classified as one image segment, while the depths may change a lot within this segment. Based on this observation, to adjust the matching volume, we first split each image segment into small local patches (segments smaller than the pre-defined local patch size are deemed as one patch. If the final remainder of a large segment after splitting is smaller than the patch size, it is deemed as one patch). The splitting process is illustrated in Fig. 2. Splitting a large segment into small local patches equals to increasing the resolution of the disparities within this segment. We assume that, within each small local patch, the disparities are very similar, i.e., if we assign an appropriate disparity to all the points in the local patch, the corresponding patch in the other view should be very similar to the original patch. We call this disparity "feature disparity" of this local patch. We can find the feature disparity  $d_i$  of patch  $p_i$  by solving

$$d_i = \underset{d \in D}{\operatorname{argmax}} \operatorname{Sim}(i, d), \quad (2)$$

where  $\operatorname{Sim}(i, d)$  is a similarity measurement when patch  $p_i$  is assigned a disparity  $d$ ,  $D = (D_{\min}, D_{\max})$  is the set of all the possible disparities.

However, performing exhaustive search in  $D$  to find the feature disparity of a patch is not a good idea. The reasons are two-fold: one is that exhaustive search is time-consuming; the other is that large search scope increases the chances for image noise to overwhelm the correct results, as typically the local patch does



**Fig. 2.** The process of splitting image segments.

not have much texture information. In our algorithm, we design a method to reduce the search scope based on a confidence measurement of each segment. The confidence measurements are delivered via cross validation<sup>2</sup>. First, a matching score volume  $E_0(x, y, d)$  is computed between a stereo pair  $I_0$  and  $I_1$  along the epipolar line. The measured disparity is the one with the largest matching score. We perform the similarity computation twice by reversing the roles of the two images and consider as valid only those matches for which we measure the same depth at corresponding points when matching from  $I_0$  to  $I_1$  and from  $I_1$  to  $I_0$ <sup>3</sup>. To further increase the *Signal/Noise* ratio, we filter out those valid points that are either isolated or have very large standard deviation in a small neighborhood. Finally, we get an initial disparity map with few errors. Fua [11] pointed out that as the *Signal/Noise* ratio decrease, the performance of cross validation degrades gracefully in the sense that the density of matches decreases accordingly but the ratio of correct to false matches remains high. In other words, a relatively *dense* disparity map is almost a *guarantee* that the matches are correct (up to the precision allowed by the resolution being used). In fact, to further guarantee the correctness of the valid matches, we can adopt a simplified version of adaptive windows [16] to perform the cross validation: We can perform cross validation by using different sizes of local windows (e.g.,  $5 \times 5$  and  $3 \times 3$ ) and consider as valid only those matches for which we measure the same depth by using different window sizes. Based on these observations, if we divide the reference image into segments with homogeneous color/intensity, we can label the confidence level

<sup>2</sup> It is also interesting to notice that from the perspective of cooperative algorithms, cross validation is actually performed in the inhibition area.

<sup>3</sup> Following this method, it is straightforward to extend cross validation to more than two frames: Compute the valid disparities between  $I_0$  and  $I_2, I_3, \dots, I_{N-1}$ , respectively. Then merge the results together and get a sparse initial “valid” disparity map. If two sets of views produce different valid disparities, the one with higher matching score wins. Also, the matching score in the matching score volume is updated accordingly.

of the disparities in each image segment according to the density of the valid matches within the corresponding segment. That is,

$$L(s) = \begin{cases} VALID & \text{if } r \geq \alpha_1; \\ SEMIVALID & \text{if } \alpha_2 \leq r < \alpha_1; \\ INVALID & \text{if } r < \alpha_2, \end{cases} \quad (3)$$

where  $r$  is the ratio of valid disparity points in segment  $s$ ,  $\alpha_1$  and  $\alpha_2$  are positive thresholds, and *VALID*, *SEMIVALID* and *INVALID* are all symbolic values. *VALID* means that we have high confidence on the disparity map within segment  $s$ . *INVALID* means low confidence, and *SEMIVALID* means medium confidence. This labeling method reflects an assumption we have made: image segments where the valid disparity points are dense are more reliable. Again, Fua's experiments [12] have shown that this assumption holds in most cases.

Once we have labeled the confidence level of image segments, we can compute the feature disparity of each local patch by

1. If patch  $p_i$  belongs to a *VALID* segment  $s$ , find the minimum ( $dmin_i$ ) and maximum ( $dmax_i$ ) disparities of all the valid points in  $s$ . Then solve

$$d_i = \underset{dmin_i - \delta < d < dmax_i + \delta}{argmax} Sim(i, d), \quad (4)$$

where  $\delta$  is a small positive number, and  $d_i$  is the feature disparity of  $p_i$ .

2. If patch  $p_i$  belongs to a *SEMIVALID* segment  $s$ , find the minimum ( $dmin_i$ ) and maximum ( $dmax_i$ ) disparities of all the valid points in  $s$  and all its neighboring segments. Solve Eq. 4, and  $d_i$  is the feature disparity of  $p_i$ . If patch  $p_i$  belongs to an *INVALID* segment  $s$ ,  $d_i$  is undefined.

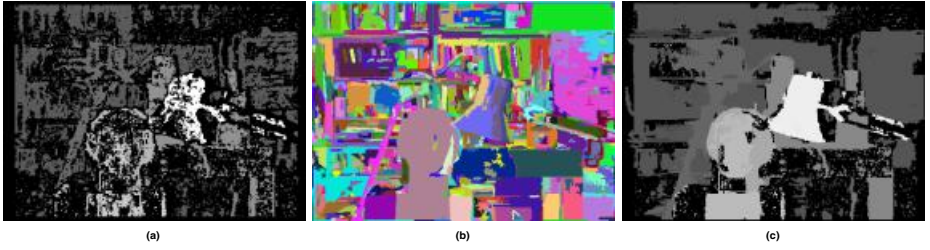
From above, we can see that for *VALID* and *SEMIVALID* segments, the disparity range for searching feature disparity has been reduced. This will not only improve the efficiency but also improve the robustness of the algorithm. Notice that we do not compute the feature disparity for *INVALID* segments. This is because it is very possible that the *INVALID* segments may be occluded areas and no reliable information is available. The feature disparities computed for Tsukuba head scene are illustrated in Fig. 3.

After we find the feature disparities for some patches, the initial matching score volume is adjusted as

$$E_0(x, y, d) = \begin{cases} C & \text{if } \exists i \ (x, y) \in p_i \text{ and } d = d_i, \\ E_0(x, y, d) & \text{else,} \end{cases} \quad (5)$$

where  $d_i$  is the feature disparity for local patch  $p_i$  and  $C$  is a relatively large matching score. In our work, it is set as the maximum value of all the initial match scores at point  $(x, y)$ .

Compared with [27] where disparities in one segment are always iteratively hypothesized according to the neighboring segments (we call this *inter-segment* hypothesis), our method should be more efficient. For a segment with high confidence, we search the feature disparity only within the valid disparity range of



**Fig. 3.** Feature disparities computation: (a) initial valid disparity map, (b) image segmentation map, (c) computed feature disparity map (black areas mean no definition).

this segment (we call this *intra-segment* hypothesis). Only for those segments without enough confidence, we utilize the information from the neighbors. It is also important to remember that the feature disparity is *not* the final disparity value assigned to every point in one local patch. However, we believe that it should be close to the correct disparities. Therefore, the feature disparity serves as a force to drag the update process towards the correct disparities. It is still the global cooperation that makes the final decision on the disparity values.

### 3.3 Segmentation-Based Local Support

Many stereo matching algorithms require a concept of “local support”, i.e., aggregation of evidence from the neighboring pixels. This is because stereo matching is in general ambiguous: there may be multiple equally good matches if the matching score is computed independently at each point. Normally a support region is a two-dimensional ( $x - y$ ) or three-dimensional ( $x - y - d$ ) neighborhood of the current pixel. Traditional local support assumptions include: the depth constancy assumption by Marr and Poggio [18,9], the disparity gradient limit by Pollard, Mayhew and Frisby (PMF) [21], the disparity similarity function by Prazdny [22]. Kanade and Okutomi [16] presented a detailed analysis of the relationship and differences among them. They also proposed a method to adaptively adjust the SSD or correlation window size and shape according to the variation of the local intensities and disparities.

However, it is important to realize that there are two kinds of “local support”. The first one is used to compute the matching score of a point given a disparity. The matching score of a point is normally the similarity measurement between a local area of the interested point and the local area in the other view corresponding to the assigned disparity. We call this local area, often in the form of a  $m \times n$  window centered on the interested point, a “matching support” area. The matching support area needs to be large enough to contain enough color/intensity variations (texture information) for reliable matching, and be small enough to avoid the effects of projective distortion. For example, the local window used to compute the matching volume and the adaptive window in [16] belong to the matching support. The underlying assumptions of this support are Lambertian surfaces, i.e., surfaces whose appearance does not vary with

viewpoint. The other kind of local support is called the “smoothing support”. The local support used in cooperative algorithm [28] belongs to this category. The purpose of the smoothing support is mainly to propagate disparity information within a neighborhood and make the resultant disparity maps smooth. The underlying assumption is that the disparities do not vary much within the smoothing support area of the interested point. In smoothing support area, we do not need rich color/intensity variations (texture information). The only concern is to make sure disparities actually do not change much within this area.

For simplicity, Zitnick and Kanade [28] chose a box-shaped 3D local smoothing support area. The problem of this simple strategy is that the depth discontinuities may be blurred because the smoothing support area may be applied across the depth boundaries. In our work, we propose a scheme to choose the smoothing support area by utilizing the image segmentation information. Specifically, we define the 3D smoothing support area of a point  $(x, y)$  as

$$\Phi(x, y, d) = \{(x', y', d') \mid (x, y) \in p_i \wedge (x', y') \in p_i \wedge d' \in [d - r_d, d + r_d]\}, \quad (6)$$

where  $p_i$  is the local patch within an image segment that contains  $(x, y)$ , and  $r_d$  is a small positive number that defines the support along the  $d$  dimension. Since the local patch within an image segment is used to define the  $x - y$  support, the image segmentation information is explicitly enforced.

Then, we can define an aggregated matching score volume by averaging the matching scores within the smoothing support areas:

$$A_n(x, y, d) = \frac{1}{N(x, y, d)} \sum_{(x', y', d') \in \Phi(x, y, d)} E_n(x', y', d'), \quad (7)$$

where  $E_n$  is the matching volume,  $n$  is the iteration number, and  $N(x, y, d)$  is the number of points in  $\Phi(x, y, d)$ .

Observing the definition of  $\Phi$  carefully, we can notice that it has some interesting characteristics. First, unlike common local support definitions, smoothing support areas  $\Phi$  of different points do not overlap with each other in the image plane. Second, all the points in patch  $p_i$  have the same aggregated matching score. This means that the aggregation process can be implemented in a very fast way. However, the drawback is that the disparity propagation within one image segment may not be enough. Therefore, a substitute smoothing support area of  $(x, y, d)$  can be defined as  $(x, y)$ 's local patch  $p_i$  and  $p_i$ 's neighbors, i.e.,

$$\Phi'(x, y, d) = \{(x', y', d') \mid (x, y) \in p_i \wedge ((x', y') \in p_i \vee ((x', y') \in p_j \wedge p_j \text{ is a neighbor of } p_i)) \wedge d' \in [d - r_d, d + r_d]\}. \quad (8)$$

Then the aggregated matching volume is defined as

$$A'_n(x, y, d) = \frac{1}{N'(x, y, d)} \sum_{(x', y', d') \in \Phi'(x, y, d)} E_n(x', y', d'), \quad (9)$$

where  $N'(x, y, d)$  is the number of points in  $\Phi'(x, y, d)$ .

$A'_n(x, y, d)$  has the advantage of propagating more information through an image segment because  $\Phi'$  overlap with each other in an image segment, thus making the disparities within the image segment more smooth. In our implementation, we use  $\Phi'$  as local support areas. It is also worth mentioning that weighted summation may be used in Eq. 9. For simplicity and efficiency reasons, we use simple summation in our implementation.

### 3.4 Matching Score Update

The uniqueness assumption states that one pixel in the reference image has only one match within a set of elements that project to the same pixel in the other view. As illustrated in Fig. 1, the inhibition area of a point  $(x, y, d)$  can be defined as all the elements that overlap this point when projected to an image. This means that the inhibition area consists of two lines of sight. In other words, the inhibition area  $\Psi$  of point  $(x, y)$  when assigned disparity  $d$  can be defined as

$$\Psi(x, y, d) = \{(x', y', d') \mid (x', y', d') \text{ projected to } (x, y) \text{ in the} \\ \text{reference view or } (x + d, y) \text{ in the other view}\}. \quad (10)$$

Many inhibition functions are available. Here we choose the one used by Zitnick and Kanade [28] for its simplicity:

$$R_n(x, y, d) = \left( \frac{A_n(x, y, d)}{\sum_{(x', y', d') \in \Psi(x, y, d)} A_n(x', y', d')} \right)^\alpha, \quad (11)$$

where  $R_n(x, y, d)$  denotes the amount of inhibition at  $(x, y, d)$  and  $\alpha$  is a positive constant called the “inhibition constant”. This constant controls the speed of convergence. To guarantee that there is a single element within  $\Psi$  that will converge to 1,  $\alpha$  must be greater than 1. Then the update function can be defined as

$$A_{n+1}(x, y, d) = A_0(x, y, d) * R_n(x, y, d). \quad (12)$$

To prevent oversmoothing to some extent, the initial aggregated match values  $A_0$  are introduced in this update function to restrict the current match values. Zitnick and Kanade [28] compared this update function with the original Marr and Poggio [18] update function. Three advantages have been claimed: First, the Marr and Poggio function used discrete match values, and was not well defined for real scenes, while Eq. 12 uses continuous matching values and is well defined for real scenes. Second, use of  $A_0$  maintains better disparity details. Third, Eq. 12 is much simpler and computationally efficient. In our work, we used this update function and found that it worked excellently in our experiments. Fig. 4 illustrates an initial versus converged slice of the matching volume.



**Fig. 4.** The convergence of a slice in the matching score volume by applying Eq. 12: (a) the initial slice, (b) the converged slice.

### 3.5 Occlusion, Confidence Measurement, and Subpixel Accuracy

Real scenes almost always contain occluded areas. Unfortunately, most stereo algorithms are not able to handle occlusion explicitly. Instead, most of them hypothesize disparities in occluded areas based on the disparities in the neighborhood and may produce errors. Increasing the number of cameras is a natural way to reduce occlusion, but it is not always feasible. Some research (e.g., [5]) have proposed finding occlusion and matches simultaneously by imposing the ordering constraint, which states that the objects maintain the same left-to-right order in different views. However, the ordering constraint may mislabel visible pixels as occluded, and this constraint may not be valid when there exist thin vertical objects in the foreground.

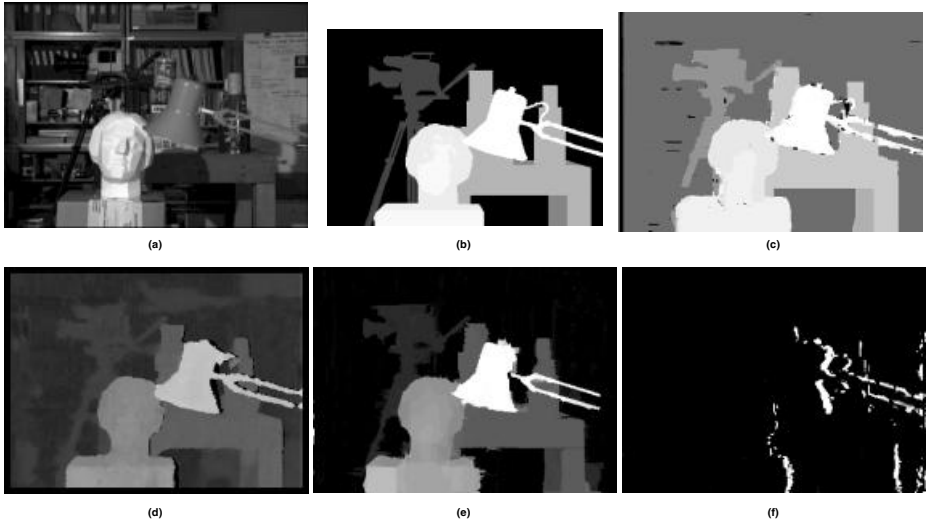
In cooperative algorithms, the converged match values can be used as a natural criterion for occlusion detection [28]. Because no correct matches exist for the occluded areas, the converged match values corresponding to occluded pixels should be small. Furthermore, the update (inhibition) process decreases the match values of occluded pixels. However, for mutually occluded areas within the disparity range, higher match values may occur in occluded areas. So, as long as the mutually occluded areas do not have similar colors/intensities, all the converged match scores at occluded pixels should be small. Thus, in our algorithm, if a converged matching score is less than a threshold, the corresponding pixel is labeled as occluded. Following the same logic, the converged match scores of the resultant disparity map can be directly used as its confidence measurements.

In our algorithm, subpixel accuracy can be achieved via two ways. One is to split the initial matching volume into half-pixel or quarter-pixel levels. The matching scores at subpixel levels can be interpolated by fitting a curve (e.g., a quadric) to the neighboring scores. The other way is to directly fitting a curve to the final matching scores after the update process converges. We adopt both methods in our implementation.

## 4 Experimental Results

We have implemented our segmentation-based cooperative (SBC) algorithm under a PC platform. The algorithm takes about four seconds per iteration with  $256 \times 256$  images on a Pentium III 800MHz machine. We have applied this algorithm to real imagery. Fig. 5 illustrates the results on the head scene from





**Fig. 5.** Results on Tsukuba heads cene: (a) is the reference image. (b) is the ground truth. (c), (d) and (e) are the disparity maps computed by using the GPM-MRF algorithm [6], the cooperative algorithm [28], and the SBC algorithm, respectively. (f) is the occlusion map computed by using the SBC algorithm.

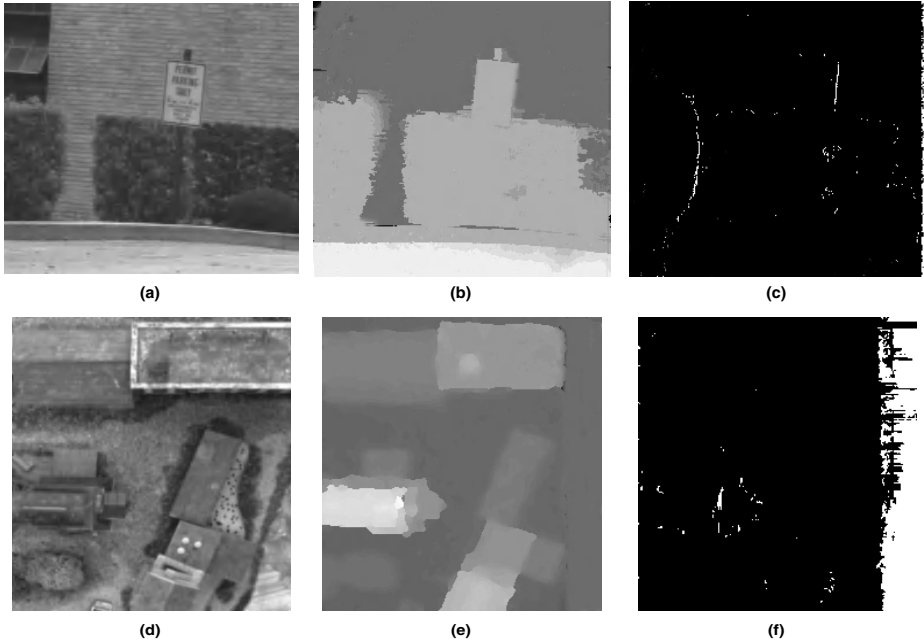
University of Tsukuba. This data set consists of 25 images taken from a  $5 \times 5$  camera array. Only two images along a horizontal row are used as the input data to our algorithm. The head scene contains textureless areas such as the table and the lampshade. It also contains thin structures such as the rods of the lamp. Figures 5 (a) and (b) show the reference image and the ground truth, respectively. For comparison purpose, Figures 5 (c), (d), (e) show the results produced by using the GPM-MRF algorithm [6], the cooperative algorithm [28], and the SBC algorithm, respectively. Fig. 5 (f) shows the detected occluded areas by the SBC algorithm. We can see that the rods of the lamp, the shape of the head, the outline of the desk and the profile of the camera are clearly preserved in our result. We can also see that for thin structures (such as the rods of the lamp), our algorithm produces least fattening errors. Our algorithm also correctly reports the occluded areas, i.e., the right sides of the lampshade, the desk and the head, the upper sides of the rods.

Since the head scene is provided with dense ground truth disparities, we can quantitatively evaluate the SBC algorithm. Table 1 shows the comparison between the SBC algorithm and some other representative algorithms. The error rate is defined as the percentage of those disparity values with absolute errors greater than one pixel compared with the ground truth. From the table we can see that our algorithm produces very accurate results.

Fig. 6 presents the results on the CMU shrub scene and coal mine scene. The disparity maps are smooth and maintain clear depth boundaries at the same time. For the CMU shrub scene, the parking sign and the shrub boundaries are

**Table 1.** Comparison of SBC and other algorithms on Tsukuba head scene

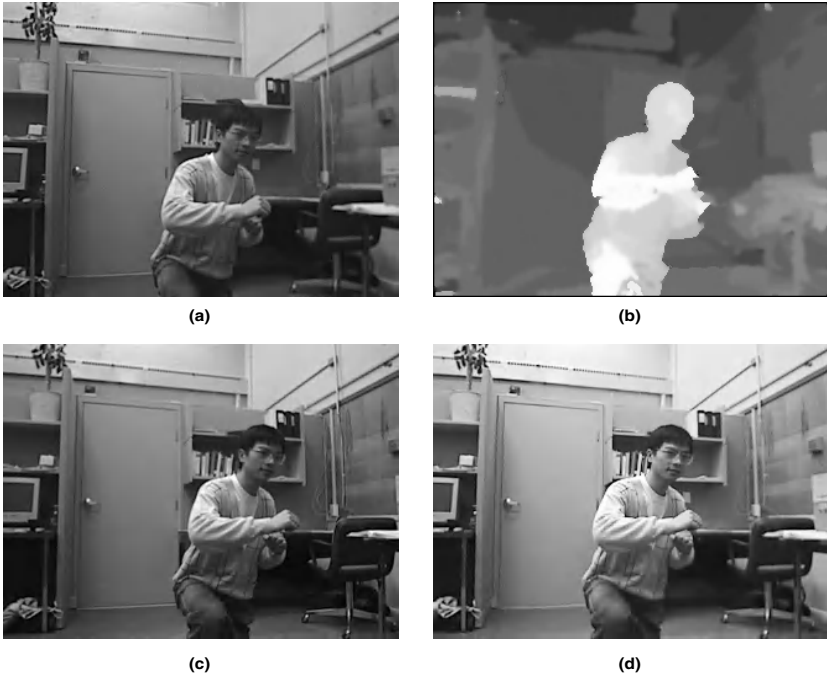
Algorithms	$Error > 1$
SBC	1.2
Zitnick and Kanade[28]	1.4
GPM-MRF [6]	2.8
LOG-filtered $L_1$ [6]	9.0
Normalized Correlation [6]	10.0



**Fig. 6.** Results on CMU shrub and coal mine scenes: (a) and (d) are the reference images of shrub and coal mine scenes, respectively. (b) and (e) are the disparity maps computed by using the SBC algorithm. (c) and (f) are the computed occlusion maps.

clearly preserved in the produced disparity map. Although the background (the brick wall) contains a lot of repetitive patterns, our algorithm still recovered the disparities correctly. For the CMU coal mine scene, the shapes of the buildings are precisely delineated in the produced disparity map.

We also extended our algorithm to three-frame stereo matching by accumulating the matching scores from all views. Fig. 7 shows the results by applying our algorithm on a Triclops snapshot. The Triclops consists of three calibrated camera heads configured as a “L” shape. Figures 7 (a), (c) and (d) show the images acquired by the top, the left and the right camera, respectively. Fig. 7 (b) shows the computed disparity map. We can see that the outline of the person



**Fig. 7.** Results on a three-view lab scene by using SBC algorithm: (a) top image, (b) disparity map, (c) left image, (d) right image.

is clearly maintained. The depth relationships between the body, the hand and the leg are accurately recovered.

We have further performed extensive experiments on other benchmark data (such as the pentagon scene, the meter machine scene, etc.) and also on stereo data produced in our lab. Our system has consistently produced accurate disparity maps.

## 5 Conclusion

We have presented a new segmentation-based cooperative algorithm for stereo matching. This algorithm extends the earlier cooperative algorithms [18,28] in two ways. First, we designed a method of adjusting the initial matching score volume to guarantee that correct matches have high matching scores. This method propagates reliable disparity information among/within image segments based on the confidence labels of image segments, thus improving the robustness of the algorithm. Second, we developed a scheme for choosing local support areas by enforcing the image segmentation information. This scheme sees that the depth discontinuities coincide with the color/intensity boundaries. As a result, the foreground fattening errors are drastically reduced. We also show that the converged

matching scores can be used as the confidence measurements and occluded areas can be easily detected by setting a threshold on the converged matching scores. Through extensive experiments, we demonstrate the effectiveness of our SBC algorithm.

Our algorithm may produce oversmooth results when depth discontinuities appear in a homogeneous image segment. One possible solution is to enforce not only color/intensity segmentation information, but also depth segmentation information. By doing so we can make sure that the smoothing support areas seldom overlap depth discontinuities, thus maintaining more detailed depth boundaries.

**Acknowledgments.** Research funding was provided by the National Science Foundation Grants CAREER IRI-9984842 and CISE CDA-9703088.

## References

1. E.H. Adelson and P. Anandan. Perceptual organization and the judgment of brightness. *Science*, 262:2042–2044, 1993.
2. P.N. Belhumeur. A bayesian-approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996.
3. P.N. Belhumeur and D. Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 506–512, 1992.
4. M.J. Black and A.D. Jepson. Estimating optical-flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.
5. A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):1–20, September 1999.
6. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.
7. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 377–384, 1999.
8. D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.
9. M. Drumheller and T.A. Poggio. On parallel stereo. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1439–1448, 1986.
10. P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, 1993.
11. P.V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.
12. P.V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.

13. E.B. Gamble and T. Poggio. Visual integration and detection of discontinuities: The key role of intensity edges. In *MIT AI Memo*, 1987.
14. D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Proceedings of European Conference on Computer Vision*, pages 425–433, 1992.
15. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proceedings of European Conference on Computer Vision*, pages xx–yy, 1998.
16. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.
17. S.B. Kang, J. Webb, C.L. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 88–93, 1995.
18. D. Marr and T.A. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, October 15 1976.
19. Y. Nakamura, T. Matsura, K. Satoh, and Y. Ohta. Occlusion detectable stereo – occlusion patterns in camera matrix. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 371–378, 1996.
20. M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
21. S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
22. K. Prazdny. Detection of binocular disparities. *BioCyber*, 52:93–99, 1985.
23. S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 492–499, 1998.
24. D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.
25. D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo methods. In *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 131–140, 2001.
26. R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, August 1999.
27. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages I: 532–539, 2001.
28. C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000.

# Coarse Registration of Surface Patches with Local Symmetries

Joris Vanden Wyngaerd<sup>1</sup> and Luc Van Gool<sup>1,2</sup>

<sup>1</sup> University of Leuven, ESAT-PSI, Kasteelpark Arenberg 10,  
B-3001 Leuven, Belgium

`joris.vandenwyngaerd@esat.kuleuven.ac.be`

`http://www.esat.kuleuven.ac.be/psi/visics`

<sup>2</sup> Kommunikationstechnik, ETZ, Gloriastr. 35, ETH Zentrum,  
CH-8092 Zürich, Switzerland

`vangool@vision.ee.ethz.ch`

**Abstract.** Most 3D recording methods generate multiple partial reconstructions that must be integrated to form a complete model. The coarse registration step roughly aligns the parts with each other. Several methods for coarse registration have been developed that are based on matching points between different parts. These methods look for interest points and use a point signature that encodes the local surface geometry to find corresponding points. We developed a technique that is complementary to these methods. Local descriptions can fail or can be highly inefficient when the surfaces contain local symmetries. In stead of discarding these regions, we introduce a method that first uses the Gaussian image to detect planar, cylindrical and conical regions and uses this information to compute the rigid motion between the patches. For combining the information from multiple regions to a single solution, we use a Hough space that accumulates votes for candidate transformations. Due to their symmetry, they update a subspace of parameter space in stead of a single bin. Experiments on real range data from different views of the same object show that the method can find the rigid motion to put the patches in the same coordinates system.

**Keywords :** Surface Registration, Surface geometry, Shape.

## 1 Introduction

Three-dimensional (3D) models of objects usually are not acquired in one piece. We will refer to a partial reconstruction from a single view as patch. An important step in the procedure to integrate them is the registration. There are good algorithms to do this, e.g. ICP [2,5] or the maximization of mutual information [19]. Nevertheless, such registration algorithms require an initial, coarse registration. Starting from very different initial positions they might not converge to the correct solution. Providing such initial coarse registration automatically is the goal of the work reported here.

## 1.1 Previous Work

Registration and 3D model-based recognition are closely related. For both applications, correspondences between 3D surfaces must be determined. Two approaches can be taken to represent surfaces for these tasks, global or local.

Global representations represent the entire surface. Some represent the entire surface by a parametric surface [15]. Another global method is the extended Gaussian image for which matching is based on correlating spherical maps [11, 13]. Because global representations represent the entire surface, they are not well suited for the registration of several partial surfaces.

Local methods can handle partial surfaces better. They typically use a local description of the surface geometry around special surface features. Correspondences are determined by comparing these point signatures [6,7,8,12,21] and the correct relative position between the patches is computed based on these corresponding points. Several methods have been developed to find these point correspondences. A good example is the work by Johnson and Hébert [12]. They use so called spin images for describing the local geometry of all the surface points. Yamani and Farag use a similar approach but apply a selection process based on surface curvature to calculate the surface signatures only for points of interest [21]. This work can also be seen as an extension to our previous work [18] where we use bitangent curve pairs as landmarks on the surface. Bitangent curve pairs represent a point pair with a common tangent plane sliding over the surface. The type of surfaces we analyze here are degenerate cases of bitangency, where bitangent curve pairs are not defined. For example, in a planar part all points have the same tangent plane.

## 1.2 Problem and Chosen Approach

Point signatures are created based on local information. This is necessary to have a compact description and for robustness against occlusion. However, such local descriptions can become useless when the surface contains (local) symmetry. For example, all points in a plane or on a cylinder will have exactly the same signature and cannot be used to find corresponding points in this kind of regions. Yamani and Farag [21] only calculate surface signatures in points with higher curvature. They discard the points with low curvature because they *"are redundant and do not serve as landmarks of the object"*. This is a problem when all points of a patch have low curvature. The method we report here is complementary because it focuses on points with low curvature, more precisely points that have zero Gaussian curvature. We developed a method that detects planar, cylindrical and conical parts in the surface patches and uses this information to calculate the unknown transformation. Computing rigid motion from many point correspondences has been studied by several authors [10,14]. The problem they solve is typically highly overdetermined because many point correspondences are available. These methods can not be used in the presence of local symmetry, because no individual points can be identified and matched. In this paper we introduce a method for computing the rigid motion by combining

the constraints imposed by the detected planar, cylindrical and conical regions in the surfaces. For example, two planes can be registered correctly while keeping three degrees of freedom, two translational components parallel to the plane and a rotation around the plane's normal vector. They do not define a unique transformation, but they do impose constraints on the possible transformations. We analyze these constraints and combine them by updating a Hough space where votes are accumulated for candidate transformations.

This paper is organized as follows. Section 2 describes how we detect the planar, cylindrical and conical regions in the surfaces. Section 3 describes how we calculate the transformation. Section 4 shows the results of the algorithm. Section 5 concludes this paper.

## 2 Detecting Planes, Cylinders, and Cones

### 2.1 The Gaussian Image

The Gaussian image [4] maps the surface normal onto the Gaussian sphere. It provides a unique representation for convex objects. If  $G$  represents this mapping,  $\delta O$  a patch on a surface and  $G(\delta O)$  the corresponding patch on the Gaussian sphere, the Gaussian curvature  $K$  can be defined as

$$K = \lim_{\delta O \rightarrow 0} \frac{G(\delta O)}{\delta O} \quad (1)$$

Several extensions have been made to encode more information. The extended Gaussian image (EGI) takes into account the area on the surface of points with the same normal [11]. The complex extended Gaussian image (CEGI) adds the distance of the tangent plane to the origin [13]. The spherical attribute image (SAI) developed by Hébert, Ikeuchi and Delingette [9] also encodes surface connectivity to provide a unique representation of the surface, up to rotation, translation and scale. These representations are useful for recognition [11] and pose determination [4] [13]. Sun and Sherrah use the EGI for symmetry detection [16].

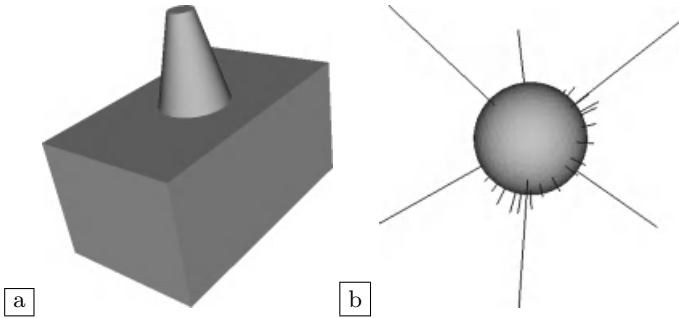
When more information is encoded, creating the representation will be more demanding. For some applications, a more complex representation is justified. When building a database for object recognition for example, using more complex models is reasonable because they can be created off-line and are used many times. In our application, the models are created at runtime and used only once. Using complex models is not practical, especially when many patches are included.

As complementary information to that which our earlier work in [18] can provide, we only want to detect the presence of planar, cylindrical and conical regions. For this, we use the orientation histogram, which is a discrete version of the extended Gaussian image. It is used to find candidate regions for fitting a shape. For these candidates, the type of shape and their parametric representation will be computed. We will refer to the planar, conical and cylindrical regions that we look for as 'shapes'. For a region to be classified as a shape, we require that it is connected.



## 2.2 Finding Special Regions

The orientation histogram is used for detecting planar, conical and cylindrical regions in a surface. The Gaussian image  $G$  maps a surface patch  $\delta O$  in a planar, cylindrical or conical region on a region on the Gaussian sphere with zero area, and points in such regions therefore have a zero Gaussian curvature. Because of this, the shapes we look for generate high values in the orientation histogram. All points in a planar part have the same surface normal and are mapped on the same point on the Gaussian sphere. A cone or cylinder is mapped onto a circle on the Gaussian sphere. This is illustrated in fig. 1.



**Fig. 1.** (a) An example object. (b) The orientation histogram or needle map. Triangles of the unit sphere are histogram bins. The normal to the triangle is also shown with a length proportional to the histogram value. The isolated big peaks correspond to the planes of the box. The circular configuration of smaller peaks corresponds to the cone.

To locate candidate regions we detect peaks in the orientation histogram. Neighboring bins are combined to form clusters. For non-convex objects, such a cluster can contain different shapes. For example, all points in parallel planar regions will fall in the same cluster. The clusters are split so that they only contain one shape. In a first step we split clusters based on the requirement that the points in the shape have to be connected. It is easy to see that two parallel but non-coplanar planar regions cannot be connected on the surface without leaving the cluster. However, this can be possible for two cylindrical regions. When a candidate region has been identified, we have to determine the type of shape and its parametric representation. In a first step, we determine whether the region can be planar or not. This is done by observing that for a plane, all normals point in the same direction, while this is not the case for cones and cylinders. If a region is classified as non-planar, we take a second step by splitting the region into convex parts.

To validate the hypotheses, we will fit a parametric model and calculate the Euclidean distance between the points that were used for fitting and the shape represented by the parametric model.

### 2.3 Fitting Models to the Data

We represent a plane by its equation

$$P: ax + by + cz + d = 0 \quad (2)$$

$\mathbf{N}(a, b, c)$  represents the normal to the plane. We require that it is a unit vector. Like this, the absolute value of the residue of this equation represents the Euclidean distance between a point and the plane. We also require that the normal is directed outward. This is done by giving it the same direction as the corresponding bin in the orientation histogram.

Fitting a 3D plane to a set of points is a text book least square problem. Suppose we have a number of points  $\mathbf{X}(x, y, z)$  and have to determine the parameters  $(a, b, c, d)$ . Least squares will minimize the residue of this equation and as such minimize the Euclidean distances of the points to the plane. Fitting of cones and cylinders to data is a lot more complicated. One approach is to use the generic equation of a quadric :

$$q(x, y, z) = ax^2 + by^2 + cz^2 + 2hxy + 2gxz + 2fyz + 2ux + 2vy + 2wz + d = 0 \quad (3)$$

The parameters of this equation can also be determined by a linear least square method, similar to the plane. This however poses two problems. First, the residue of this equation is not the Euclidean distance, but an algebraic distance. Bolle and Cooper [3] use a different parameterization of cylinders such that the residue represents the Euclidean distance. The solution is found with a non-linear optimization, which is computationally expensive. Secondly, no guarantee exists that the resulting quadric represents a cone or a cylinder. Especially when only partial data is available and in the presence of noise, the fitting can result in other types of quadrics. This is illustrated in figure 2. Werghi *et al.* [20] developed a method for constraining the solution to for example a cylinder. They introduce Lagrange multipliers to impose constraints on the unknown parameters. The nonlinear constrained optimization problem is solved with a Levenberg-Marquardt method. This leads to a computationally expensive numerical algorithm.

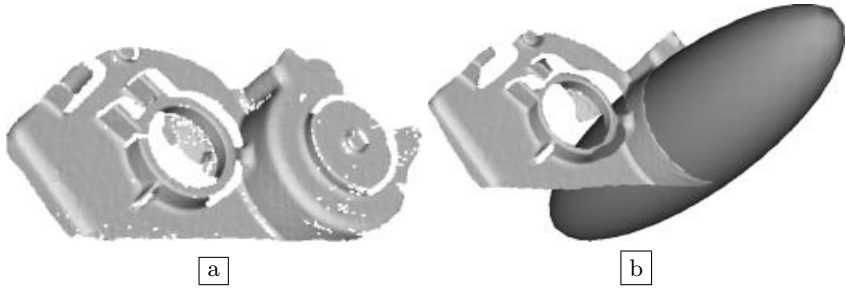
The reason why a procedure that minimizes the Euclidean distance is complicated is that no closed form expression exists. We illustrate this by explaining the procedure we use to calculate the distance between a point  $\mathbf{X}_1(x_1, y_1, z_1)$  and a quadric represented by (3). Let  $\mathbf{X}(x, y, z)$  denote the (unknown) point on the quadric that is closest to  $\mathbf{X}_1$ . This point can be found by minimizing the square of the Euclidean distance between  $\mathbf{X}_1$  and  $\mathbf{X}$  :

$$d^2(x, y, z) = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \quad (4)$$

subject to the constraint (3), i.e. that  $\mathbf{X}$  lies on the quadric. The function to be minimized becomes:

$$d^2(x, y, z) + \lambda q(x, y, z) \quad (5)$$

Partial differentiation with respect to the unknowns  $(x, y, z, \lambda)$  leads to the following set of equations :



**Fig. 2.** (a) Surface with a conical region. Due to the partial nature of our data and the geometry of the object, only a very small part of the mathematical cone is present in the data. (b) Due to measurement noise, the quadric fitted to these points is not a cone, but an ellipsoid.

$$\begin{cases} x - x_1 + \lambda ax + \lambda hy + \lambda gz + \lambda u & = 0 \\ y - y_1 + \lambda hx + \lambda by + \lambda fz + \lambda v & = 0 \\ z - z_1 + \lambda gx + \lambda fy + \lambda cz + \lambda w & = 0 \\ ax^2 + by^2 + cz^2 + 2hxy + 2gxz + 2fyz + 2ux + 2vy + 2wz + d & = 0 \end{cases} \quad (6)$$

The first three equations are linear and can be solved for  $x, y$  and  $z$ . Substitution in the last equations leads to a polynomial in  $\lambda$  of degree 6. The real roots are substituted in (4) and the one with the smallest distance is chosen.

It is clear that this distance is not a closed form solution. Using this for calculating the residue in an iterative optimization procedure would be computationally expensive. We made a trade-off between accuracy and speed. As will be clear in the next section, for our registration application there is no need to have the most accurate parameters. For the parameter fitting we minimize the algebraic distance. Only in a validation step we will calculate the Euclidean distances.

### 3 Registration

In this section we explain how we compute the correct relative position of two patches. We need to determine six degrees of freedom, three translation parameters and three rotation parameters. Assume we found a number of shapes and their parametric representation as explained in the previous section. Because of the symmetry of the shapes, a single pair of corresponding shapes does not result in a unique transformation to align the surfaces. A pair of corresponding shapes imposes constraints on the possible transformations, but will leave some degrees of freedom. For example, when two planes are aligned, three degrees of freedom remain. The planes can still be translated in directions parallel to the plane and rotated around the plane's normal  $\mathbf{N}$ . Combining the constraints imposed by all shapes can lead to a unique solution. For example, three non-parallel planes

with different angles between them result in a unique solution. The second plane still leaves a translation along the intersection line which is solved by the third plane.

To combine the constraints imposed by all the shapes, we use a Hough space [17]. Tian and Shah used a Hough space to recover 3D motion in 2D images. They divide the input images in patches and for every patch, a candidate transformation is computed and generates a single vote in a single bin in parameter space. In our application, updating the parameter space for candidate corresponding shapes is more complicated, because many possible transformations exist and many bins have to be updated. The benefit of using a voting scheme is that no matching is needed, and matching for example partial planar parts would be difficult. For all shapes found on one surface, we update parameter space for all shapes of the same type on the other surface, unless we have additional information that can eliminate some possibilities.

For practical reasons, we do not directly work in the six dimensional parameter space. Instead, we separate rotation and translation, each in a three dimensional parameter space. We will first determine the rotation and then the translation. Separating rotation and translation has several advantages. First, a 6D parameter space would be memory and computationally expensive. Second, initially we do not know the bounds of the possible translations. After applying the rotation we can calculate the bounds for the translation.

We will describe our method by the constraints imposed by a pair of corresponding planes. The extension for cones and cylinders is straightforward. Let  $S_R$  be the reference surface and  $S_M$  the moving surface. We will determine the Euclidean transformation that, when applied to  $S_M$ , best aligns both patches. We denote the unknown rotation and translation that aligns  $S_M$  to  $S_R$  by  $\mathbf{R}_{MR}$  and  $\mathbf{T}_{MR}$  respectively. We represent a translation as  $\mathbf{T}(t_x, t_y, t_z)$  and a rotation around an axis  $\mathbf{A}(a_x, a_y, a_z)$  over an angle  $\alpha$  is represented by three angles

$$\mathbf{R}(\alpha, \beta, \gamma). \quad (7)$$

$\beta$  and  $\gamma$  represent the direction of the axis,  $\alpha$  the angle of rotation around this axis. Without loss of generality, we assume

$$a_z \geq 0, \|\mathbf{A}\| = 1, \quad (8)$$

If  $a_z < 0$ , we invert the axis  $\mathbf{A}$  and the angle  $\alpha$ .  $\beta$  represents the angle of the projection of the axis in the  $xy$  plane and the  $x$  axis,  $\gamma$  represents the angle of the axis with the  $xy$  plane. We require that

$$0 \leq \alpha < 2\pi, -\pi \leq \beta < \pi, 0 \leq \gamma \leq \frac{\pi}{2} \quad (9)$$

$\beta$  and  $\gamma$  are calculated as

$$\begin{aligned} \beta &= \arctan\left(\frac{a_x}{a_y}\right) \\ \gamma &= \arccos\left(\sqrt{a_x^2 + a_y^2}\right) \end{aligned} \quad (10)$$

### 3.1 Aligning Planes

Assume we have a pair of planes for which we will investigate the constraints imposed on the transformation parameters. Let  $P_m$  be a plane on the moving surfaces that we want to align to the plane  $P_r$  on the reference surface, each with their parametric representation

$$\begin{aligned} P_r : \quad & a_R x + b_R y + c_R z + d_R = 0 \\ P_m : \quad & a_M x + b_M y + c_M z + d_M = 0 \end{aligned} \quad (11)$$

Their respective unit normal vectors (pointing outward) are  $\mathbf{N}_R$  and  $\mathbf{N}_M$ . One possibility to determine these constraints is to use the general representation of a rotation and translation, determine how the parameters of  $P_m$  are transformed and identifying with the parameters of  $P_r$ . This leads to a system of four equations which is not easy to solve symbolically because of two reasons. First, because of the rotation, the problem becomes nonlinear in the unknown parameters. Second, we have less equations than unknowns. Since the equations can be made polynomial, Gröbner bases [1] allowed us to find a solution. However, solving for planes is hard, but solving this for quadrics becomes infeasible. We found the same solution by looking at it geometrically. The two planes can be aligned in two steps :

1. We rotate  $P_m$  such that it becomes parallel to  $P_R$ . This is the same as making the normals parallel and can be done by a rotation  $\mathbf{R}_1$  with axis  $\mathbf{A}_1 = \mathbf{N}_R \times \mathbf{N}_M$  over an angle  $\alpha_1 = \arccos(\mathbf{N}_R \cdot \mathbf{N}_M)$ .
2. We translate  $P_m$  such that it aligns to  $P_R$ . This is done by translating along  $\mathbf{N}_R$  over a distance  $|d_M| - |d_R|$ ,

$$\mathbf{T}_1 = (|d_M| - |d_R|)\mathbf{N}_R \quad (12)$$

We extend this solution to all possible solutions by adding two operations :

1. A rotation  $\mathbf{R}_p$  around the reference  $\mathbf{N}_R$ , the normal of the reference plane, over a free angle  $\alpha_p$
2. A translation  $\mathbf{T}_p(t_{xp}, t_{yp}, t_{zp})$  parallel to the reference plane. This can be expressed as

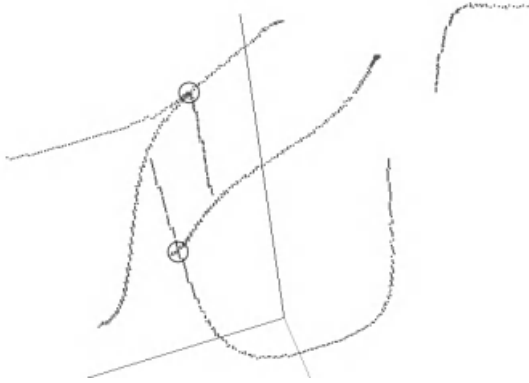
$$a_R t_{xp} + b_R t_{yp} + c_R t_{zp} = 0 \quad (13)$$

We conclude that any transformation that aligns the planes can be described as a sequence of four operations :  $\mathbf{T}_p \circ \mathbf{R}_p \circ \mathbf{T}_1 \circ \mathbf{R}_1$ . In general, rotations and translations are not commutative, but here we can change the order of  $\mathbf{R}_p$  and  $\mathbf{T}_1$ , because  $\mathbf{T}_1$  is a translation along the axis of  $\mathbf{R}_p$ . Translations and rotations can be combined to a single rotation followed by a single translation :  $\mathbf{T}_{MR} \circ \mathbf{R}_{MR}$  with  $\mathbf{T}_{MR} = \mathbf{T}_p \circ \mathbf{T}_1$  and  $\mathbf{R}_{MR} = \mathbf{R}_p \circ \mathbf{R}_1$ .

### 3.2 Determining the Rotation $\mathbf{R}_{MR}$

To find a rotation that can align all shapes and hopefully aligns the surfaces we first examine the constraint imposed by a pair of corresponding planes that have

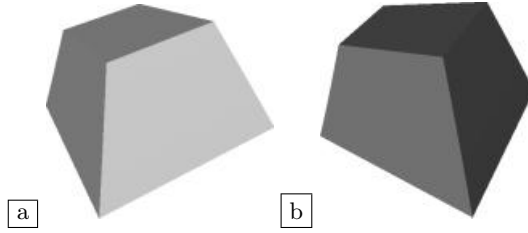
to be made parallel. For a plane pair,  $\mathbf{R}_{MR}$  has one free parameter,  $\alpha_p$ . This  $\alpha_p$  is a parameterization of a 3D curve in the parameter space  $(\alpha, \beta, \gamma)$ . This is illustrated for a simple surface with 2 planar parts in figure 3. Every possible plane pair  $(P_m, P_r)$  generates such a curve. Rotations that can align all planes are found at intersections of curves that correspond to matching plane pairs. For finding these intersections, the three dimensional rotation parameter space



**Fig. 3.** The rotation parameter space showing the constraints imposed by 2 plane pairs. This generates four mathematically closed curves in parameter space. In this visualization, the curves are not closed because the values represented are angles defined in the interval  $[0, 2\pi[$ . The circles indicate two intersection points, which correspond to the two possibilities to align two pairs of planes.

is discretized by dividing the volume defined by (9) with a rectilinear grid. This grid subdivides the volume into a number of bins. The center of the bin serves as a quantization for all points in the volume defined by the bin. The size of the bins determines the accuracy of the solution. The bins will be used to count the number of curves that visit them.

For all possible pairs  $(P_m, P_r)$ , we update the bins that are visited by their constraint curve. The free parameter  $\alpha_p$  is varied from 0 to  $2\pi$  and for all values, the corresponding  $\mathbf{R}_{MR}$  is calculated. The bin containing this rotation is updated. This process is repeated for all possible plane pairs  $(P_m, P_r)$ . The resulting parameter space is illustrated in figure 5 (a) for a surface with 6 planar parts as shown in figure 4. The bins that have been visited by the highest number of curves are used as candidate rotations. The accuracy of the rotation is limited by the resolution of the bins. For candidate rotations we do a refinement step. For such a candidate some of all the possible plane pairs can be eliminated because the candidate rotation does not align them. A Levenberg-Marquardt optimization is used in which the sum of the angles between remaining plane combinations serves as the error function.



**Fig. 4.** (a) A test surface consisting of six planes. It is a box that tapers off along  $x$ ,  $y$  and  $z$ . The surface does not have global symmetry such that a unique solution exists. (b) The surface after rotation and translation.

### 3.3 Determining the Translation $\mathbf{T}_{MR}$

We first examine the constraint on the possible translations imposed by a pair of corresponding planes. Since  $\mathbf{T}_{MR} = \mathbf{T}_1 + \mathbf{T}_p$ ,

$$\mathbf{T}_p = \mathbf{T}_{MR} - \mathbf{T}_1. \quad (14)$$

Substituting (14) in (13) results in a constraint in the parameter space for  $\mathbf{T}_{MR}(t_x, t_y, t_z)$  :

$$a_R t_x + b_R t_y + c_R t_z - a_R t_{x1} - b_R t_{y1} - c_R t_{z1} = 0, \quad (15)$$

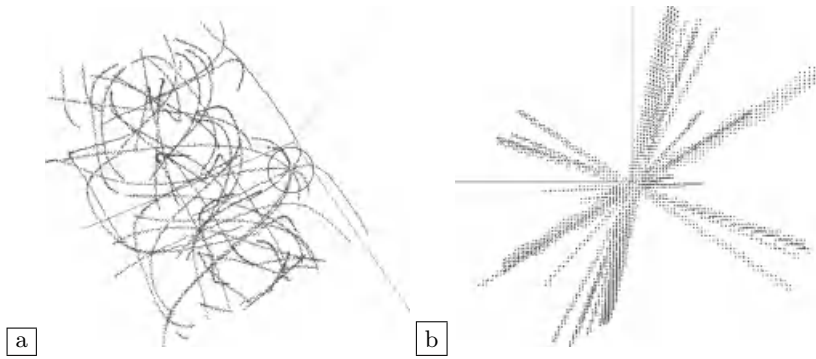
or by substituting (12) :

$$a_R t_x + b_R t_y + c_R t_z - (|d_M| - |d_R|) = 0 \quad (16)$$

This means that the constraint imposed by aligning two planes represents a plane in the translation space. The translation that is able to align all corresponding planes is found as the intersection of all their corresponding constraint planes.

To find this solution, we take a similar approach as used for finding the rotation. The translation parameter space is discretized and for every constraint plane the visited bins are updated. An update in the translation space will update more bins, since we have two degrees of freedom. An additional complication is that initially, no bounds for the solution exist. Updating is done efficiently by using a recursive subdivision scheme. We also take into account the candidate rotations found as described in the previous section. This allows us to calculate the bounds for the translation. It also allows us to reduce the number of plane pairs for which we have to update the translation parameter space. Only plane pairs are kept which, after applying the rotation, are parallel.

Figure 5 (b) shows an example of the translation space. The bins that have been visited by the highest number of constraint planes are selected. The translation represented by a bin is an approximation due to the discretization of the parameter space. However, we can perform an inexpensive refinement step. We select the constraint planes that intersect the selected bin and the intersection of the remaining constraint planes is calculated with a least square method.



**Fig. 5.** Parameter spaces for the surfaces of figure 4. (a) The orientation parameter space. The circle indicates the intersection point of 6 curves. (b) The translation parameter space. Only bins which have been visited by at least two constraint planes are shown.

### 3.4 Extension to Cones and Cylinders

After aligning two cones or cylinders, a rotational degree of freedom around their axis remains. Theoretically, the position of the top of a cone can fix the translation. However, as illustrated in figure 2, we have very partial data and often it is impossible to get an accurate estimate of the top of the cone. Because of this, the top is not used in determining the transformation and we only use the axes of cones and cylinders. Aligning these axis is very similar to aligning planes. Updating the rotation parameter space is the same as for a pair of planes, but in stead of aligning normals, the axes are aligned. The difference is that the axes do not have a direction and because of this, two possibilities for the initial rotation  $R_1$  have to be taken into account. The translation constraint imposed by a pair of axes is a straight line in translation parameter space. The point-direction representation of this constraint line is

$$\mathbf{T}(t_x, t_y, t_z) = \mathbf{X}_R - \mathbf{X}_M + t\mathbf{A}_R, \quad (17)$$

in which  $\mathbf{X}_M$  represents a point on the axis of the reference shape.  $\mathbf{X}_M$  is a point on the axis of the moving shape after applying the rotation that was computed.  $\mathbf{A}_R$  represents the axis direction of the reference shape.

### 3.5 Validation

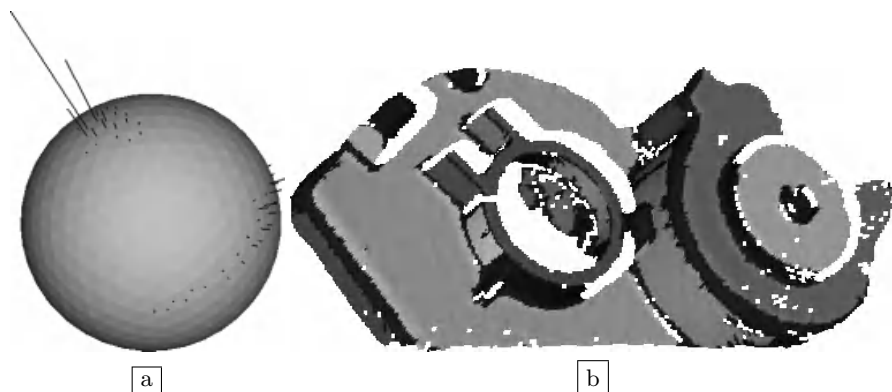
The procedure as explained in the previous sections does not always lead to a unique solution although the data presents enough information. For example, because the rotation is calculated independently, sometimes more than one rotation exists that can make all planes parallel, even though only one of them leads to the correct translation. A typical example is data containing planes in three perpendicular directions, such that three possible rotations exist. When



multiple possibilities remain, the best one is chosen by ‘measuring’ the distance between the patches after applying the corresponding transformation. The distance measure used is the number of points on  $S_M$  that find a partner point on  $S_R$  at close range.

## 4 Experiments

Figure 6 shows the orientation histogram and the shapes that are found for the surface in figure 2 (a). The shapes found provide enough constraints for registration. Figure 7(a) shows the same surface subject to a known rotation and translation. This transformation is correctly found by our method as illustrated in figure 7(b). The orientation space has  $180 \times 180 \times 45$  bins resulting in a one degree accuracy. The translation space has  $100 \times 60 \times 70$  bins. The solution is correct taking into account the discretization of the parameter space.

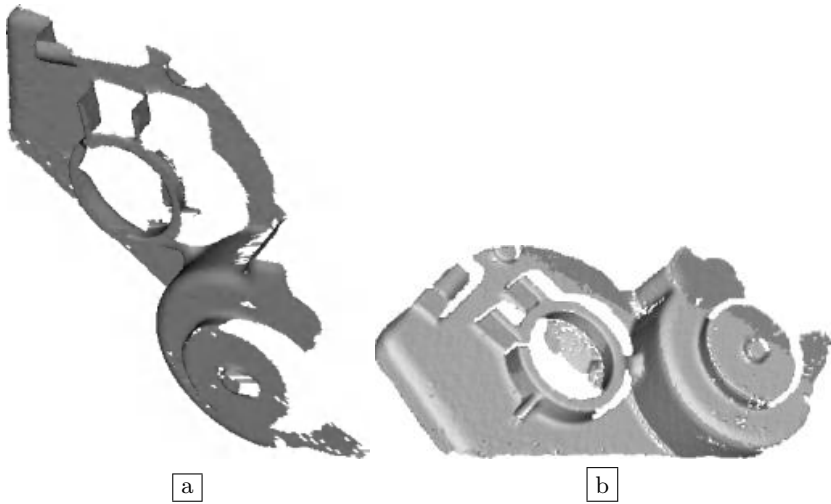


**Fig. 6.** (a) The orientation histogram for the partial reconstruction from an engine casting from figure 2 (a). The largest peak corresponds to the large parallel planar regions. On the lower side of the sphere, a circle of smaller peaks can be observed. They correspond to the conical part (b) The regions in which a shape was found (gray). The darkest parts do not belong to any shape. 13 shapes are detected on this surface. 11 are classified as planar. The most planar conical part has been incorrectly classified as planar.

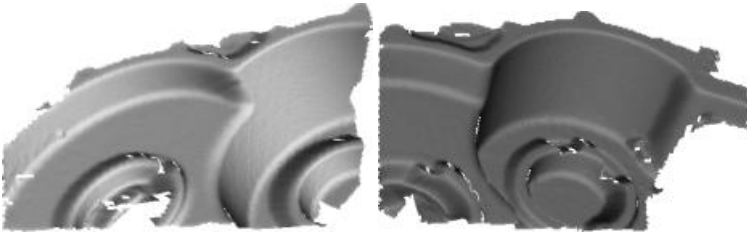
Figure 8 shows two partial reconstructions of an engine casting taken from different views. In this example, the rotation histogram results in two candidate rotations. This results in two possible solutions, shown in 9(a) and (b). The second solution is chosen in the validation step.

## 5 Summary and Conclusions

We developed a technique for the coarse registration of surface patches that is complementary to methods that look for features of interest and describe the



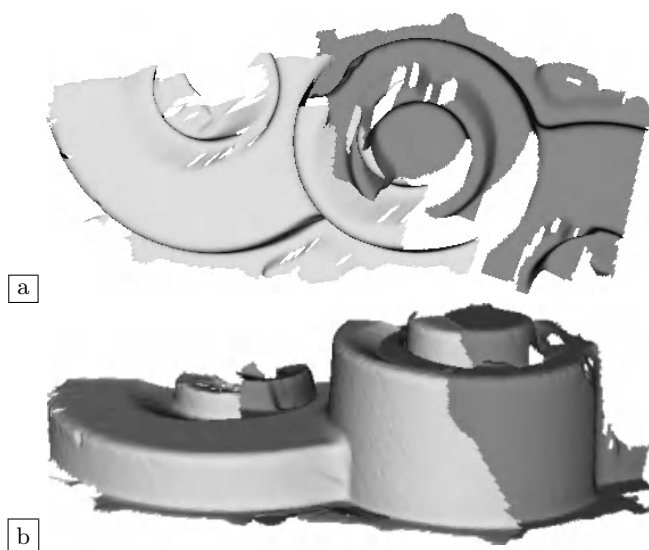
**Fig. 7.** (a) The surface of figure 2 (a), after applying a translation  $\mathbf{T}(5, 5, 5)$  and a rotation of 100 degrees around an axis with direction  $(1, 1, 1)$ . (b) Both surfaces after applying the transformation that was found.



**Fig. 8.** Two partial reconstructions of an engine casting.

local geometry around them. Such methods could fail for surfaces containing regions with local symmetries, whereas the proposed method exploits their information to perform registration. Our method uses an orientation histogram to detect regions on a surface that are planar, conical or cylindrical. A parametric surface is fitted to the points in such a region. A symmetric shape present in both patches leaves degrees of freedom but does pose constraints on the possible transformations. For combining the constraints imposed by the special regions, we use a Hough space where votes are accumulated for candidate transformations. Transformations that have high votes are validated and the transformation that best aligns both surfaces is selected. Experiments show that this method is able to position partial views that were taken over a wide baseline.

Future work will include the integration of the technique reported here with feature signature methods, which also generate candidate transformations that could be integrated in the voting scheme. The points of the surface can be split



**Fig. 9.** The solution found for the surfaces of figure 8. For this example, two possible rotations are found. These possibilities differ by a rotation of 180 degrees around the axis of symmetry of the central part. (a) Solution when the wrong rotation is used. (b) Correct solution.

based on curvature. For points with high curvature, a point signature method could be used, and the method proposed here for the low curvature points.

**Acknowledgments.** Joris Vanden Wyngaerd gratefully acknowledges support by a grant of the Flemish Institute for the Advancement of Science in Industry ‘IWT’. The authors also gratefully acknowledge support from the Belgian Federal Government IUAP4/24 “IMechS project” and the GOA project ‘VHS+’, financed by Kath.Un.Leuven.

## References

1. W.W. Adams and P. Loustau, *An Introduction to Gröbner Bases*, Graduate Studies in Mathematics, Vol. 3., American Mathematical Society, 1994
2. P. Besl, N. McKay, *A method of registration of 3-D shapes*, IEEE Trans. PAMI 12 (2) pp. 239-256, 1992
3. R. M. Bolle and D. B. Cooper. *On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data*. IEEE Trans. PAMI 8 (5) pp. 619-638, 1986
4. P. Brou, *Using the Gaussian Image to Find the Orientation of an Object*, Int’l J. Robotics Research, vol. 3, pp. 89-125, 1983
5. Y. Chen and G. Medioni, *Object modeling by registration of multiple range images*, Proc. Int. Conf. on Robotics and Automation, pp. 2724-2729, 1991

6. C.S. Chua and R. Jarvis, *Point signatures : A new representation for 3D object recognition*, Int. J. of Computer Vision, 25(1), pp. 63-85, 1997
7. J. Feldmar and N. Ayache, *Rigid, affine and locally affine registration of free-form surfaces*, TR INRIA Epidaure, No. 2220, 1994
8. J. Feldmar, N. Ayache, and F. Betting, *3D-2D projective registration of free-form curves and surfaces*, TR INRIA Epidaure, No. 2434, dec. 1994
9. H. Hebert, K. Ikeuchi, and H Delingette, *A spherical representation for recognition of free-form surfaces*, IEEE Trans. PAMI 17 (7) pp. 681, 1995
10. Berthold K. P. Horn (1987), *Closed-form solution of absolute orientation using unit quaternions*, Journal of the Optical Society of America A, 4:629-642
11. K. Ikeuchi., *Recognition of 3-D Objects Using the Extended Gaussian*, Image. In Proc. of Seventh IJCAI, pages 595-600, 1981.
12. A. Johnson and M. Hebert, *Recognizing objects by matching oriented points*, Proc. Conf. Computer Vision and Pattern Recognition, 684-689, San Juan, 1997
13. S.B. Kang and K. Ikeuchi, *3-D Object Pose Determination Using Complex EGI*, T.R. CMU-RI-TR-90-18, Robotics Institute, Carnegie Mellon University, 1990
14. Y. Liu and M.A. Rodrigues, *Essential Representation and Calibration of Rigid Body Transformations*, Machine Graphics and Vision Journal Vol 9 (2000).
15. A. P. Pentland, *Perceptual Organizations And The Representation Of Natural Form*, Artificial Intelligence, vol. 28, no. 2, pp. 293-331, 1986
16. C. Sun, and J.Sherrah, *3-D Symmetry Detection Using The Extended Gaussian Image*, IEEE Trans. PAMI 19 (2) pp. 164-168, 1997
17. Tina Y. Tian and Mubarak Shah. *Recovering 3d motion of multiple objects using adaptative hough transform*. IEEE Trans. PAMI, 19(10):1178
18. Vanden Wyngaerd, J., Van Gool, L., Koch, R., Proesmans, M., 1999. Invariant-based registration of surface patches. Proc. International Conference on Computer Vision, IEEE Computer Society Press, pp. 301-306.
19. P. Viola and W. Wells, *Alignment by maximisation of mutual information*, Proc. Int. Conf. on Computer Vision, pp. 16-23, 1995
20. N. Werghi R.B. Fisher, A. Ashbrook and C. Robertson, *Faithful Recovering of Quadric Surfaces from 3D Range Data*, Proc. 2nd Int. Conf. on 3-D Digital Imaging and Modeling, Ottawa, Canada, pp 280-289, October 1999
21. S. Yamany and A. Farag, *Free-form surface registration using surface signatures*, Int. Conf. on Computer Vision, pp. 1098-1104, 1999

# Multiview Registration of 3D Scenes by Minimizing Error between Coordinate Frames

Gregory C. Sharp<sup>1</sup>, Sang W. Lee<sup>2</sup>, and David K. Wehe<sup>3</sup>

<sup>1</sup> Department of Electrical Engineering and Computer Science  
University of Michigan, Ann Arbor, MI 48109

<sup>2</sup> Department of Media Technology  
Sogang University, Seoul, Korea

<sup>3</sup> Department of Nuclear Engineering and Radiological Sciences  
University of Michigan, Ann Arbor, MI 48109

**Abstract.** This paper addresses the problem of large scale multiview registration of range images captured from unknown viewing directions. To reduce the computational burden, we decouple the local problem of pairwise registration on neighboring views from the global problem of distribution of accumulated errors. We define the global problem over the graph of neighboring views, and we show that this graph can be decomposed into a set of cycles such that the optimal transformation parameters for each cycle can be solved in closed form. We then describe an iterative procedure that can be used to integrate the solutions for the set of cycles across the graph. This method for error distribution does not require point correspondences between views, and therefore can be used together with robot odometry or any method of pairwise registration. Experimental results demonstrate the effectiveness of this technique on range images of an indoor facility.

## 1 Introduction

To build three dimensional models using a range scanner, multiple scans are usually required due to occluded surfaces and limitations in the field of view of the scanner. These multiple scans must then be registered within a common coordinate frame before a coherent parametric description of the object can be formed. The registration of two views is most often performed through a variant of the iterative closest point registration algorithm [4,7,30,17,8,6,24,14], although methods based on matching features [10] or maximum likelihood search [28,18] are also often used. However, registration of more than two views is a somewhat more difficult problem, and there is not yet consensus on the best approach. Chen and Medioni, and Masuda both incrementally register views against a growing global union of view points [7,17]. Pulli also performs incremental registration against a growing set, but includes a backtracking step when global error becomes unacceptable [21]. Pennec describes a method that alternates between computing an average shape for the set of images, and registration of the scans against the mean shape [19]. Bergevin et al. place all views into a global frame of

reference, and then repeatedly select a view and register it against all others [3]. Blais and Levine use simulated annealing to simultaneously minimize a cost metric based on the total distance between all matches in all views [5]. Stoddart et al. find pairwise correspondences between points in all views, and then iteratively minimizing correspondence errors over all views using a descent algorithm [27]. Eggert et al. improve on this technique, adding special processing for boundary points and a multiresolution framework [9]. Williams and Bennamoun suggested a further refinement by including individual covariance weights for each point [13]. Sawhney et al. and Shum and Szeliski perform the global alignment of two dimensional mosaics by non-linear minimization of distance between point correspondences [22,26]. Benjemaa and Schmitt propose a nonlinear solution based on the quaternion representation [1]. Their formulation is a multiview extension of the pairwise solution proposed by Horn [11], and using distance between pairwise correspondences as the optimization criteria.

When there are large numbers of views, or when information such as odometry is used in conjunction with point correspondences, the global registration parameters can be solved as the parameters that minimize error with respect to the estimates of the relative motion between view pairs. Lu and Milios solve this problem by linearizing the rotational component [16]. This formulation is useful when the total rotational error is small. In this paper, we propose an analytic method for solving the global registration parameters using the relative motion between view pairs as the error criteria. This criteria does not require linearization, and therefore can be used even when the accumulated rotational error is large. Furthermore, since this criteria does not require point correspondences, our multiview registration method can be used together with robot odometry and any pairwise registration. Our overall approach is to build up a graph that describes the neighboring of views, and then decompose the graph into basis cycles. We solve the nonlinear system over each basis cycle in closed form, and the solutions for the constituent basis cycles are merged using an averaging technique.

## 2 Error Distribution in a Cycle

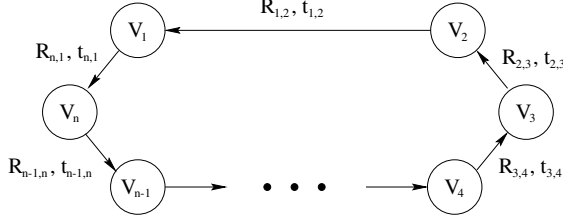
We shall take as given that the translation can be decoupled from the rotation, and solved using a linear method [16,1,23]. This effectively decouples the rotation and translation problem, allowing us to analyze rotation and translation separately. For the remainder of the paper, we consider only the rotations.

Let the captured views lie within coordinate frames  $\mathbf{V}_1, \dots, \mathbf{V}_n$ , where two frames  $\mathbf{V}_i$  and  $\mathbf{V}_j$  are related by the rotation and translation  $(\mathbf{R}_{i,j}, \mathbf{t}_{i,j})$ . A point  $\mathbf{p}_j$  in frame  $\mathbf{V}_j$  can be described in frame  $\mathbf{V}_i$  according to the change of coordinates

$$\mathbf{p}_i = \mathbf{R}_{i,j}\mathbf{p}_j + \mathbf{t}_{i,j}. \quad (1)$$

This change of coordinates between neighboring views can be found using a pairwise registration procedure; we shall call these *measurements*. When each

view is connected only to its immediate neighbors in a cycle of views (see fig. 1, the measurements  $\{(\mathbf{R}_{1,2}, \mathbf{t}_{1,2}), \dots, (\mathbf{R}_{n,1}, \mathbf{t}_{n,1})\}$  are an overparameterization of the space with six extra degrees of freedom, and therefore the composition of these changes of coordinates about the cycle may not be identity. We define a cycle *consistent* if its associated rotations or rigid transformations compose to identity. Our goal is to find a new set of rigid transformations, which we call *estimates* and denote  $\{(\hat{\mathbf{R}}_{1,2}, \hat{\mathbf{t}}_{1,2}), \dots, (\hat{\mathbf{R}}_{n,1}, \hat{\mathbf{t}}_{n,1})\}$ , that is consistent and that minimizes the mean squared error in frame space between the estimate and measurement.



**Fig. 1.** A cycle of view-pairs where view  $V_1$  has been aligned with  $V_2$ ,  $V_2$  aligned with  $V_3$ , and so on.

Considering only rotation, the estimates that minimizes the mean squared angular error between the estimate and measurement can be solved in closed form [23]. Define the matrix  $\mathbf{E}_{k,k+1}$  to be the rotation such that

$$\mathbf{R}_{1,2} \cdots \mathbf{R}_{k,k+1} \mathbf{E}_{k,k+1} \mathbf{R}_{k+1,k+2} \cdots \mathbf{R}_{n,1} = \mathbf{I}.$$

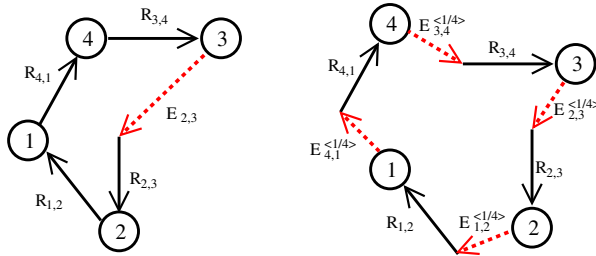
That is,  $\mathbf{E}_{k,k+1}$  removes the error that occurs when composing the measurement rotations about the cycle from frame  $k+1$  all the way back to frame  $k+1$  (see fig. 2 left). We represent fractional portions of this error matrix using the notation  $\mathbf{E}_{k,k+1}^{<\alpha>}$ , where  $\mathbf{E}_{k,k+1}^{<\alpha>}$  shares the same axis of rotation as  $\mathbf{E}_{k,k+1}$ , but the angle of rotation has been scaled by  $\alpha$ . The best estimate for the rotation  $\mathbf{R}_{k,k+1}$  is given by

$$\hat{\mathbf{R}}_{k,k+1} = \mathbf{E}_{k-1,k}^{<1/n>} \mathbf{R}_{k,k+1} = \mathbf{R}_{k,k+1} \mathbf{E}_{k,k+1}^{<1/n>}. \quad (2)$$

This idea is shown in the right half of fig. 2. This estimate is both correct, in the sense that it commutes to identity about the cycle, and optimal in the sense of minimizing mean squared angular error with respect to the measurements [23].

### 3 Error Distribution in a Graph

Given that we know how to find optimal relative rotations which satisfy the consistency constraint about a single cycle, we now consider a finite number of views connected pairwise by edges in a graph structure. We seek to satisfy



**Fig. 2.** (Left)  $\mathbf{E}_{k,k+1}$  applied to  $\mathbf{R}_{k,k+1}$  create a consistent cycle. (Right)  $\mathbf{E}_{k,k+1}^{<1/n>}$  applied to each  $\mathbf{R}_{k,k+1}$  creates an optimal set of estimates.

the consistency constraint about all circuits in the graph while minimizing error from the initial measurements. We limit our attention to finite, connected, and simple undirected graphs (i.e. having neither loops nor parallel edges), for which each edge lies within at least one cycle. Graphs that are not connected may be processed component by component, and edges that do not lie within any cycle may be given estimates equal to their initial measurements.

Our method for distributing the rotation error is described in algorithm 1. First, choose a set of cycles that form a cycle basis, which will be described in proposition 1. Next, distribute the error about each cycle independently, recording the best rotation for each edge. Finally, since some edges may be contained in more than one cycle, average all rotation estimates for a given edge [20]. Because of this averaging, consistency is not guaranteed. However, it will be shown in proposition 2 that a repetition of this process will reduce the inconsistency to zero.

We define a *circuit* to be a closed path that may contain repeated edges or nodes, and a *cycle* to be a circuit that contains neither repeated edges nor repeated nodes. From graph theory we know that any circuit in a graph can be represented as a linear combination of a set of *basis cycles* in the edge space of the graph  $\mathcal{T}$  [2]. This result is used to prove the validity of loop analysis in circuit theory, where the superposition principle allows currents to be added together in linear combinations. For non-commutative groups such as rotations, we establish a slightly weaker result to guarantee consistency throughout all circuits in the graph.

**Proposition 1** *Let  $\mathcal{T}$  be a spanning tree for a graph  $\mathcal{G}$ , and let  $\mathcal{B}$  be the set of cycles formed by adding a single edge not in  $\mathcal{T}$  to  $\mathcal{T}$ . If the cycles in  $\mathcal{B}$  are consistent then all circuits in  $\mathcal{G}$  are consistent.*

**Proof:** Let us represent a circuit as a string, each token in the string representing a node traversed from start to finish. The initial and final token in the string must be the same. For example, the circuit going from  $a$  to  $b$  to  $c$  back to  $a$  would be represented  $abca$ . The circuit  $abca$  is consistent if  $\mathbf{R}_{a,b}\mathbf{R}_{b,c}\mathbf{R}_{c,a} = \mathbf{I}$ .



We define four operations that operate on circuits: (a) node insertion, (b) node deletion, (c) circuit insertion, and (d) circuit deletion. If  $X$  and  $Y$  are substrings,  $a$ ,  $b$ , and  $c$  are nodes, and  $C$  is a circuit that starts and finishes on  $c$ , these operations are described grammatically as follows:

$$XabaY \leftarrow XaY \quad (a)$$

$$XaY \leftarrow XabaY \quad (b)$$

$$XC Y \leftarrow XcY \quad (c)$$

$$XcY \leftarrow XC Y \quad (d).$$

Node insertion and node deletion preserve the consistency of a circuit, since going  $a$  to  $b$  back to  $a$  inserts or deletes  $\mathbf{R}_{ab}\mathbf{R}_{ba} = \mathbf{I}$  into the composition of rotations. Similarly, if the circuit  $C$  is consistent, then circuit insertion and circuit deletion preserve the consistency of a circuit because the operation inserts or deletes  $\mathbf{R}_C = \mathbf{I}$ .

Now suppose that we are given an initial graph which is a spanning tree. All circuits in the graph can be formed by a sequence of node insertion operations from of a single node and therefore the graph is consistent. Proceeding inductively, we assume that a partial graph containing the initial spanning tree is consistent, and we add an edge  $ab$ . The edge  $ab$ , when added to the initial spanning tree, forms a single cycle  $C = ab\dots a$  in  $\mathcal{B}$ . By hypothesis, both  $C$  and its reverse circuit  $ba\dots b$  are consistent.

Consider any circuit in this new partial graph. By a sequence of node deletions, it is equivalent to a circuit that contains substrings of  $a$  and  $b$  of length at most two. For example, two node deletions applied to the substring  $ababab$  yields  $ab$ , which has length of two. Next, each substring of length two can be modified, using circuit expansion with  $C = ab\dots a$  (or  $ba\dots b$ ) into a substring that contains two substrings of  $a$  and  $b$ , one of length three and one of length one. For example,  $xbay$  is equivalent to  $xbCy = xbab\dots ay$ . Finally, we apply node deletion to the length three substrings of  $a$  and  $b$  to find an equivalent circuit that contains substrings of  $a$  and  $b$  of length at most one. This final circuit does not contain the substring  $ab$  or  $ba$ , and therefore was a circuit in the partial graph before adding the new edge. Hence all circuits in the new partial graph are consistent, and by induction  $\mathcal{G}$  is consistent. ■

This proposition is weaker than the results found using superposition. Although  $\mathcal{B}$  is indeed a set of basis cycles [2], there are some sets of basis cycles that cannot be constructed using the spanning tree method. However, this property is not restricted to graphs of rotations. This property holds for graphs of any non-commutative group, including affine and projective transforms.

**Proposition 2** *The rotation error distribution algorithm (algorithm 1) converges to a consistent graph.*

**Proof:** Let the graph have  $n$  edges and  $k$  basis cycles. Consider some edge  $i$ , with measurement  $\mathbf{R}_i$ , and estimates from  $m > 1$  different cycles:  $\hat{\mathbf{R}}_i^1, \dots, \hat{\mathbf{R}}_i^m$ .

The measurement error of edge  $i$  in cycle  $j$  before distribution is  $\angle \hat{\mathbf{R}}_i^j(\mathbf{R}_i)^{-1}$ . When we average the estimates to find  $\hat{\mathbf{R}}_i$ , because the averaging minimizes deviation from the estimates, its total contribution to deviation over all cycles is decreased

$$\sum_{j=1}^m \angle \hat{\mathbf{R}}_i^j(\hat{\mathbf{R}}_i)^{-1} < \sum_{j=1}^m \angle \hat{\mathbf{R}}_i^j(\mathbf{R}_i)^{-1}.$$

Furthermore, the edges that are not in the spanning tree now contribute zero deviation. Hence, the deviation is reduced by at least a fixed percentage per iteration, and the sum deviation from identity about all basis cycles monotonically decreases. From proposition 1, we conclude that this procedure converges to a consistent graph. ■

---

**Algorithm 1** Error Distribution

---

Given graph, basis cycles, and measurements

$\mathcal{R}^0 \leftarrow$  measurements

**repeat**

**for all** basis cycles in graph **do**

    Use  $\mathcal{R}^{i-1}$  to compute rotation estimates for each edge in cycle

**end for**

**for all** edges in graph **do**

    Average all rotation estimates for this edge and add to  $\mathcal{R}^i$

**end for**

**until** convergence

---

## 4 Multiview Registration

The global error distribution method described in section 3 is used together with any pairwise registration algorithm to perform completely automatic globally consistent multiview range image registration. As described in algorithm 2, the pairwise registration method is run for a while to find locally good fits for each pair. Then the error distribution method is run to convert the local fits into a global fit. This global fit then provides feedback into the local algorithm, giving it new initial conditions for the next set of local fits.

## 5 Experimental Results

Figure 3 shows 9 views from an sequence of range images taken from on board of a mobile robot in an indoor environment. The range sensor is a structured light range camera that uses a DLP data projector and CCD camera to compute range points as described by Trobina [29]. The range of the sensor is between

---

**Algorithm 2** Multiview Registration

---

```

Let  $\mathcal{T}$  be the initial set of pairwise transformations over a graph of views
repeat
    Use pairwise registration to update  $\mathcal{T}$ 
    Use error distribution to update  $\mathcal{T}$ 
until convergence

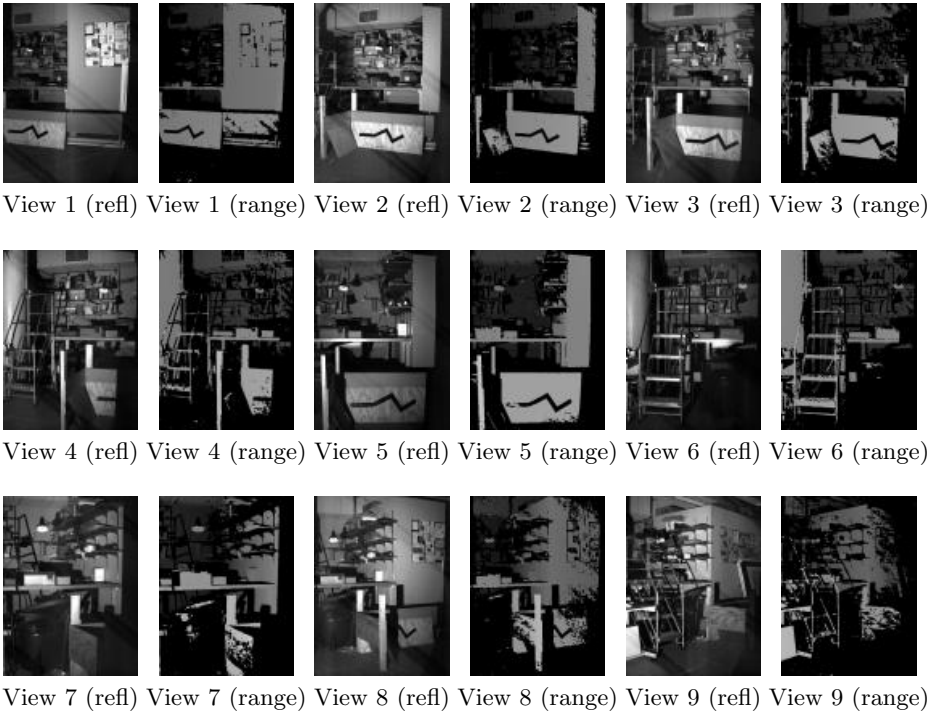
```

---

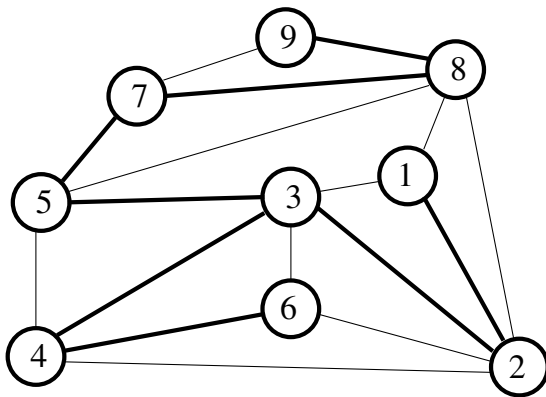
about 3 to 8 meters, with a noise standard deviation of about 1 cm at 5 meters. The total volume captured by the 9 views is about  $5 \times 5 \times 5$  meters.

A graph of neighboring views, shown in figure 4, was constructed manually for these images. In future work we wish to consider automatic construction of the graph, building on the work of Sawhney et al. [22], Kang et al. [15], and Huber [12]. After the graph is found, a spanning tree for the graph was found, as shown in bold on the graph. These neighboring views were registered pairwise using a maximum likelihood matching procedure that uses both valid and missing data points to align the pair of views [25].

The final output of three iterations of algorithm 2 is shown in figure 5, where each of the nine views is rendered in a different color. Errors due to registration



**Fig. 3.** Reflectance and range images of an indoor scene.



**Fig. 4.** Graph of neighboring views for range images in figure 3. The spanning tree for the graph is shown in bold.

errors are consistently lower than 0.5 cm, half of the noise standard deviation of the sensor.

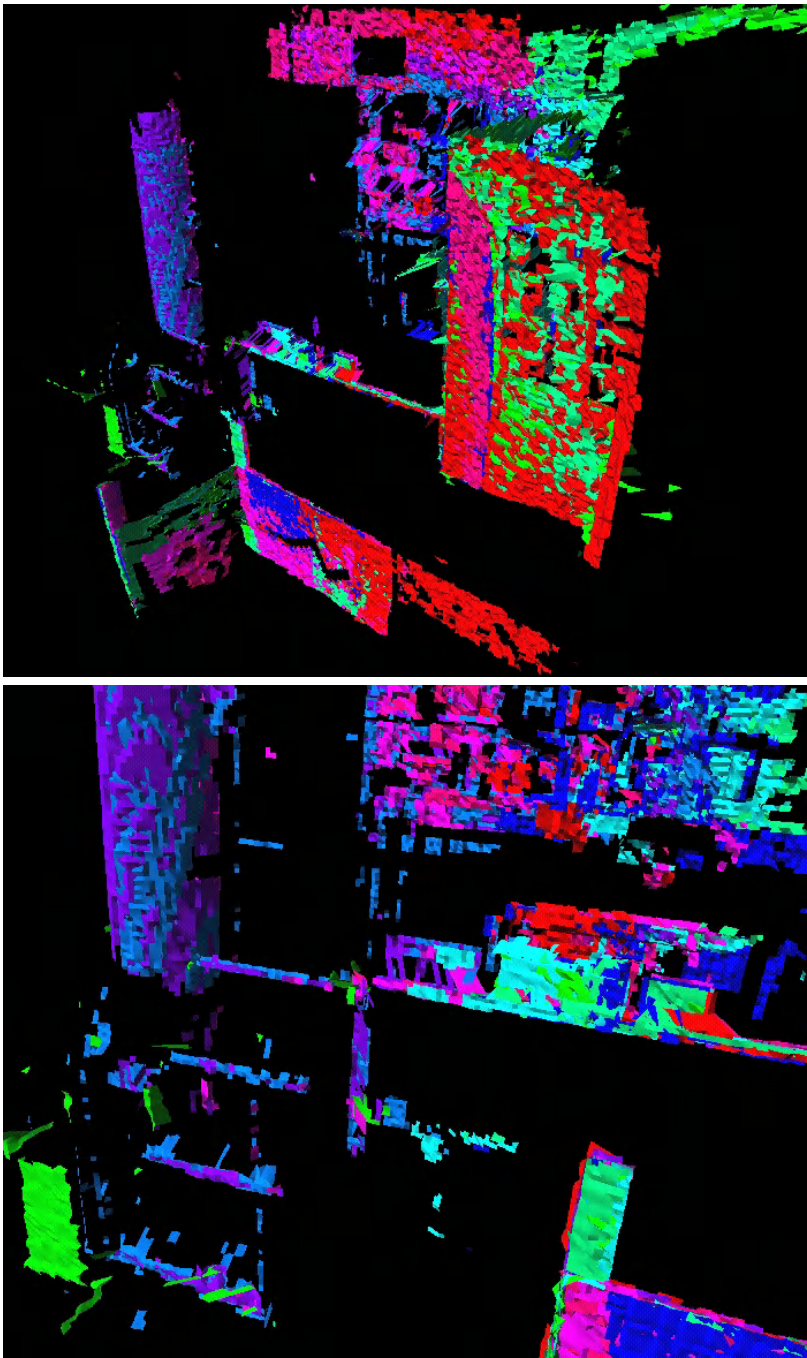
## 6 Discussion

The error distribution method described in equation 2 is easily modified to perform error distribution on a simply weighted graph, where the weights are real valued coefficients that reflect our relative confidence in the rotation estimates. However, it is not clear how this framework can be modified to include covariance, or unequal weights for different directions of rotation. Using real covariance matrices is possible, but this implies linearization of the rotations which limits utility to small rotational error. Using point correspondences to determine the covariance is also problematic. Because the correspondences are only valid in a small region, any covariance information obtained through point correspondences is suspect.

The chief benefit of the graph representation is that it allows updates to proceed locally on each cycle. We believe that this approach may offer insight into related high dimensional sensing problems, including multiview two dimensional image registration, structure from motion, and multisensor fusion. Related sensor readings can be put into a graph structure, and cycles within the graph can be analyzed independently. The nonlinear analysis will be facilitated by the simpler structure of a single cycle compared to the entire graph.

## 7 Summary

We have described a general multiview registration algorithm that distributes pairwise error accumulated over multiple views back to its constituent views in



**Fig. 5.** (Top) Overview of registration results. Each of the nine views is rendered in a different color. (Bottom) Detail of registration results near the step ladder.

a fair manner. Any pairwise registration algorithm may be used to generate the estimates of relative motion between each pair of views, and the accumulated error is distributed to all views. By posing the problem as the minimization of error between coordinate frames, this method is useful for large scale registration problems that may involve hundreds of views.

**Acknowledgments.** This material is based upon work supported by DOE under Award No. DE-FG04-86NE37969.

## References

1. R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3-d point sets using unit quaternions. In *European Conference on Computer Vision*, 1998.
2. C. Berge. *The Theory of Graphs and its applications*. Barnes & Noble, Inc., 1962.
3. R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, May 1996.
4. P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
5. G. Blais and M.D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, August 1995.
6. C.S. Chen, Y.P. Hung, and J.B. Cheng. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234, November 1999.
7. Y. Chen and G.G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.
8. C. Dorai, J. Weng, and A.K. Jain. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1131–1138, October 1997.
9. D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Computer Vision and Image Understanding*, 69(3):253–272, March 1998.
10. O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotics Research*, 5(3):27–52, 1986.
11. B.K.P. Horn. Closed form solutions of absolute orientation using unit quaternions. *Journal of the Optical Society of America-A*, 4(4):629–642, April 1987.
12. D.F. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *International Conference on 3D Digital Imaging and Modeling*, pages 153–160, May 2001.
13. Williams J. and Bennamoun M. A multiple view 3d registration algorithm with statistical error modeling. *IEICE Transactions on Information and Systems*, 83-D(8):1662–1670, August 2000.
14. A.E. Johnson. Surface landmark selection and matching in natural terrain. In *Computer Vision and Pattern Recognition*, volume 2, pages 413–420, 2000.

15. E.Y. Kang, I. Cohen, and G.G. Medioni. A graph-based global registration for 2d mosaics. In *International Conference on Pattern Recognition*, pages Vol I: 257–260, 2000.
16. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.
17. T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, May 1995.
18. C.F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, February 2000.
19. X. Pennec. Multiple Registration and Mean Rigid Shapes - Application to the 3D case. In *Image Fusion and Shape Variability Techniques (16th Leeds Annual Statistical (LASR) Workshop)*, pages 178–185, july 1996.
20. X. Pennec. Computing the mean of geometric features: Application to the mean rotation. Technical Report RR-3371, INRIA, 1998.
21. K. Pulli. Multiview registration for large data sets. In *International Conference on 3D Digital Imaging and Modeling*, pages 160–168, 1999.
22. H.S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *European Conference on Computer Vision*, pages 103–119, 1998.
23. G.C. Sharp, S.W. Lee, and D.K. Wehe. Toward multiview registration in frame space. In *IEEE International Conference on Robotics and Automation*, 2001.
24. G.C. Sharp, S.W. Lee, and D.K. Wehe. Invariant features and the registration of rigid bodies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102, January 2002.
25. G.C. Sharp, S.W. Lee, and D.K. Wehe. Registration of range images in the presence of occlusions and missing data. Technical report, University of Michigan, 2002.
26. H.Y. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, February 2000.
27. A.J. Stoddart and A. Hilton. Registration of multiple point sets. In *International Conference on Pattern Recognition*, page B6A.5, 1996.
28. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.
29. M. Trobina. Error model of a coded-light range sensor. Technical Report BIWI-TR-164, Communication Technology Laboratory, Image Science Group, ETH Zurich, 1995.
30. Z.Y. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, October 1994.

# Recovering Surfaces from the Restoring Force

George Kamberov<sup>1</sup> and Gerda Kamberova<sup>2</sup>

<sup>1</sup> Stevens Institute of Technology, Hoboken, NJ 07030, USA

kamberov@cs.stevens-tech.edu

<sup>2</sup> Hofstra University, Hempstead, NY 11549, USA

kamberov@grasp.cis.upenn.edu

**Abstract.** We present a new theoretical method and experimental results for direct recovery of the curvatures, the principal curvature directions, and the surface itself by explicit integration of the Gauss map. The method does not rely on polygonal approximations, smoothing of the data, or model fitting. It is based on the observation that one can recover the surface restoring force from the Gauss map, and (i) applies to orientable surfaces of arbitrary topology (not necessarily closed); (ii) uses only first order linear differential equations; (iii) avoids the use of unstable computations; (iv) provides tools for filtering noise from the sampled data. The method can be used for stable extraction of surfaces and surface shape invariants, in particular, in applications requiring accurate quantitative measurements.

## 1 Introduction

In this paper we consider a classical computer vision problem: to what extent can we determine a surface and its properties from its Gauss map. We show that given the Gauss map  $\mathbf{N}$  by solving a linear system of first order differential equations we can extract the mean curvature function  $H$  of the surface without building a surface model or a surface parameterization. Then we turn around and use  $H$  and  $\mathbf{N}$  to determine the Gauss curvature, the principal curvature axes, and ultimately a global parameterization of the surface. The presented method works for orientable surfaces of arbitrary topology. The method is based on the solution of first order linear differential equations and explicit quadratic expressions. This leads to computationally stable discretization of the method. A discrete version of the presented method is used to recover the curvatures, the principal axes, and parameterizations from clouds of points and normals sampled from actual surfaces.

Surface parameterizations are a convenient tool for analyzing surface properties. They are used in many computer vision tasks, for example, in matching and for the computation of surface invariants. One typical approach is to, first, collect a sufficiently dense set of sampled surface points using range sensors, stereo, MRI, or CT techniques; second, to approximate the parameterization from an estimated 3D surface model; and third, to compute the differential invariants from the approximated surface. The 3D model is obtained by applying a marching cubes method, a Delaunay triangulation, or some model fitting or smoothing



technique, [6,7,18,2,14,17,19]. The typical approaches do not come with robust and general error estimates. Another approach is to extract the curvature invariants directly from range, stereo, or photometric data without building the 3D model [6]. Other powerful methods for obtaining global parameterizations of closed surfaces involve superquadrics, deformable superquadrics, Brechbüller's constrained optimization algorithm for surfaces with spherical topology [13,4].

Ultimately with all these methods differential invariants are computed using classical differential geometric methods. They involve taking second order derivatives, and solving general characteristic polynomials. An additional source of errors is the computational instability of the methods. For example, to compute the principal curvature vectors and the principal curvatures, the methods rely on diagonalizing general symmetric matrices (in fact the operators are often only close to symmetric due to noise and round off errors). The standard diagonalization routines introduce additional errors.

We set out to develop a method which (i) applies to orientable surfaces of arbitrary topology (not necessarily closed); (ii) uses only first order linear differential equations; (iii) avoids the use of unstable computations; (iv) provides tools for filtering noise from the sampled data.

The principal decision is what should be the data used to derive the surface invariants and the parameterization. We chose the Gauss map of the surface as the primary data from which everything else is inferred. The motivation for this choice stems from geometry, physical intuition, and the current practice in computer vision. Furthermore, the Gauss map is a first order invariant and so takes an intermediate position between the parameterization and the curvature invariants of a surface (one derivative in each direction).

The geometric motivation will be explained further in the next section but the basic idea is that a generic surface is determined up to scaling and translation by its Gauss map. The physics intuition is based on the realization that one can obtain the mean curvature function  $H$  of a surface by solving a differential equation involving only the Gauss map  $\mathbf{N}$ . Thus given  $\mathbf{N}$  we can get the vector field  $H\mathbf{N}$ . Ever since the times of Laplace and Young it is known that if we think of the surface as an isotropic membrane with constant surface tension, then up to multiplication by a constant the vector field  $H\mathbf{N}$  determines completely the restoring force which shapes the surface [16]. Finally the extraction of the Gauss map of a surface is a staple of computer vision.

Motivated by these observations we take a new approach to surface and surface shape recovery. First we remove outliers using integrability conditions, then we compute the mean curvature directly from the Gauss map, and then turn around and compute as many attributes as possible before we finally recover the immersion, instead of following the usual path, recover the immersion and then compute the rest of the attributes. The point in our approach is to reduce round off errors and other numeric noise, and also to exploit other useful ingredients which one can extract directly from the data set. An alternative discrete procedure which avoids solving differential equations is presented in [10].

## 2 Parameterized Surfaces in $\mathbf{R}^3$

Here we outline the necessary background theory and terminology. For a detailed exposition see [5]. A parameterized surface,  $S$ , in space is a vector-valued map,  $\mathbf{f}$ , from some two-dimensional domain  $M$  into Euclidean three space:

$$\mathbf{f} : M \rightarrow \mathbf{R}^3, \quad S = \mathbf{f}(M).$$

The domain  $M$  is often chosen to be a planar region endowed with some coordinates  $(u, v)$  but one can use any smooth 2D manifold. The differential,  $d\mathbf{f}_p$ , of  $\mathbf{f}$  at a point  $p \in M$  is a linear map that maps tangent vectors to tangent vectors, i.e., if  $\mathbf{u}$  is the velocity (tangent) vector to a curve in  $M$ ,  $d\mathbf{f}_p(\mathbf{u})$  is the velocity (tangent) vector to the image of that curve in  $S = \mathbf{f}(M)$ . Thus

$$d\mathbf{f}_p : T_p(M) \rightarrow T_{\mathbf{f}(p)}(S) \subset \mathbf{R}^3$$

where  $T_p(M)$  denotes the tangent plane to the abstract surface  $M$  and  $T_{\mathbf{f}(p)}(S)$  is the range  $d\mathbf{f}_p(T_p(M))$  of the differential  $d\mathbf{f}_p$ , respectively. The tangent plane  $T_p(M)$  to a surface  $M$  at a point  $p$  is the linear space, that best approximates the surface at  $p$ . It is customary to omit the subscript  $p$  when discussing the differential or the tangent plane, and so we do.

The map  $\mathbf{f}$  is an immersion if its differential  $d\mathbf{f}$  is an isomorphism and we say that  $S$  is an immersed parameterized surface.

### 2.1 Oriented Surfaces

In this paper, we consider only oriented surfaces, that is, there is a consistent way of identifying positively oriented frames in the tangent plane. (See Chapter 2-6 in [5].) Intuitively, a surface is oriented if one has chosen a counterclockwise direction of rotation in all tangent planes.

If  $S$  is an immersed parameterized surface and the domain  $M$  of the parameterization  $\mathbf{f}$  is oriented, then we can define a continuous unit vector field

$$\mathbf{N} : M \rightarrow \mathbf{R}^3$$

such that  $\mathbf{N}(p)$  is perpendicular to the plane  $T_{\mathbf{f}(p)}(S) \subset \mathbf{R}^3$  for every point  $p$  in the domain  $M$ . The map  $\mathbf{N}$  is called the Gauss map of the immersion.

The Gauss map  $\mathbf{N}$  (the surface normal), the Gauss curvature, the mean curvature, and all other differential invariants are expressed in terms of the map  $\mathbf{f}$  and its derivatives.

### 2.2 The Gauss Map

If  $M$  is an abstract oriented two dimensional manifold then the value of the Gauss map at a point  $p \in M$  is defined by

$$\mathbf{N} = \frac{1}{\|d\mathbf{f}(\mathbf{v}_1) \times d\mathbf{f}(\mathbf{v}_2)\|} d\mathbf{f}(\mathbf{v}_1) \times d\mathbf{f}(\mathbf{v}_2)$$

where  $(\mathbf{v}_1, \mathbf{v}_2)$  is a positively oriented frame of the tangent plane  $T_p(M)$ . Here  $\times$  is the usual cross product in  $\mathbf{R}^3$ . In particular, if  $M$  is a planar domain with a fixed coordinate system  $(u, v)$ , then

$$d\mathbf{f} = \frac{\partial \mathbf{f}}{\partial u} du + \frac{\partial \mathbf{f}}{\partial v} dv,$$

and the Gauss map is the vector-valued function

$$\mathbf{N} = \frac{1}{\|\frac{\partial \mathbf{f}}{\partial u} \times \frac{\partial \mathbf{f}}{\partial v}\|} \frac{\partial \mathbf{f}}{\partial u} \times \frac{\partial \mathbf{f}}{\partial v}. \quad (1)$$

In general, it is convenient to think of the Gauss map as a map from  $M$  to the unit sphere,  $\mathbf{S}^3$ ,

$$\mathbf{N} : M \rightarrow \mathbf{S}^3 \subset \mathbf{R}^3.$$

In our examples we use the fish-scales method designed by Šára and Bajcsy in [14] to extract samples of the Gauss map of surfaces in  $\mathbf{R}^3$ .

### 2.3 A Conformal Structure and a Complex Structure Induced by a Parameterization

A conformal structure on a surface is a choice of angles between tangent vectors. On an oriented surface, a conformal structure is equivalent to defining the operation,  $J$ , of rotating tangent vectors by ninety degrees counterclockwise in the tangent plane. This operation is also called a complex structure. For the general theory see [11].

A surface parameterization,  $\mathbf{f} : M \rightarrow \mathbf{R}^3$ , defines a complex structure  $J_f$  on the domain  $M$ . Indeed, let  $\mathbf{v}$  be a vector tangent to  $M$  at some point  $p \in M$ , then  $J_f(\mathbf{v})$  is the unique vector tangent to the domain satisfying

$$d\mathbf{f}(J_f(\mathbf{v})) = \mathbf{N} \times d\mathbf{f}(\mathbf{v}).$$

Thus the defining relation for the complex structure  $J_f$  is

$$d\mathbf{f} \circ J_f = \mathbf{N} \times d\mathbf{f}. \quad (2)$$

Suppose that  $M$  is an abstract oriented 2D manifold equipped with a complex structure  $J$ . A surface immersion  $\mathbf{f} : M \rightarrow \mathbf{R}^3$  is called a conformal immersion if the induced complex structure coincides with the abstract complex structure  $J$ ,  $J_f = J$ .

### 2.4 The Differential Invariants: Mean Curvature, Gauss Curvature, Principal Axes, and Principal Curvatures

Recall that the second fundamental form of  $\mathbf{f}$  is a symmetric quadratic form defined by

$$\mathbb{I}(\mathbf{u}, \mathbf{v}) = - \langle d\mathbf{N}(\mathbf{u}) | d\mathbf{f}(\mathbf{v}) \rangle$$

where  $\langle \cdot | \cdot \rangle$  is the Euclidean scalar product in  $\mathbf{R}^3$ . At every point  $p \in M$  there exists a positively oriented orthonormal frame  $\{\mathbf{e}_1, \mathbf{e}_2 = J_f(\mathbf{e}_1)\}$ ,  $\|d\mathbf{f}(\mathbf{e}_i)\| = 1$ , of  $T_p(M)$  in which the symmetric quadratic form  $\mathbb{I}(\cdot, \cdot)$  is represented by a diagonal matrix

$$\begin{pmatrix} \mathbb{I}(\mathbf{e}_1, \mathbf{e}_1) & \mathbb{I}(\mathbf{e}_1, \mathbf{e}_2) \\ \mathbb{I}(\mathbf{e}_2, \mathbf{e}_1) & \mathbb{I}(\mathbf{e}_2, \mathbf{e}_2) \end{pmatrix}$$

where

$$\begin{aligned} \mathbb{I}(\mathbf{e}_1, \mathbf{e}_2) &= 0 \\ \mathbb{I}(\mathbf{e}_2, \mathbf{e}_1) &= 0 \\ \mathbb{I}(\mathbf{e}_1, \mathbf{e}_1) &= \kappa_1 \\ \mathbb{I}(\mathbf{e}_2, \mathbf{e}_2) &= \kappa_2. \end{aligned} \tag{3}$$

The vectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are called principal curvature vectors, they define the principal axes, and the numbers  $\kappa_1, \kappa_2$  are the principal curvatures. The mean curvature,  $H$  is the average of the principal curvature, and the Gauss curvature is the product of the principal curvatures.

### 3 Computing the Differential Invariants: Theory

We now present the theoretical results for computing from the Gauss map and the conformal structure, the mean curvature, the differential of the immersion, the Gauss curvature, the principal axes, and finally the immersion itself. All proofs are in the Appendix. These results underline our new computational strategy: clean up outliers in the Gauss map using Theorem 3.3, compute  $H$  directly from  $N$ , then proceed to compute the rest. (For motivation see the last paragraphs of Section 3.2.)

**Theorem 3.1.** *Let  $\mathbf{N}$  be the Gauss map of a parameterized surface  $\mathbf{f} : M \rightarrow \mathbf{R}^3$  and let  $J_f$  be the induced complex structure. If  $\mathbf{f}$  is twice continuously differentiable then the differential  $d\mathbf{N}$  of the Gauss map satisfies*

$$d\mathbf{N} = -Hd\mathbf{f} + \omega, \tag{4}$$

where  $H$  is the mean curvature, and  $\omega$  is a  $\mathbf{R}^3$ -valued one form from the tangent plane to the Euclidean three space,

$$\omega : T(M) \rightarrow \mathbf{R}^3, \tag{5}$$

such that, for every vector  $\mathbf{v}$  tangent to the domain  $M$  the image  $\omega(\mathbf{v})$  satisfies

$$\omega(\mathbf{v}) \perp \mathbf{N} \tag{6}$$

$$\omega(J_f(\mathbf{v})) = -\mathbf{N} \times \omega(\mathbf{v}). \tag{7}$$

Therefore we have the following corollaries expressing the differential invariants in terms of the Gauss map and the complex structure.

**Corollary 3.1.** (*Mean curvature*) *Let  $\mathbf{N}$  be the Gauss map of a parameterized surface  $\mathbf{f} : M \rightarrow \mathbf{R}^3$  and let  $J_f$  be the induced complex structure. If  $\mathbf{f}$  is twice continuously differentiable, then*

$$-Hd\mathbf{f} = \frac{1}{2} (d\mathbf{N} - \mathbf{N} \times d\mathbf{N} \circ J_f) \quad (8)$$

We now come to the key observation that we can compute the mean curvature of a conformal immersion directly from the Gauss map.

### 3.1 Computing the Mean Curvature

**Theorem 3.2.** *Let  $M$  be an abstract 2D manifold equipped with a complex structure  $J$ , and  $\mathbf{N}$  be the Gauss map of a conformal immersion of  $M$  in  $\mathbf{R}^3$ . Let  $\tau$  be a the one-form defined by*

$$\tau := \frac{1}{2} (d\mathbf{N} - \mathbf{N} \times d\mathbf{N} \circ J). \quad (9)$$

Then

1. *The mean curvature vanishes precisely when the form  $\tau$  is trivial, that is,  $H(p) = 0$  if and only if  $\tau_p = \mathbf{0}$ .*
2. *Away from its zero locus,  $H$  is a non-vanishing solution of the linear system of first order differential equations*

$$H d\tau = dH \wedge \tau, \quad (10)$$

where  $\wedge$  denotes the wedge product of 1-forms.

In local coordinates  $(x, y)$  we represent the form  $\tau$  as  $\tau = \tau_x dx + \tau_y dy$  and the system (10) as

$$\frac{\partial H}{\partial x} \tau_y - \frac{\partial H}{\partial y} \tau_x = H \left( \frac{\partial \tau_x}{\partial y} - \frac{\partial \tau_y}{\partial x} \right).$$

The system (10) is over-determined and admits non-vanishing solutions only if its coefficients satisfy competability conditions. These conditions amount to a test whether a vector field  $\mathbf{N}$  is indeed the Gauss map of an immersion. If the conditions are satisfied then one can integrate for  $H$  starting from the value at an arbitrary chosen point. Note that the system of equations determines  $H$  up to a constant non-zero multiple, so one needs initial data to nail down an unique solution. This degree of freedom is a manifestation of the invariance of the Gauss map under global scaling.

The competability conditions have an important application in computer vision as a filter.

### 3.2 The Gauss Map Filter

**Theorem 3.3.** *Let  $\mathbf{N}$  be Gauss map of a conformal immersion. Then at every point  $p$  in the domain of the immersion,  $\mathbf{N}$  must satisfy either the equations*

$$\frac{1}{2} (d\mathbf{N}_p - \mathbf{N}(p) \times (d\mathbf{N}_p \circ J)) = \mathbf{0} \quad (11)$$

or the system of equations

$$\left\langle \frac{\partial^2 \mathbf{N}}{\partial x^2} + \frac{\partial^2 \mathbf{N}}{\partial y^2} \middle| \mathbf{N} \right\rangle = \left| \frac{\partial \mathbf{N}}{\partial x} \right|^2 + \left| \frac{\partial \mathbf{N}}{\partial y} \right|^2 \quad (12)$$

$$\left( \frac{\partial^2 \mathbf{N}}{\partial x^2} + \frac{\partial^2 \mathbf{N}}{\partial y^2} \right) \times \mathbf{N} = \left( \frac{\partial \tau_x}{\partial y} - \frac{\partial \tau_y}{\partial x} \right) \quad (13)$$

where  $(x, y)$  is an arbitrary coordinate system defined in a neighborhood of  $p$  and satisfying  $\frac{\partial}{\partial y} = J \left( \frac{\partial}{\partial x} \right)$ , and  $\tau$  is the 1-form defined in (9).

The system (12), (13) expresses the completeness conditions necessary for finding non-vanishing solutions  $H$  of (10).

In practice we use the equations in Theorem 3.3 to filter out noise in the Gauss map sampled by the sensors and the Gauss map extraction algorithms. A given sample is declared valid only if it satisfies at least one of the systems within given a threshold range.

Once the mean curvature is computed we are ready to compute the Gauss curvature the principal axes, and a parameterization (immersion). The usual approach is to recover the immersion and then compute the rest using the expression for the immersion. We adopt a slightly different tack. The motivation for this twist comes from two directions. First we want to avoid computations which may introduce round off errors and other numeric noise – if we stuck with the classical approach we would have to integrate and then differentiate numerically. Second, once we have  $\mathbf{N}$  and  $H$  we can extract other ingredients needed for the computations directly from the sampled data. To clarify these points let us look at the theory we propose to use to compute the Gauss curvature and the principal axes.

### 3.3 The Gauss Curvature and the Principal Axes

For the rest of the paper let  $\mathbf{N}$  be Gauss map of a twice differentiable conformal immersion  $\mathbf{f}$ . The complex structure  $J$  on the domain  $M$  of  $\mathbf{f}$  coincides with the induced structure  $J_{\mathbf{f}}$  and so for the rest of the paper we use only the notation  $J$  for the complex structure. Let  $\omega$  be the  $\mathbf{R}^3$ -valued 1-form introduced in Theorem 3.1. Therefore  $\omega = d\mathbf{N} + Hd\mathbf{f}$ . We show how to express  $\omega$  purely in terms of  $d\mathbf{N}$  and the complex structure and relate it to the principal curvature axes and the Gauss curvature.

**Corollary 3.2.** *(The form  $\omega$  and principal axes) Let  $\mathbf{N}$  be the Gauss map of a parameterized surface  $\mathbf{f} : M \rightarrow \mathbf{R}^3$ , and let  $J$  be the complex structure. If  $\mathbf{f}$  is twice continuously differentiable, then*

$$\omega = \frac{1}{2} (d\mathbf{N} + \mathbf{N} \times d\mathbf{N} \circ J). \quad (14)$$

Furthermore,  $\omega(\mathbf{u})$  is collinear to  $d\mathbf{f}(\mathbf{u})$  if and only if the vector  $\mathbf{u}$  is collinear to a principal curvature vector. Thus the quadratic form  $\langle \omega(\cdot) | d\mathbf{f}(\cdot) \rangle$  is symmetric and trace-free (i.e., has zero trace), and its eigenvalues are precisely  $\pm \frac{1}{2}(\kappa_1 - \kappa_2)$ , where  $\kappa_1$  and  $\kappa_2$  are the principal curvatures.

We can estimate the principal curvature vectors by solving

$$\frac{1}{2} (d\mathbf{N}(\mathbf{u}) + \mathbf{N} \times d\mathbf{N}(J(\mathbf{u}))) = \lambda d\mathbf{f}(\mathbf{u}) \quad (15)$$

for the scalar  $\lambda$  and the vector  $\mathbf{u}$ . This amounts to diagonalizing a symmetric trace free matrix representing the quadratic form  $\langle \omega(\cdot) | d\mathbf{f}(\cdot) \rangle$ . The diagonalization of such matrices is more stable than the diagonalization of general matrices.

**Corollary 3.3.** *(Gauss curvature:) Let  $\mathbf{N}$  be the Gauss map of a parameterized surface  $\mathbf{f} : M \rightarrow \mathbf{R}^3$  and let  $J_f$  be the induced complex structure. Let  $H$  be the mean curvature. Let  $\mathbf{f}$  be twice continuously differentiable,  $\omega$  be the one form defined in (5), and  $\lambda^2$  be the sum of the squares of the eigenvalues of the quadratic form,  $\langle \omega(\cdot) | d\mathbf{f}(\cdot) \rangle$ . Then, the Gauss curvature,  $K$ , satisfies*

$$K = H^2 - \lambda^2. \quad (16)$$

Equation (16) gives a stable method for computing the Gauss curvature  $K$ . We do not need to diagonalize the quadratic form matrix. To compute  $\lambda^2$ , we can choose any orthonormal basis of the tangent plane to the surface in  $\mathbf{R}^3$ , then we represent the quadratic form  $\langle \omega(\cdot) | d\mathbf{f}(\cdot) \rangle$  as matrix  $A$ , and set  $\lambda^2$  as follows

$$A = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}, \quad \lambda^2 = a^2 + b^2. \quad (17)$$

*Remark 3.1.* The upshot of all this is that to compute curvatures and principal axes we need the Gauss map, the complex structure, and estimates for the differential  $d\mathbf{f}$ . Furthermore, the computations are more stable than the ones based on the usual geometric formulae.

Finally, we turn to estimating the differential  $d\mathbf{f}$ , and the immersion itself. The idea is that in practice we can either solve for them or use estimates provided from the data, or use a combination of both methods.

### 3.4 Recovering $d\mathbf{f}$ and the Conformal Immersion

We begin by noticing that it is very easy to obtain  $d\mathbf{f}$  in the closure of the region  $M' = \{p \mid H(p) \neq 0\}$  where the mean curvature is not zero,  $H \neq 0$ , that is, where the form  $\tau$  defined in (9) is nondegenerate. Indeed, from (8) we get

$$d\mathbf{f} = \frac{1}{H}\tau, \quad \text{if } H \neq 0.$$

By continuity we get  $d\mathbf{f}$  in the closed domain  $\overline{M'}$ . This leaves us with the task of determining the differential  $d\mathbf{f}$  on open sets on which the mean curvature vanishes identically, if such exist. To do this precisely we solve a Dirichlet problem for a linear elliptic differential equation. The boundary data is provided from the boundary values of  $d\mathbf{f}$  along the boundary  $\partial M'$  and from assumptions about the surface edge properties if the surface has boundary components where  $H$  vanishes identically, or if the surface has an end within the region where  $H$  is identically zero. The later case can be safely disregarded in computer vision applications because of the natural clipping that takes place as we sample surfaces with sensors. The exact mathematical method will be presented in a forthcoming paper. It is based on the classical Weierstrass representation techniques for minimal surfaces and the new techniques introduced in a recent preprint [9].

In principle once we obtain the differential  $d\mathbf{f}$  then a parameterization is obtained by integrating the differential. In practice of course, we have to use quadratures to evaluate the integrals from the discrete data but that amounts to using the differential itself to determine explicitly the displacements between the discrete points on the surface recovered via the immersion.

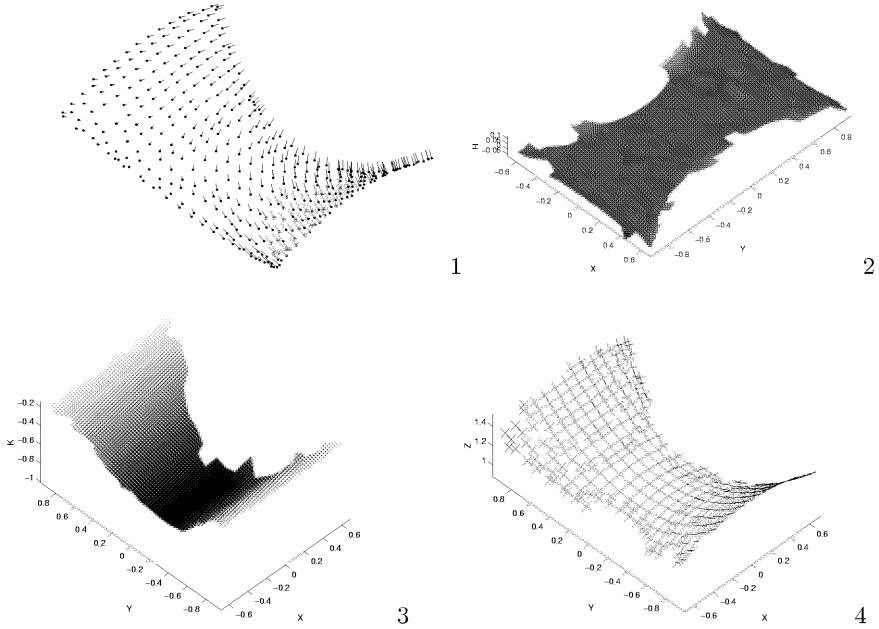
In applications where the initial data set is a cloud of 3D points sampled from the surface of the object we have been estimating  $d\mathbf{f}$  in the region where  $\tau$  is degenerate, equivalently, where  $H \equiv 0$ , directly from the local displacements between sample points weighted by local scale units in available neighborhood directions, [10]. See Fig. 1.

## 4 Examples and Future Work

The proposed method has been tested on various types of data: from MRI images, stereo images, range images, and computer generated surfaces. We show examples of each of these categories here. In all cases, a fish-scales procedure, [14], extracts the Gauss map and neighborhood stratification from the 3D sampled cloud of surface points. **MRI:** The data sets are from the data base of Gill Barequet, Dept of CS, Tel Aviv University and the surface points are extracted manually by Bernhard Geiger, INRIA, Sophia Anapolis. The 3D surface points of human hip joint and cartilage are extracted from MRI images. We show the normals at the sampled points that are input for our method, and the results that the method has produced, Fig. 2.

**Stereo data:** The 3D clouds of points sampled from human faces are obtained using stereo [1,14]. See Fig. 4 and Fig. 5.





**Fig. 1.** Catenoid: (1)Gauss map; (2) and (3) the recovered mean curvature surface,  $(x, y, H)$ , and the Gauss curvature surface,  $(x, y, K)$ ; (4) the principal axes superimposed on the catenoid surface.

**Computer generated:** We have tested the approach on various computer generated surfaces and reported the results elsewhere. Here we just show results for the catenoid, Fig. 1. See [10] for details.

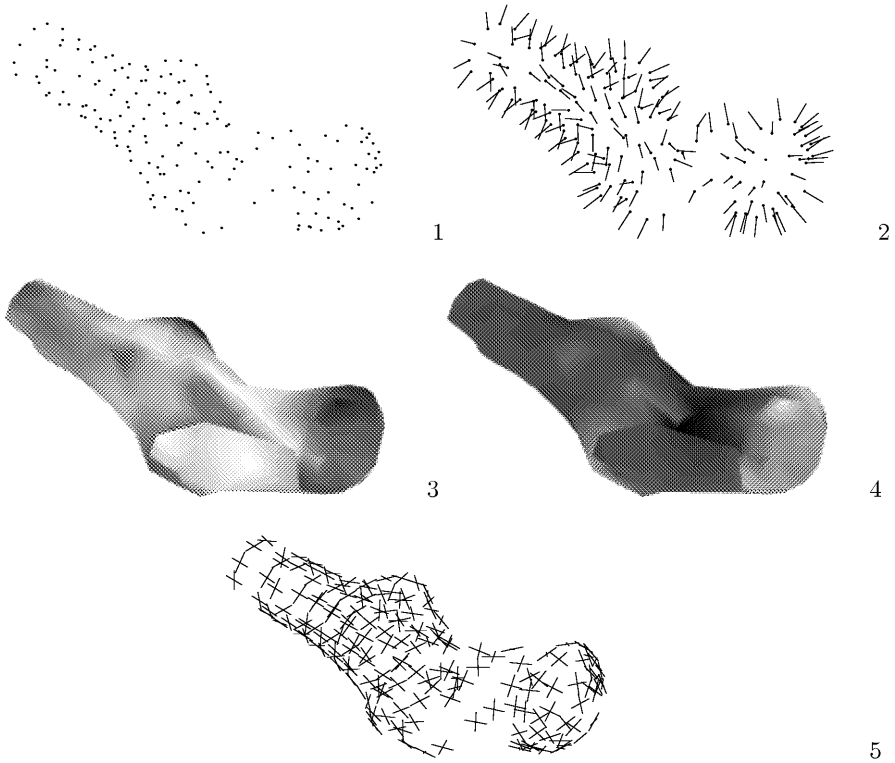
**Range data:** The 3D cloud of points is sampled from the surface of the head of a mannequin. The 3D data are collected by a Cyberware scanner. The range data set is generated at the GRASP Laboratory, University of Pennsylvania. For our results see Fig. 3

Thus we demonstrated, that we can handle various types of input data, from sparse (MRI example) to dense (Range data example), and from clean (Computer generated example) to noisy (Stereo example). Our next goal is to do rigorous performance evaluation of the method, including propagation of errors, empirical evaluation based on ground truth, and empirical comparisons with other existing methods.

## Appendix: Proofs

Recall the definitions of the differential invariants from Section 2. Note that the equations (3) are equivalent to

$$df(e_1) \perp df(e_2) \quad (18)$$



**Fig. 2.** MRI data, human hip joint: (1) the 3D cloud of points extracted from the MRI data; (2) the Gauss map; (3) the restored surface shaded by the recovered mean curvature values; (4) the restored surface shaded by the recovered Gauss curvature; (5) the principal directions at the 3D surface points. The mean curvature values range from  $-3.6622$  to  $-0.0441$  with a mean value of  $-1.4332$ . The Gauss curvature values range from  $-6.4579$  to  $10.8133$  with a mean value of  $0.4488$ . Lighter shades in (4) represent higher, positive Gauss curvatures.

$$d\mathbf{N}(\mathbf{e}_1) = -\kappa_1 d\mathbf{f}(\mathbf{e}_1) \quad (19)$$

$$d\mathbf{N}(\mathbf{e}_2) = -\kappa_2 d\mathbf{f}(\mathbf{e}_2) \quad (20)$$

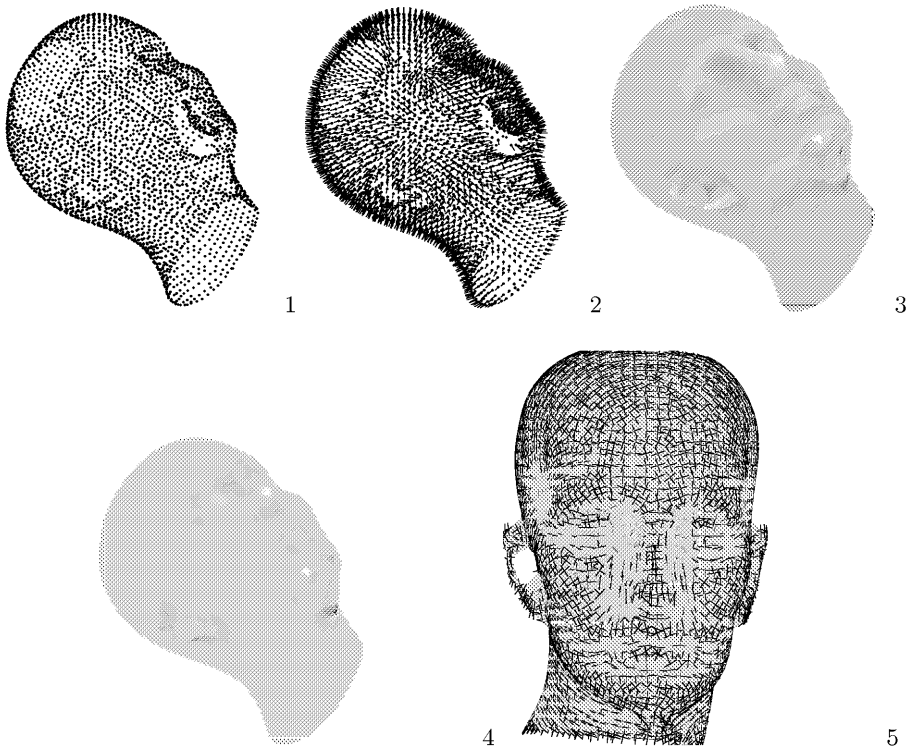
**Proof of Theorem 3.1** The form of equations (19) and (20) suggests that the one form  $d\mathbf{N}$  can be represented as

$$d\mathbf{N} = A d\mathbf{f} + \omega \quad (21)$$

for some coefficient  $A$  and some  $\mathbf{R}^3$ -valued one-form  $\omega$ . We decide to look for a form  $\omega$  satisfying the condition

$$\omega(J_f(\mathbf{u})) = -\mathbf{N} \times \omega(\mathbf{u}). \quad (22)$$

This choice for  $\omega$  can be motivated by the decomposition of symmetric tensors into diagonal and trace-free components. A direct way to motivate our choice is



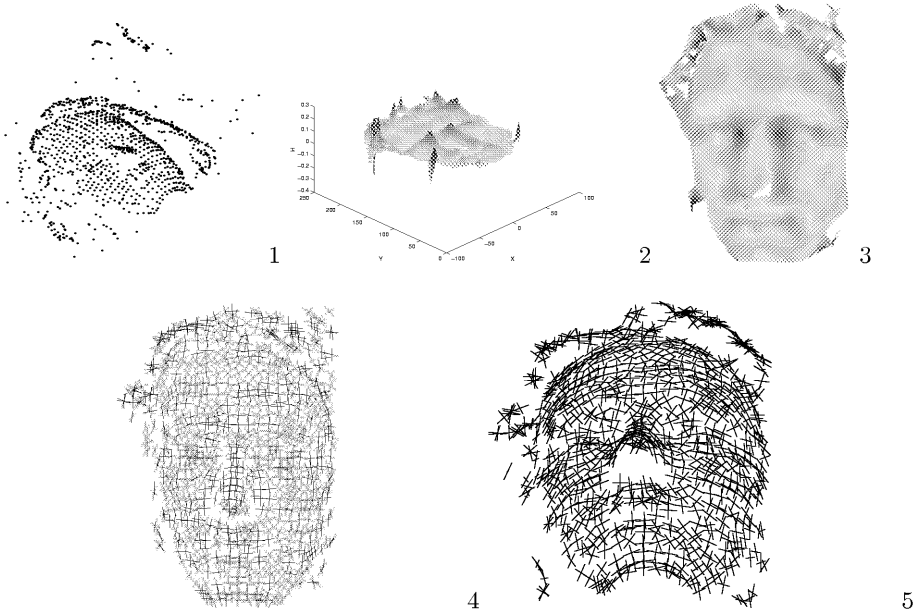
**Fig. 3.** Range data, mannequin head: (1) the 3D cloud of points extracted by Cyberware; (2) the Gauss map; (3) the restored surface shaded by the recovered mean curvature values; (4) the restored surface shaded by the recovered Gauss curvature; (5) the principal directions at at the 3D surface points. Lighter shades in (4) represent higher, positive Gauss curvatures. One can clearly identify the curvature lines, despite the bad Postscript conversion: in the electronic version of this image there were no holes in the curvature lines.

to notice that the form  $d\mathbf{f}$  satisfies

$$d\mathbf{f}(J_f(\mathbf{u})) = \mathbf{N} \times d\mathbf{f}(\mathbf{u}). \quad (23)$$

That is,  $d\mathbf{f}$  relates a counter-clockwise rotation by ninety degrees in  $T_p(M)$  to a counterclockwise rotation by ninety degrees around the axes  $\mathbf{N}$  in  $\mathbf{R}^3$ . On the other hand, the condition (22) guarantees that the form  $\omega$  relates a counter-clockwise rotation by ninety degrees in  $T_p(M)$  to a clockwise rotation by ninety degrees around the axes  $\mathbf{N}$  in  $\mathbf{R}^3$ . The representation (21) accounts for the possibility that the one-form  $d\mathbf{N}$  may be a combination of forms which rotate in different directions around the  $\mathbf{N}$  axes. From (21) and (19) and (20) we obtain

$$\omega(\mathbf{e}_i) = (\kappa_i + A) d\mathbf{f}(\mathbf{e}_i), \quad i = 1, 2.$$



**Fig. 4.** Stereo data, "face". (1) The reconstructed 3D points, from stereo. (2) The surface graph  $(x, y, H)$  of recovered mean curvature. (3) The recovered mean curvature surface  $(x, y, H)$  frontal view. Note that we use as height the curvature values. (4) and (5) The principal axes attached at the 3D points. We can identify well-formed curvature lines in nonplanar regions (on the chin, the eyebrow areas, the cheeks).

These identities show that (6) holds. Furthermore combining the identities with (22) and  $\mathbf{e}_2 = J_f(\mathbf{e}_1)$  we obtain

$$\begin{aligned}\omega(\mathbf{e}_2) &= (\kappa_2 + A) d\mathbf{f}(\mathbf{e}_2) \\ -\mathbf{N} \times \omega(\mathbf{e}_1) &= \mathbf{N} \times (\kappa_2 + A) d\mathbf{f}(\mathbf{e}_1) \\ -\mathbf{N} \times (\kappa_1 + A) d\mathbf{f}(\mathbf{e}_1) &= \mathbf{N} \times (\kappa_2 + A) d\mathbf{f}(\mathbf{e}_1)\end{aligned}$$

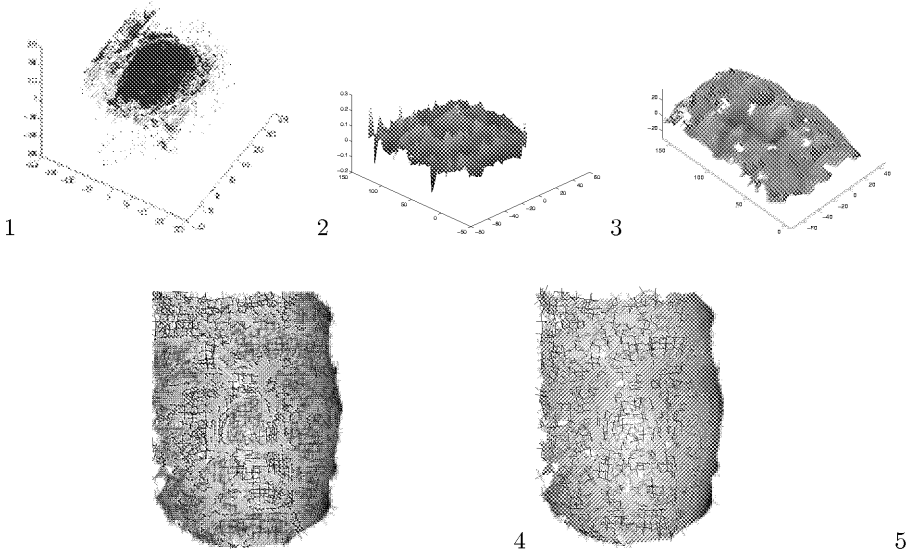
The last identity implies that the tangential vector  $(\kappa_1 + \kappa_2 + 2A) d\mathbf{f}(\mathbf{e}_1)$  is colinear to the normal  $\mathbf{N}$ . This can only happen if it is the zero vector in  $\mathbf{R}^3$ , that is,  $A = -\frac{1}{2}(\kappa_1 + \kappa_2) = -H$ .

#### **Proof of Corollaries 3.1, 3.2, 3.3.**

From (4), (2), and (22) we get

$$\begin{aligned}\mathbf{N} \times d\mathbf{N} \circ J_f &= -H\mathbf{N} \times (\mathbf{N} \times d\mathbf{f}) - \mathbf{N} \times (\mathbf{N} \times \omega) \\ &= H d\mathbf{f} + \omega.\end{aligned}$$

The identities (8) and (14) follow directly from (4) and the identity  $\mathbf{N} \times d\mathbf{N} \circ J_f = H d\mathbf{f} + \omega$ . Rewriting (4) in the form  $\omega = d\mathbf{N} + H d\mathbf{f}$  we conclude that  $\omega(\mathbf{u})$  is colinear to  $d\mathbf{f}(\mathbf{u})$  if and only if the later is colinear to  $d\mathbf{N}(\mathbf{u})$ , that is, if and only



**Fig. 5.** Stereo data, "face 2". (1) The reconstructed 3D points, from stereo. (2) The surface graph  $(x, y, H)$ . (3) The recovered surface shaded by mean curvature. (We used a simple polygonization to render the surface.) (4) and (5) The principal axes attached at the 3D points. We can identify well-formed curvature lines in non-planar regions (on the chin, the eyebrow areas, the cheeks). In (4) we show the principle frames at all sample points. In (5) we show only half the points. Note that the conversion to Postscript lead to image quality loss.

if  $\mathbf{u}$  is parallel to a principal curvature vector. Furthermore, from equations (19) and (20) we obtain

$$\omega(\mathbf{e}_1) = \frac{k_2 - k_1}{2} d\mathbf{f}(\mathbf{e}_1), \quad \omega(\mathbf{e}_2) = \frac{k_1 - k_2}{2} d\mathbf{f}(\mathbf{e}_2).$$

**Proof of Theorem 3.2** Conclusion 1 follows directly from (8), that is, from

$$-Hd\mathbf{f} = \tau. \quad (24)$$

To obtain Conclusion 2 differentiate (24) and multiply both sides by  $H$ .

**Proof of Theorem 3.3** In the regions where  $H = 0$ , that is,  $\tau$  is degenerate we must have  $\tau \equiv 0$  which is the same as (11). The rest of the proof follows from splitting the system  $Hd\tau = dH \wedge \tau$  into pieces tangential and perpendicular to  $\mathbf{N}$  and restating them in terms of the derivatives of the Gauss map.

## References

- [1] Bajcsy, R., Enciso, R., Kamberova, G., Nocera, L., Šara, R.: 3D Reconstruction of Environments for Virtual Collaboration, *Proc. 4th IEEE Workshop on Applications of Computer Vision*, Princeton, NJ, (1998).
- [2] Boissonnat, J.D., Cazals, F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *ACM Proc XVIIth Annl Symp. in Comp. Geom.* Hong Kong (2000).
- [3] Besl, P., Jain, R.: Invariant surface characteristics and 3d object recognition in range images. *CVGIP*, 33:33–80, (1986).
- [4] Quicken, M., Brechbüller, C., et al: Parameterization of Closed Surfaces for Parametric Surface Description. *Proceedings IEEE CVPIR 2000*, volume 1, 354–360.
- [5] DoCarmo, M.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, (1976).
- [6] Devernay, F.: “Computing Differential Properties of 3-D Shapes from Stereoscopic Images without 3-D Models”, INRIA, RR-2304, Sophia Antipolis, (1994).
- [7] Feldmar, J., Ayache, N.: *Registration of Smooth Surfaces Using Differential Properties*. Number 801. Springer-Verlag, (1994).
- [8] Ikeuchi, K., Herbert, M.: Spherical Representations: from EGI to SAI CMU Technical report: CMU-CS-95-197 (1995).
- [9] Kamberov, G., Norman, P., Pinkall, U., Pedit, F.: Quaternions, Spinors, and Surfaces to appear in *Contemporary Mathematics*, AMS
- [10] Kamberov, G., Kamberova, G.: Shape Invariants and Principal Directions from 3D Points and Normals, *Proceedings of 10th Intl. Conf. in Central Europe on Computer Graphics and Visualization*, (2002), Plzen. *Journal of WSCG*, Volume 10, (2002), Pages 537–544.
- [11] Kobayashi, S., Nomidzu, K.: *Foundations of Differential Geometry*. Volume 2. Chapter IX. Interscience Publ. New York. (1969).
- [12] Koenderink, J.: *Solid Shape* MIT press, (1990).
- [13] Metaxas, D., Terzopoulos, D.: Dynamic 3D models with local and global deformations: deformable superquadrics, *IEEE PAMI*, 13(7):703–714, (1991).
- [14] Šara, R., Bajcsy, R.: “Fish-Scales: Representing Fuzzy Manifolds,” *Proc. Int. Conference on Computer Vision*, Bombay, India, Narosa Publishing House, (1998).
- [15] Shum, H., Herbert, M., Ikeuchi, K.: On 3d shape synthesis. In *Proc. Image Under. Workshop*, volume 2, pages 1103–1112, (1996).
- [16] Sommerfeld, A.: *Mechanics of Deformable Bodies*. *Lectures in Theoretical Physics* volume 2, pp 122–124, Academic Press, New York, (1950).
- [17] Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. ICCV*. IEEE Comp. Soc. Press, (1995).
- [18] Worthington, P., Hancock, E.: Histogram-based Object Recognition Using Shape from Shading. *Proc. IEEE Conf. CVPR 2000*, volume 1, 643–648 (2000).
- [19] Yuen, P., et al: Curvature and Torsion Feature Extraction from 3-D Meshes and Multiple Scales. *IEEE Proc. Vis. Image Signal Process*, volume 147, No 5, 454–462 (2000).

# Interpolating Sporadic Data

Lyle Noakes<sup>1</sup> and Ryszard Kozera<sup>2</sup>

<sup>1</sup> Department of Mathematics & Statistics,  
The University of Western Australia, 35 Stirling Highway,  
Crawley 6009 WA, Australia  
[lyle@maths.uwa.edu.au](mailto:lyle@maths.uwa.edu.au)

<http://www.maths.uwa.edu.au/~lyle/>

<sup>2</sup> Department of Computer Science & Software Engineering,  
The University of Western Australia, 35 Stirling Highway,  
Crawley 6009 WA, Australia  
[ryszard@cs.uwa.edu.au](mailto:ryszard@cs.uwa.edu.au)

<http://www.cs.uwa.edu.au/~ryszard/>

**Abstract.** We report here on the problem of estimating a smooth planar curve<sup>a</sup>  $\gamma : [0, T] \rightarrow \mathbb{R}^2$  and its derivatives from an ordered sample of interpolation points  $\{\gamma(t_0), \gamma(t_1), \dots, \gamma(t_{i-1}), \gamma(t_i), \dots, \gamma(t_{m-1}), \gamma(t_m)\}$ , where  $0 = t_0 < t_1 < \dots < t_{i-1} < t_i < \dots < t_{m-1} < t_m = T$ , and the  $t_i$  are *not known precisely* for  $0 < i < m$ . Such situation may appear while searching for the boundaries of planar objects or tracking the mass center of a rigid body with no times available. In this paper we assume that the distribution of  $t_i$  coincides with *more-or-less uniform sampling*. A fast algorithm, yielding *quartic convergence rate* based on 4-point piecewise-quadratic interpolation is analysed and tested. Our algorithm forms a substantial improvement (with respect to the speed of convergence) of piecewise 3-point quadratic Lagrange interpolation [19] and [20]. Some related work can be found in [7]. Our results may be of interest in computer vision and digital image processing [5], [8], [13], [14], [17] or [24], computer graphics [1], [4], [9], [10], [21] or [23], approximation and complexity theory [3], [6], [16], [22], [26] or [27], and digital and computational geometry [2] and [15].

**Keywords:** shape, image analysis and features, curve interpolation

## 1 Introduction

Let  $\gamma : [0, T] \rightarrow \mathbb{R}^n$  be a smooth regular curve, namely  $\gamma$  is  $C^k$  for some  $k \geq 1$  and  $\dot{\gamma}(t) \neq \mathbf{0}$  for all  $t \in [0, T]$  (with  $0 < T < \infty$ ). Consider the problem of estimating  $\gamma$  from an ordered  $m + 1$ -tuple

$$\mathcal{Q} = (q_0, q_1, \dots, q_m)$$

---

<sup>a</sup> This research was conducted under an Australian Research Council Small Grant<sup>1,2</sup>. Additional support was received under an Alexander von Humboldt Research Fellowship.<sup>2</sup>

of points in  $\mathbb{R}^n$ , where  $q_i = \gamma(t_i)$ , and  $0 = t_0 < t_1 < \dots < t_{i-1} < t_i < \dots < t_{m-1} < t_m = T$ . If the  $t_i$  are given then  $\gamma$  can be approximated in a variety of ways.

*Example 1.* Let  $\gamma$  be  $C^{r+2}$  where  $r > 0$ , and take  $m$  to be a multiple of  $r$ . Then  $\mathcal{Q}$  gives  $\frac{m}{r}$  subsets of  $r+1$ -tuples of the form

$$(q_0, q_1, \dots, q_r), \quad (q_r, q_{r+1}, \dots, q_{2r}), \quad \dots, \quad (q_{m-r}, q_{m-r+1}, \dots, q_m).$$

The  $j$ th  $r+1$ -tuple can be interpolated by a polynomial  $\hat{\gamma}_j : [t_{(j-1)r}, t_{jr}] \rightarrow \mathbb{R}^n$  of degree  $r$ , and the track-sum  $\hat{\gamma}$  of the  $\hat{\gamma}_j$  is everywhere-continuous, and  $C^\infty$  except at  $t_r, t_{2r}, \dots, t_{m-r}$ . Suppose that sampling is uniform i.e.  $t_i = \frac{iT}{m}$  for  $0 \leq i \leq m$ . Then  $\hat{\gamma}(t) = \gamma(t) + O(\frac{1}{m^{r+1}})$  for  $t \in [0, T]$ , and  $\dot{\hat{\gamma}}(t) = \dot{\gamma}(t) + O(\frac{1}{m^r})$  for  $t \neq t_r, t_{2r}, \dots, t_{m-r}$ . The error in length can be shown to be  $O(\frac{1}{m^{r+1}})$  or  $O(\frac{1}{m^{r+2}})$ , accordingly as  $r$  is odd or even (see Theorem 1 in [19]).

In practice the  $t_i$  might not be given for  $0 < i < m$ .

*Example 2.* Let  $\gamma$  be  $C^4$  curve in  $\mathbb{R}^n$ . For  $0 \leq \varepsilon \leq 1$  the  $t_i$  are said to be  $\varepsilon$ -uniformly sampled when there is an order-preserving  $C^k$  reparameterization  $\phi : [0, T] \rightarrow [0, T]$  such that

$$t_i = \phi\left(\frac{iT}{m}\right) + O\left(\frac{1}{m^{1+\varepsilon}}\right).$$

Although the set  $\mathcal{Q}$  does not arise from perfectly uniform sampling, we can pretend that they do, and apply the method of Example 1. This is done in [19] and [20] with a view to estimating the length  $d(\gamma)$  of  $\gamma$ . So far as  $\gamma$  and its derivatives are concerned the proof of Theorem 2 in [19] gives estimates of  $\gamma$  and  $\dot{\gamma}$  with uniform  $O(\frac{1}{m^{1+2\varepsilon}})$  and  $O(\frac{1}{m^{2\varepsilon}})$  errors, respectively. The latter implies that  $d(\hat{\gamma}) - d(\gamma) = O(\frac{1}{m^{4\varepsilon}})$  (see Theorem 2 in [19]). So when the distribution of the  $t_i$  is most nearly uniform ( $\varepsilon = 1$ ) piecewise-quadratic Lagrange interpolation gives good estimates for  $\gamma$ ,  $\dot{\gamma}$ , and  $d(\gamma)$ , namely cubic, quadratic and quartic, respectively. At the other extreme, where  $\varepsilon = 0$ , the methods of Example 1 have very little value. The extension of  $\varepsilon$ -uniform sampling for  $\varepsilon > 1$  could also be considered. This case represents, however, a very small perturbation of uniform sampling (up to  $\phi$ -order-preserving shift) which seems to be of less interest in applications. Nevertheless, by repeating the argument used in Theorem 2 (see [19]) it can be shown, that the case  $\varepsilon > 1$  renders for  $\gamma$ ,  $\dot{\gamma}$ , and  $d(\gamma)$  estimation with piecewise-quadratic Lagrange interpolation the same results as for  $\varepsilon = 1$ .

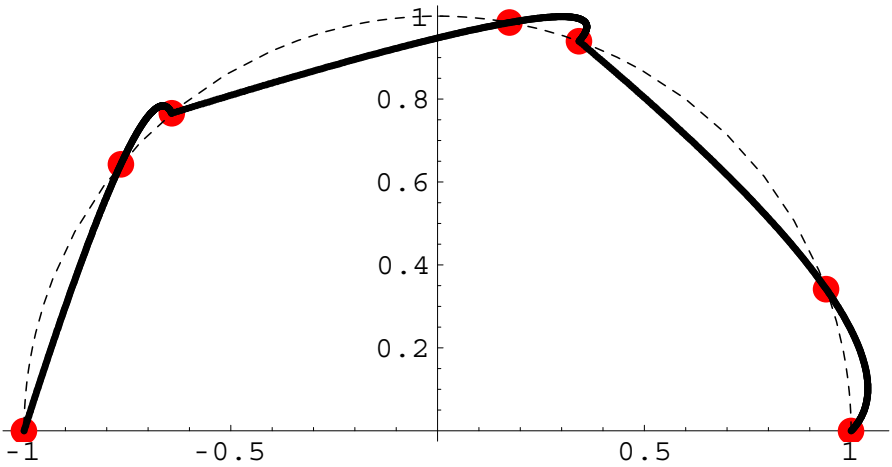
A typical instance is shown in Figure 1, where ordinary Lagrange interpolation by piecewise-quadratics does not work well.

In general a less restrictive hypothesis than  $\varepsilon$ -uniformity is that the  $t_i$  should be sampled *more-or-less uniformly* in the following sense.

**Definition 1.** Sampling is *more-or-less uniform* when there are constants  $0 < K_l < K_u$  such that, for any sufficiently large integer  $m$ , and any  $1 \leq i \leq m$ ,

$$\frac{K_l}{m} \leq t_i - t_{i-1} \leq \frac{K_u}{m}.$$





**Fig. 1.** 7 data points ( $m = 6$ ), with 3 successive triples interpolated by piecewise-quadratics, giving length estimate  $\pi + 0.0601035$  for the semicircle (shown dashed).

With more-or-less uniform sampling, increments between successive parameters are neither large nor small in proportion to  $\frac{T}{m}$ .

*Example 3.* For  $0 < i < m$  set  $t_i = \frac{(3i+(-1)^i)T}{3m}$ . Then sampling is more-or-less uniform, with  $K_l = \frac{T}{3}, K_u = \frac{5T}{3}$ . Let  $\gamma : [0, \pi] \rightarrow \mathbb{R}^2$  be the parameterization  $\gamma(t) = (\cos t, \sin t)$  of the unit semicircle in the upper half-plane. When  $m$  is small the image of  $\hat{\gamma}$  does not much resemble a semicircle, as in Figure 1, where  $m = 6$ .

*Example 4.* For  $0 < i < m$  let  $t_i$  be random (according to some distribution) in the interval  $[\frac{(3i-1)T}{3m}, \frac{(3i+1)T}{3m}]$ . Then sampling is more-or-less uniform, with  $K_u, K_l$  as in Example 3.

*Example 5.* Choose  $\theta > 0$  and  $0 < L_l < L_u$ . Set  $s_0 = 0$ . For  $1 \leq i \leq m$  choose  $\delta_i \in [\frac{L_l}{m}, \frac{L_u}{m}]$  independently from (say) the uniform distribution. Define  $s_i = s_{i-1} + \delta_i$  for  $i = 1, 2, \dots, m$ . The expectation of  $s_m$  is  $\frac{L_u + L_l}{2}$  and the standard deviation  $\frac{L_u - L_l}{2\sqrt{3m}}$ . So if  $m$  is large  $s_m \approx \frac{L_u + L_l}{2}$  with high probability. For  $0 \leq i \leq m$ , define  $t_i = \frac{s_i T}{s_m}$ . Set

$$K_l = \frac{2L_l T}{L_u + L_l} - \theta, \quad K_u = \frac{2L_u T}{L_u + L_l} + \theta.$$

Then with high probability for  $m$  large, the sampling  $(t_0, t_1, t_2, \dots, t_m)$  from  $[0, T]$  is more-or-less uniform with constants  $K_l, K_u$ .

More-or-less uniform sampling is invariant with respect to reparameterizations, namely if  $\phi : [0, T] \rightarrow [0, T]$  is an order-preserving  $C^1$  diffeomorphism, and if  $(t_0, t_1, \dots, t_m)$  are sampled more-or-less uniformly, then so are  $(\phi(t_0), \phi(t_1), \dots, \phi(t_m))$ . So reparameterizations lead to further examples from the ones already given. From now on take  $n = 2$  and suppose that  $\gamma$  is  $C^4$  and (without loss) parameterized by arc-length, namely  $\|\dot{\gamma}\|$  is identically 1. The *curvature* of  $\gamma$  is defined as

$$k(t) = \det(M(t)) , \quad (1)$$

where  $M(t)$  is the  $2 \times 2$  matrix with columns  $\dot{\gamma}(t), \ddot{\gamma}(t)$ . Suppose that  $k(t) \neq 0$  for all  $t \in [0, T]$ , namely  $\gamma$  is *strictly convex*. Let the  $t_i$  be sampled more-or-less uniformly. Then in section 2 we show how to carry out *piecewise 4-point quadratic interpolations* based on  $\mathcal{Q}$ . This approximation scheme is rather specialised, and much more elaborate than Lagrange interpolation. On the other hand it works well in cases such as in Figure 1. More precisely, from the proof of [18] Theorem 1, we obtain

**Theorem 1.** *Let  $\gamma$  be strictly convex and suppose that sampling is more-or-less uniform. Then we can estimate  $\gamma$  and  $\dot{\gamma}$  from  $\mathcal{Q}$  with  $O(\frac{1}{m^4})$  and  $O(\frac{1}{m^3})$  errors respectively.*

As a consequence of the last theorem we obtain  $d(\hat{\gamma}) - d(\gamma) = O(\frac{1}{m^4})$  (see [18]). Applying piecewise 4-point quadratic interpolation to the data of Figure 1, gives a much more satisfactory estimate of the semicircle than Lagrange interpolation. This can be seen in Figure 2.



**Fig. 2.** Piecewise 4-point quadratic using 7 data points ( $m = 6$ ) from a semicircle (shown dashed). Length estimate:  $\pi - 0.00723637$ .

The improvement is the result of a serious effort to estimate the parameters  $t_i$  from  $\mathcal{Q}$ . Although in practice it is difficult to discern a problem, our piecewise 4-point quadratic estimates  $\tilde{\gamma}$  are usually not  $C^1$ . In theory, at least, this is a

serious defect since  $\gamma$  is  $C^4$ . In section 3 we show how to refine the construction in section 2, replacing  $\tilde{\gamma}$  by a  $C^1$  curve with the same properties of approximation to  $\gamma$ .

## 2 Piecewise 4-Point Quadratics

Let  $\mathcal{Q}$  be sampled more-or-less uniformly from  $\gamma$ , and suppose (without loss) that  $m$  is a positive integer multiple of 3. For each quadruple  $(q_i, q_{i+1}, q_{i+2}, q_{i+3})$ , where  $0 \leq i \leq m-3$ , define  $a_0, a_1, a_2 \in \mathbb{R}^2$  and  $Q^i(s) = a_0 + a_1 s + a_2 s^2$ , by

$$Q^i(0) = q_i, \quad Q^i(1) = q_{i+1}, \quad Q^i(\alpha) = q_{i+2} \quad \text{and} \quad Q^i(\beta) = q_{i+3}.$$

Then  $a_0 = q_i$ ,  $a_2 = q_{i+1} - a_0 - a_1$ , and we obtain two vector equations

$$\alpha a_1 + \alpha^2(p_1 - a_1) = p_\alpha, \quad \beta a_1 + \beta^2(p_1 - a_1) = p_\beta, \quad (2)$$

where  $(p_1, p_\alpha, p_\beta) \equiv (q_{i+1} - q_i, q_{i+2} - q_i, q_{i+3} - q_i)$ . Then (2) amounts to four quadratic scalar equations in four scalar unknowns  $a_1 = (a_{11}, a_{12})$ ,  $\alpha, \beta$ . Set

$$c = -\det(p_\alpha, p_\beta), \quad d = -\det(p_\beta, p_1)/c, \quad e = -\det(p_\alpha, p_1)/c, \quad (3)$$

where  $c, d, e \neq 0$  by strict convexity, and define

$$\rho_1 = \sqrt{e(1+d-e)/d}, \quad \rho_2 = \sqrt{d(1+d-e)/e}. \quad (4)$$

Then (2) has two solutions (see Appendix 1)

$$(\alpha_+, \beta_+) = \frac{(1 + \rho_1, 1 + \rho_2)}{e - d}, \quad (\alpha_-, \beta_-) = \frac{(1 - \rho_1, 1 - \rho_2)}{e - d} \quad (5)$$

provided  $\rho_1, \rho_2$  are real and  $e - d \neq 0$ . In Appendix 1 it is also shown that these conditions hold, and in Appendix 2 it is proved that precisely one of (5) satisfies the additional constraint

$$1 < \alpha < \beta. \quad (6)$$

From now on, suppose<sup>b</sup> that  $k(t) < 0$ , for all  $t \in [0, T]$ . Define now

$$l(t) = \frac{\det(\frac{d\gamma}{dt}, \frac{d^3\gamma}{dt^3})}{k(t)}$$

and let  $l = l(t_i)$ . Then it is proved in [18] that

$$(\alpha_+, \beta_+) = \frac{((t_{i+2} - t_i)(1 + \frac{l(t_{i+2} - t_{i+1})}{6}), (t_{i+3} - t_i)(1 + \frac{l(t_{i+3} - t_{i+1})}{6}))}{t_{i+1} - t_i} + O(\frac{1}{m^2}). \quad (7)$$

<sup>b</sup> The other case, where  $k(t)$  is everywhere positive, is dealt with by considering the reversed curve  $\gamma_r(t) = (\gamma_1(T-t), \gamma_2(T-t))$ .

Note that here a third-order Taylor's expansion of  $\gamma$  is needed to justify the asymptotic behaviour of approximation results claimed in Theorem 1. Hence the assumption of  $\gamma \in C^4$  is imposed. On the other hand, the justification of Appendices 1 and 2 requires only a second-order Taylor's expansion of  $\gamma$  and therefore a weaker restriction on smoothness of curve  $\gamma$ , namely  $\gamma \in C^3$  is required. Set now  $(\alpha, \beta) = (\alpha_+, \beta_+)$ . Then, for  $0 \leq s \leq \beta$ ,  $Q^i(s) = q_i + a_1 s + a_2 s^2$ , where

$$a_1 = \frac{p_\alpha - \alpha^2 p_1}{\alpha - \alpha^2} = \frac{p_\beta - \beta^2 p_1}{\beta - \beta^2}, \quad a_2 = \frac{\alpha p_1 - p_\alpha}{\alpha - \alpha^2} = \frac{\beta p_1 - p_\beta}{\beta - \beta^2}. \quad (8)$$

The quadratics  $Q^i$ , determined by  $\mathcal{Q}$  and  $i$ , need to be reparameterized for comparison with the original curve  $\gamma$ . In doing so, let  $\psi : [t_i, t_{i+3}] \rightarrow [0, \beta]$  be the cubic given by

$$\psi(t_i) = 0, \quad \psi(t_{i+1}) = 1, \quad \psi(t_{i+2}) = \alpha, \quad \psi(t_{i+3}) = \beta.$$

For  $m$  large  $\psi$  is an order-preserving diffeomorphism, and we define  $\tilde{\gamma}_i = Q^i \circ \psi : [t_i, t_{i+3}] \rightarrow \mathbb{R}^2$ . Then  $\tilde{\gamma}_i$  is polynomial of degree at most 6. It turns out (as part of a difficult proof, given in [18]) that

$$\tilde{\gamma}_i(t) = \gamma(t) + O\left(\frac{1}{m^4}\right) \quad \text{and} \quad \dot{\tilde{\gamma}}_i(t) = \dot{\gamma}(t) + O\left(\frac{1}{m^3}\right), \quad \text{for } t \in [t_i, t_{i+3}].$$

Then the track-sum  $\tilde{\gamma}$  of the arcs swept out by the  $Q^{3j}$  gives a  $O(\frac{1}{m^4})$  uniformly accurate approximation of the image of  $\gamma$ . Although  $\tilde{\gamma}$  is not  $C^1$  at  $t_3, t_6, \dots, t_{m-3}$ , the differences in left and right derivatives are  $O(\frac{1}{m^3})$ , and hardly discernible when  $m$  is large. In section 3 we show how to correct this minor defect.

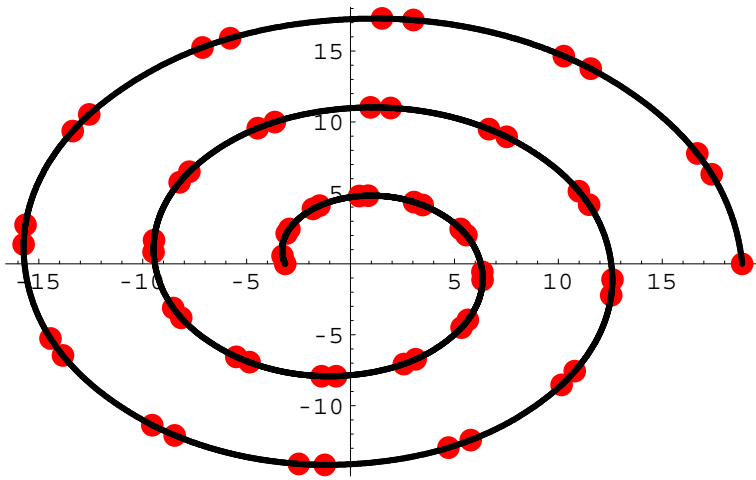
The experiments verifying the rate of length estimation are discussed in [18].

### 3 $C^1$ Approximations

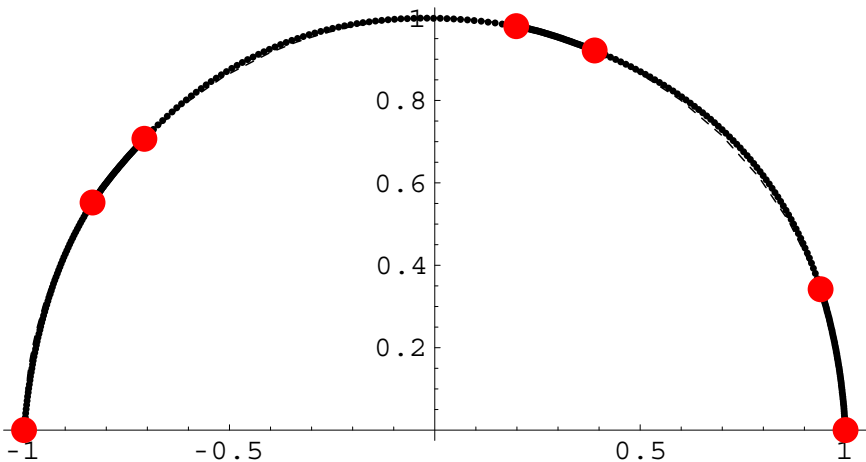
Instead of using 4-point quadratics as estimates of segments of  $\gamma$ , we can use them to estimate slopes of  $\gamma$  at  $t_0, t_1, \dots, t_m$ . Except for  $i = 0, m$  there is more than one choice of 4-point quadratic whose domain contains  $t_i$ . The choice does not appear to be critical, but for  $0 < i < m$  we used estimates calculated from quadratics whose domain contained  $t_i$  in the interior.

It is then straightforward to produce a  $C^1$  piecewise quadratic  $\hat{\gamma}$  interpolating the given data points with the estimated slopes (for instance using the deCasteljau construction [3]). In practice  $\hat{\gamma}$  seems slightly preferable to the already excellent estimate  $\tilde{\gamma}$ .

*Example 6.* Compared with the large discontinuities in derivatives at data points 3, 5 (from the right) in Figure 1, the tiny corner at the middle data point 4 in Figure 2 is only just discernible. The modification to a  $C^1$  piecewise quadratic removes this blemish. Although there are only 7 sample points, and the  $t_i$  are unknown for  $0 < i < 6$ , the estimate  $\hat{\gamma}$  shown in Figure 4 is difficult to distinguish from the underlying semicircle.



**Fig. 3.** A piecewise 4-point quadratic approximation to a spiral (singular point excluded), using the more-or-less uniform sampling of Example 3 and 61 data points ( $m = 60$ ). True length: 173.608, estimate: 173.539, piecewise 3-point quadratic estimate: 181.311.



**Fig. 4.**  $C^1$  piecewise-quadratic using 7 data points ( $m = 6$ ) from a semicircle (shown dashed).

## 4 Concluding Remarks

Lagrange interpolation is reasonably effective for sporadic data when the  $t_i$  are distributed in an  $\varepsilon$ -uniform fashion and  $\varepsilon \in (0, 1]$ . Better results are achieved with larger values of  $\varepsilon$  (see [19] and [20]). In general, the less restrictive condition, that the  $t_i$  be distributed more-or-less uniformly, is less straightforward, but (for strictly convex planar curves) the piecewise 4-point quadratic estimate of section 2 works well. The estimate  $\tilde{\gamma}$  of section 2 is piecewise-polynomial, but not  $C^1$ . In section 3 we showed how to replace  $\tilde{\gamma}$  by a piecewise-quadratic  $C^1$  curve  $\hat{\gamma}$ .

There is also some analogous work for estimating lengths of digitized curves; indeed the analysis of digitized curves in  $\mathbb{R}^2$  is one of the most intensively studied subjects in image data analysis. A digitized curve is the result of a process (such as contour tracing, 2D skeleton extraction, or 2D thinning) which maps a curve-like object (such as the boundary of a region) onto a computer-representable curve. As before,  $\gamma : [0, T] \rightarrow \mathbb{R}^2$  is a strictly convex curve parameterized by arc-length. An analytical description of  $\gamma$  is not given, and numerical measurements of points on  $\gamma$  are corrupted by a process of *digitization*:  $\gamma$  is digitized within an orthogonal grid of points  $(\frac{i}{m}, \frac{j}{m})$ , where  $i, j$  are permitted to range over integer values, and  $m$  is a fixed positive integer called *the grid resolution*. Depending on the digitization model [11],  $\gamma$  is mapped onto a digital curve and approximated by a polygon whose length is an estimator for that of  $\gamma$  (see [2], [5], [8], [12], [13], [14] or [24]). We expect to revisit these issues in future.

Related work on interpolation, length estimation, noisy signal reconstruction and complexity involved can be found in [1], [4], [6], [7], [9], [10], [16], [17], [21], [22], [23], [25], [26] or [27].

## References

1. Barsky, B.A., DeRose, T.D.: Geometric Continuity of Parametric Curves: Three Equivalent Characterizations. IEEE. Comp. Graph. Appl. **9**:6 (1989) 60–68
2. Bertrand, G., Imiya, A., Klette, R. (eds): Digital and Image Geometry. Lecture Notes in Computer Science Vol. 2243, Springer-Verlag, Berlin Heidelberg New York (2001)
3. Bézier, P.E.: Numerical Control: Mathematics and Applications. John Wiley, New York (1972)
4. Boehm, W., Farin, G., Kahmann, J.: A Survey of Curve and Surface Methods in CAD. Comput. Aid. Geom. Des. **1** (1988) 1–60
5. Bülow, T., Klette, R.: Rubber Band Algorithm for Estimating the Length of Digitized Space-Curves. In: Sneli, A., Villanva, V.V., Vanrell, M., Alquézar, R., Crowley, J., Shirai, Y. (eds): Proceedings of 15th International Conference on Pattern Recognition. Barcelona, Spain. IEEE, Vol. III. (2000) 551–555
6. Davis, P.J.: Interpolation and Approximation. Dover Pub. Inc., New York (1975)
7. Dąbrowska, D., Kowalski, M.A.: Approximating Band- and Energy-Limited Signals in the Presence of Noise. J. Complexity **14** (1998) 557–570
8. Dorst, L., Smeulders, A.W.M.: Discrete Straight Line Segments: Parameters, Primitives and Properties. In: Melter, R., Bhattacharya, P., Rosenfeld, A. (eds): Ser. Contemp. Maths, Vol. 119. Amer. Math. Soc. (1991) 45–62

9. Epstein, M.P.: On the Influence of Parametrization in Parametric Interpolation. *SIAM. J. Numer. Anal.* **13**:2 (1976) 261–268
10. Hoschek, J.: Intrinsic Parametrization for Approximation. *Comput. Aid. Geom. Des.* **5** (1988) 27–31
11. Klette, R.: Approximation and Representation of 3D Objects. In: Klette, R., Rosenfeld, A., Sloboda, F. (eds): *Advances in Digital and Computational Geometry*. Springer, Singapore (1998) 161–194
12. Klette, R., Bülow, T.: Critical Edges in Simple Cube-Curves. In: Borgefors, G., Nyström, I., Sanniti di Baja, G. (eds): *Proceedings of 9th Conference on Discrete Geometry for Computer Imagery*. Uppsala, Sweden. *Lecture Notes in Computer Science*, Vol. 1953. Springer-Verlag, Berlin Heidelberg (2000) 467–478
13. Klette, R., Kovalevsky, V., Yip, B.: On the Length Estimation of Digital Curves. In: Latecki, L.J., Melter, R.A., Mount, D.A., Wu, A.Y. (eds): *Proceedings of SPIE Conference, Vision Geometry VIII*, Vol. 3811. Denver, USA. The International Society for Optical Engineering (1999) 52–63
14. Klette, R., Yip, B.: The Length of Digital Curves. *Machine Graphics and Vision* **9** (2000) 673–703
15. Klette, R., Rosenfeld, A., Sloboda, F. (eds): *Advances in Digital and Computational Geometry*. Springer, Singapore (1998) 161–194
16. Kvasov, B.I.: *Method of Shape-Preserving Spline Approximation*. World Scientific Pub. Co., Singapore, New Jersey, London, Hong Kong (2000)
17. Moran, P.A.P.: Measuring the Length of a Curve. *Biometrika* **53**:3/4 (1966) 359–364
18. Noakes, L., Kozera, R.: More-or-Less Uniform Sampling and Lengths of Curves. *Quart. Appl. Maths.* In press
19. Noakes, L., Kozera, R., and Klette R.: Length Estimation for Curves with Different Samplings. In: Bertrand, G., Imiya, A., Klette, R. (eds): *Digital and Image Geometry. Lecture Notes in Computer Science* Vol. 2243, Springer-Verlag, Berlin Heidelberg New York, (2001) 339–351
20. Noakes, L., Kozera, R., and Klette R.: Length Estimation for Curves with  $\varepsilon$ -Uniform Sampling. In: Skarbek, W. (ed.): *Proceedings of 9th International Conference on Computer Analysis of Images and Patterns*. Warsaw, Poland. *Lecture Notes in Computer Science*, Vol. 2124. Springer-Verlag, Berlin Heidelberg New York, (2001) 518–526
21. Piegl, L., Tiller, W.: *The NURBS Book*. 2nd edn Springer-Verlag, Berlin Heidelberg (1997)
22. Plaskota, L.: *Noisy Information and Computational Complexity*. Cambridge Uni. Press, Cambridge (1996)
23. Sederberg, T.W., Zhao, J., Zundel, A.K.: Approximate Parametrization of Algebraic Curves. In: Strasser, W., Seidel, H.P. (eds): *Theory and Practice in Geometric Modelling*. Springer-Verlag, Berlin (1989) 33–54
24. Sloboda, F., Zařko, B., Stör, J.: On approximation of Planar One-Dimensional Continua. In: Klette, R., Rosenfeld, A., Sloboda, F. (eds): *Advances in Digital and Computational Geometry*. Springer, Singapore (1998) 113–160
25. Steinhaus, H.: Praxis der Rektifikation und zur Längenbegriff. (in German) *Akad. Wiss. Leipzig Ber.* **82** (1930) 120–130
26. Traub, J.F., Werschulz, A.G.: *Complexity and Information*. Cambridge Uni. Press, Cambridge (1998)
27. Werschulz, A.G., Woźniakowski, H.: What is the Complexity of Surface Integration? *J. Complexity.* **17** (2001) 442–466

## 5 Appendix 1

In this Appendix we solve (2). Note that  $\alpha$  (and  $\beta$ ) cannot vanish as otherwise, by (2), the vector  $p_\alpha = q_2 - q_0 = \mathbf{0}$  ( $p_\beta = q_3 - q_0 = \mathbf{0}$ ) - a contradiction as interpolation points  $\mathcal{Q}$  are assumed to be different. Similarly, as  $q_2 \neq q_1$  and  $q_3 \neq q_1$  we have  $\alpha \neq 1$  and  $\beta \neq 1$ . Thus elimination of  $u_1$  from (2) and further simplification yields

$$\alpha\beta(\alpha - \beta)p_1 = (\beta - \beta^2)p_\alpha - (\alpha - \alpha^2)p_\beta. \quad (9)$$

Consider now two vectors  $p_\beta^\perp = (-p_{\beta 2}, p_{\beta 1})$  and  $p_\alpha^\perp = (-p_{\alpha 2}, p_{\alpha 1})$ , which are perpendicular to  $p_\beta$  and  $p_\alpha$ , respectively. Taking the dot product of (9) first with  $p_\beta^\perp$  and then with  $p_\alpha^\perp$  results in

$$\begin{aligned} \alpha\beta(\alpha - \beta) \langle p_1 | p_\beta^\perp \rangle &= (\beta - \beta^2) \langle p_\alpha | p_\beta^\perp \rangle, \\ \alpha\beta(\alpha - \beta) \langle p_1 | p_\alpha^\perp \rangle &= -(\alpha - \alpha^2) \langle p_\beta | p_\alpha^\perp \rangle. \end{aligned}$$

Since  $\alpha$  and  $\beta$  cannot vanish and  $\langle p_\alpha | p_\beta^\perp \rangle \neq 0$  and  $\langle p_\beta | p_\alpha^\perp \rangle \neq 0$  hold asymptotically (as  $\gamma$  is strictly convex) we obtain

$$\frac{\alpha(\alpha - \beta) \langle p_1 | p_\beta^\perp \rangle}{\langle p_\alpha | p_\beta^\perp \rangle} = (1 - \beta), \quad \frac{\beta(\alpha - \beta) \langle p_1 | p_\alpha^\perp \rangle}{\langle p_\beta | p_\alpha^\perp \rangle} = \alpha - 1. \quad (10)$$

Note that by (3) and convexity of  $\gamma$ ,  $c \neq 0$  asymptotically. A simple verification shows:

$$c = - \langle p_\beta | p_\alpha^\perp \rangle = \langle p_\alpha | p_\beta^\perp \rangle. \quad (11)$$

Similarly

$$d = \frac{- \langle p_1 | p_\beta^\perp \rangle}{c}, \quad e = \frac{- \langle p_1 | p_\alpha^\perp \rangle}{c}.$$

The latter coupled with (11) yields

$$d = \frac{- \langle p_1 | p_\beta^\perp \rangle}{\langle p_\alpha | p_\beta^\perp \rangle}, \quad e = \frac{\langle p_1 | p_\alpha^\perp \rangle}{\langle p_\beta | p_\alpha^\perp \rangle}$$

which combined with (10) renders

$$\alpha(\alpha - \beta)d = \beta - 1, \quad \beta(\alpha - \beta)e = \alpha - 1. \quad (12)$$

The first equation of (12) yields

$$\alpha^2 d + 1 = \beta(1 + d\alpha). \quad (13)$$

Note that  $(1 + d\alpha) \neq 0$  as otherwise since  $\alpha \neq 0$  we would have  $d = -\alpha^{-1}$  and by (13)  $\alpha^2 d + 1$  would vanish which combined with  $d = -\alpha^{-1}$  would lead to  $\alpha = 1$ , a contradiction. Thus by (13)

$$\beta = \frac{\alpha^2 d + 1}{1 + d\alpha}. \quad (14)$$



Substituting (14) into the second equation of (12) yields

$$\frac{\alpha^2 d + 1}{1 + d\alpha}(1 - \alpha)e = \alpha - 1$$

and taking into account that  $\alpha \neq 1$  results in

$$(d^2 - de)\alpha^2 + 2d\alpha + 1 - e = 0. \quad (15)$$

Assuming temporarily

$$\Delta = 4de(1 + d - e) > 0 \quad (16)$$

we arrive at (5). Having found  $(\alpha_{\pm}, \beta_{\pm})$  the corresponding formulae (8) follow immediately. To show (16) recall that

$$d = \frac{\det(p_{\beta}, p_1)}{\det(p_{\alpha}, p_{\beta})} \quad \text{and} \quad e = \frac{\det(p_{\alpha}, p_1)}{\det(p_{\alpha}, p_{\beta})}. \quad (17)$$

As  $\det(v, w) = \|v\|\|w\|\sin(\sigma)$  (where  $\sigma$  is the oriented angle between  $v$  and  $w$ ) for convex  $\gamma$  both  $e < 0$  and  $d < 0$  hold. Thus to justify (16) it is enough to show  $1 + d - e > 0$ . In fact, as  $\gamma$  is strictly convex all of above inequalities are separated from zero. The second-order Taylor's expansion of  $\gamma$  at  $t = t_i$  yields

$$\gamma(t) = \gamma(t_i) + \dot{\gamma}(t_i)(t - t_i) + (1/2)\ddot{\gamma}(t_i)(t - t_i)^2 + O\left(\frac{1}{m^3}\right)$$

as  $0 < T < \infty$  and  $\gamma \in C^4$  (in fact we need here only  $C^3$ ). Thus taking into account that  $\gamma(t_i) = q_i$ ,  $\gamma(t_{i+1}) = q_{i+1}$ ,  $\gamma(t_{i+2}) = q_{i+2}$ , and  $\gamma(t_{i+3}) = q_{i+3}$  we have

$$\begin{aligned} p_1 &= \dot{\gamma}(t_i)(t_{i+1} - t_i) + (1/2)\ddot{\gamma}(t_i)(t_{i+1} - t_i)^2 + O\left(\frac{1}{m^3}\right), \\ p_{\alpha} &= \dot{\gamma}(t_i)(t_{i+2} - t_i) + (1/2)\ddot{\gamma}(t_i)(t_{i+2} - t_i)^2 + O\left(\frac{1}{m^3}\right), \\ p_{\beta} &= \dot{\gamma}(t_i)(t_{i+3} - t_i) + (1/2)\ddot{\gamma}(t_i)(t_{i+3} - t_i)^2 + O\left(\frac{1}{m^3}\right). \end{aligned} \quad (18)$$

Introducing  $\gamma_2(t) = \dot{\gamma}(t_i)(t - t_i) + (1/2)\ddot{\gamma}(t_i)(t - t_i)^2$  and coupling it with (18) and more-or-less uniformity results in:

$$\begin{aligned} \det(p_{\beta}, p_{\alpha}) &= \det(\gamma_2(t_{i+3}), \gamma_2(t_{i+2})) + O\left(\frac{1}{m^4}\right), \\ \det(p_1, p_{\beta}) &= \det(\gamma_2(t_{i+1}), \gamma_2(t_{i+3})) + O\left(\frac{1}{m^4}\right), \\ \det(p_{\alpha}, p_1) &= \det(\gamma_2(t_{i+2}), \gamma_2(t_{i+1})) + O\left(\frac{1}{m^4}\right). \end{aligned} \quad (19)$$

Set now  $\mathcal{P}(c, d, e) = c(1 + d - e)$ . Thus by (3) and (17) we have

$$\mathcal{P}(c, d, e) = \det(p_{\beta}, p_{\alpha}) + \det(p_1, p_{\beta}) + \det(p_{\alpha}, p_1).$$

The latter combined with (1) and (19) yields

$$\mathcal{P}(c, d, e) = (1/2)k(t_i)(t_{i+1} - t_{i+2})(t_{i+1} - t_{i+3})(t_{i+2} - t_{i+3}) + O\left(\frac{1}{m^4}\right). \quad (20)$$

Similarly, by repeating the previous analysis we obtain

$$c = (1/2)k(t_i)(t_{i+3} - t_i)(t_{i+2} - t_i)(t_{i+2} - t_{i+3}) + O\left(\frac{1}{m^4}\right). \quad (21)$$

Upon coupling (20) and (21) some factorization renders (note that curvature  $k(t)$  is here bounded and separated from zero):

$$1 + d - e = \frac{(t_{i+2} - t_{i+1})(t_{i+3} - t_{i+1})}{(t_{i+3} - t_i)(t_{i+2} - t_i)} + O\left(\frac{1}{m}\right). \quad (22)$$

As sampling is more-or-less uniform (see Definition 1) the latter amounts to

$$0 < \frac{K_l^2}{3K_u^2} \leq \frac{(t_{i+2} - t_{i+1})(t_{i+3} - t_{i+1})}{(t_{i+3} - t_i)(t_{i+2} - t_i)} + O\left(\frac{1}{m}\right).$$

Hence (16) follows.

## 6 Appendix 2

In this Appendix we show that one of the pairs  $(\alpha_{\pm}, \beta_{\pm})$  satisfies (6). More precisely, if curvature of curve  $\gamma$  satisfies  $k(t) < 0$  then the pair  $(\alpha_+, \beta_+)$  fulfills

$$1 < \alpha_+ < \beta_+. \quad (23)$$

The opposite case involves the pair  $(\alpha_-, \beta_-)$ . It is sufficient (due to the analogous argument) to justify the first case only. To prove (23) we combine more-or-less uniformity, convexity of  $\gamma$ , with (3) and (18) to obtain for  $\mathcal{R}(c, d) = cd$  which coincides with

$$\begin{aligned} \mathcal{R}(c, d) &= \det(\gamma_2(t_{i+1}), \gamma_2(t_{i+3})) + O\left(\frac{1}{m^4}\right) \\ &= (1/2)k(t_i)(t_{i+1} - t_i)(t_{i+3} - t_i)(t_{i+3} - t_{i+1}) + O\left(\frac{1}{m^4}\right). \end{aligned}$$

Hence

$$d = \frac{-(t_{i+1} - t_i)(t_{i+3} - t_{i+1})}{(t_{i+2} - t_i)(t_{i+3} - t_{i+2})} + O\left(\frac{1}{m}\right). \quad (24)$$

Similarly (taking also into account (24)) we arrive at

$$\begin{aligned} e &= \frac{-(t_{i+1} - t_i)(t_{i+2} - t_{i+1})}{(t_{i+3} - t_i)(t_{i+3} - t_{i+2})} + O\left(\frac{1}{m}\right), \\ \frac{e}{d} &= \frac{(t_{i+2} - t_{i+1})(t_{i+2} - t_i)}{(t_{i+3} - t_{i+1})(t_{i+3} - t_i)} + O\left(\frac{1}{m}\right). \end{aligned}$$

Thus the latter combined with (22) yields

$$\begin{aligned}\frac{e}{d}(1+d-e) &= \frac{(t_{i+2}-t_{i+1})^2}{(t_{i+3}-t_i)^2} + O\left(\frac{1}{m}\right), \\ e-d &= 1 - \frac{(t_{i+2}-t_{i+1})(t_{i+3}-t_{i+1})}{(t_{i+3}-t_i)(t_{i+2}-t_i)} + O\left(\frac{1}{m}\right).\end{aligned}\tag{25}$$

Hence Taylor's Theorem coupled with (4) and first equation from (25) renders

$$\rho_1 = \frac{(t_{i+2}-t_{i+1})}{(t_{i+3}-t_i)^2} + O\left(\frac{1}{m}\right).$$

Finally, the second equation from (25) and further factorization in (5) results in

$$\alpha_+ = 1 + \frac{t_{i+2}-t_{i+1}}{t_{i+1}-t_i} + O\left(\frac{1}{m}\right).\tag{26}$$

A similar analysis used to prove (26) shows that

$$\beta_+ = 1 + \frac{t_{i+3}-t_{i+1}}{t_{i+1}-t_i} + O\left(\frac{1}{m}\right).\tag{27}$$

Because sampling is more-or-less uniform the formulae (26) and (27) guarantee that  $1 < \alpha_+ < \beta_+$ .

# Highlight Removal Using Shape-from-Shading

Hossein Ragheb and Edwin R. Hancock

Department of Computer Science,  
University of York, York YO10 5DD, UK.  
{hossein,erh}@minster.cs.york.ac.uk

**Abstract.** One of the problems that hinders the application of conventional methods for shape-from-shading to the analysis of shiny objects is the presence of local highlights. The first of these are specularities which appear at locations on the viewed object where the local surface normal is the bisector of the light source and viewing directions. Highlights also occur at the occluding limb of the object where roughness results in backscattering from microfacets which protrude above the surface. In this paper, we consider how to subtract both types of highlight from shiny surfaces in order to improve the quality of surface normal information recoverable using shape-from-shading.

## 1 Introduction

Shape-from-shading is concerned with recovering surface orientation from local variations in measured brightness. There is strong psychophysical evidence for its role in surface perception and recognition [12]. Some of the pioneering work in the area was performed by Horn and his co-workers [10]. However, despite considerable effort over the past two decades, reliable shape recovery from shading information has proved an elusive goal [10]. The reasons for this are twofold. Firstly, the recovery of surface orientation from the image irradiance equation is an under-constrained process which requires the provision of boundary conditions and constraints on surface smoothness to be rendered tractable. Secondly, real-world imagery rarely satisfies these constraints. Several authors have attempted to develop shape-from-shading methods which overcome these shortcomings. For instance, Oliensis and Dupuis [18], and Bichsel and Pentland [3] have developed solutions for which shape-from-shading is not under-constrained, but which require prior knowledge of the heights of singular points of the surface. Meanwhile, Kimmel and Brookstein have shown how the apparatus of level-set theory can be used to solve the image irradiance equation as a boundary value problem [11]. Frankot and Chellappa [8] have focused on the differential geometry of the recovered surface and develop a Fourier domain approach for imposing integrability constraints. Ferrie and Lagarde [7] have used the Darboux-frame smoothing method of Sander and Zucker [21] to impose constraints from differential geometry on the recovered needle-map. A detailed comparative review of these and other related methods for shape-from-shading can be found in the recent comprehensive survey paper of Zhang, Tsai, Cryer and Shah [27].

The observation underpinning this paper is that although considerable effort has gone into the recovery of accurate surface geometry, existing shape-from-shading methods are confined to situations in which the reflectance is predominantly Lambertian. When the surface under study is shiny, then the estimated geometry may be subject to error. The main problem that can occur is that surface intensity highlights may lead to misestimation of surface curvature. The most familiar example here is that of surface specularities. These occur at locations on the surface where the local surface normal direction is the bisector of the light source and viewing directions. For this reason, if specular highlights can be accurately located, then they can provide important cues that can be used to constrain the recovery of surface shape. However, there is a less well known effect that results in limb brightening. This is due to surface roughness and results from oblique scattering from microfacets that protrude above the limb perpendicular to the line of sight.

The problem of non-Lambertian and specular reflectance has been widely studied [13,22]. For instance, Healey and Binford [9] have shown how to simplify the Beckmann distribution [1] using a Gaussian approximation to the distribution of specular angle. This simplification can be used in conjunction with the Torrance and Sparrow model [23] to model intensity variations in the analysis of surface curvature. In a comprehensive treatment of specular shape-from-shading Brelstaff and Blake [5] have analysed the geometric constraints provided by specularities, and have shown how to detect specularities using Lambertian irradiance constraints. Drawing on psychophysics, Blake and Bulthoff [4] have developed a computational model of the shape information available to a moving observer from the apparent movement of specularities. Several authors have looked critically at the physics underlying specular reflectance. For instance, Nayar, Ikeuchi and Kanade [16] have shown that the Torrance and Sparrow model [23] is applicable to the modelling of the specular lobe rather than the specular spike. Wolff [24] also has a model which combines diffuse and specular reflectance components, in which the parameters are chosen on the basis of the known physical properties of particular surfaces. In a series of recent papers, Lin and Lee have shown how specular reflections due to multiple light-sources can be located in multi-band imagery [14]. Finally, Nayar, Fang and Boulton [17] have used polarisation filters to detect specular reflection.

There has also been a considerable body of work devoted to reflectance from rough surfaces. As noted above, this process is responsible for limb brightening. Oren and Nayar [19] have developed a model which can be used to account for reflectance from surfaces with a rough microfacet structure. Dana, Nayar, Van Ginneken and Koenderink [6] have catalogued the BRDF's for 3D surface textures. Recently, Magda, Kriegman, Zickler and Belhumeur [15] have commented on how shape can be recovered from surfaces with arbitrary BRDF's. Finally, Wolff [24] has shown how the Fresnel term can be used to model reflectance from a variety of surfaces.

In this paper our aim is to incorporate both specular and rough limb reflectance into the shape-from-shading process. This is a two-step process. First, we make estimates of the local surface normals using geometric constraints on

the directions of Lambertian and specular reflectance to recover surface normal directions. The approach is a probabilistic one, which uses a mixture model to estimate the posterior mean direction of Lambertian and specular reflectance.

Once the posterior mean surface normals are to hand, then we can perform photometric correction on the original image. This is again a two-step process. First, we subtract specularities using the Torrance and Sparrow model. Second, we correct the residual non-specular component using the Oren and Nayar model. The result is a corrected Lambertian image from which both local specularities and limb-brightening effects are removed. By applying a Lambertian shape-from-shading algorithm to the corrected image, we obtain an improved estimate of the surface normal directions.

## 2 Reflectance Geometry

In this section we outline the geometry of the reflectance processes which underpin our shape-from-shading model. We adopt a two-component model in which the predominantly Lambertian surface reflectance exhibits local specular highlights.

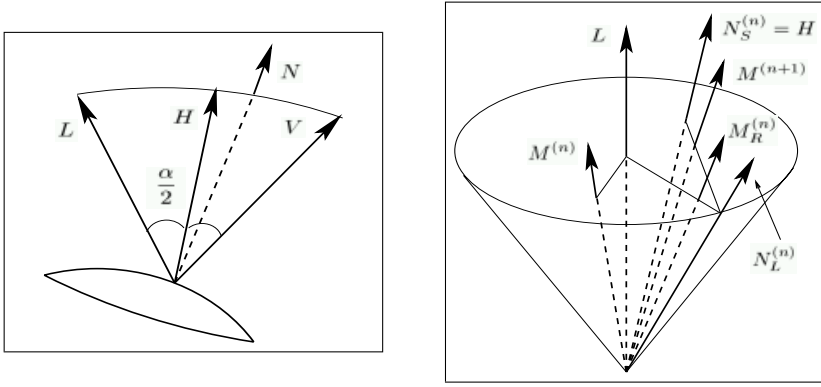
### 2.1 Specular Reflectance

The first component of our reflectance process is concerned with modelling local specular highlights on the observed surface. For specular reflection the surface normal, the light source direction and the viewing direction are coplanar. The incidence angle is equal to the angle of specular reflectance. Hence, for specular reflection, the direction of the surface normal  $\mathbf{N}_S^{(n)}$  is the bisector of the light source ( $\mathbf{L}$ ) and the viewing ( $\mathbf{V}$ ) directions and the unit-vector is  $\mathbf{N}_S^{(n)} = \frac{(\mathbf{L} + \mathbf{V})}{\|\mathbf{L} + \mathbf{V}\|}$ . It is important to stress that the surface normal for the specular reflectance component is fully constrained if the light source direction and the viewing direction are known. We therefore keep the specular surface normal direction fixed throughout our iterative recovery of the needle-map. The geometry of the specular reflectance process is illustrated in Figure 1a.

### 2.2 Lambertian Reflectance

In the case of Lambertian reflectance from a matte surface of constant albedo illuminated with a single collimated light-source, the observed intensity is independent of the viewing direction. The observed intensity depends only on the quantity of absorbed light, and this in turn is proportional to the cosine of the incidence angle. Suppose that  $\mathbf{L}$  is the unit-vector in the direction of the light source and that  $\mathbf{N}_L(i, j)$  is the unit-vector in the surface normal direction for Lambertian reflectance at the pixel  $(i, j)$ . According to Lambert's law, the observed image intensity at the pixel with coordinates  $(i, j)$  is  $E(i, j) = \mathbf{N}_L(i, j) \cdot \mathbf{L}$ .

Lambert's equation provides insufficient information to uniquely determine the surface normal direction. However, as recently observed by Worthington and



**Fig. 1.** Geometry of the specular reflectance and the needle-map update process.

Hancock [26], the equation does have a simple geometric interpretation which can be used to constrain the direction of the surface normal. The equation specifies that the surface normal must fall on the surface of a right-cone whose axis is aligned in the light-source direction  $\mathbf{L}$  and whose apex angle is  $\cos^{-1}(E)$ .

Worthington and Hancock [26] exploit this property to develop a two-step iterative process for shape-from-shading. The process commences from a configuration in which the surface normals are placed on the position on the irradiance cone where their projections onto the image plane are aligned in the direction of the local (Canny) image gradient. This geometry is illustrated in figure 1b.

In the first step, the surface normal directions are subjected to smoothing in such a way as to satisfy curvature consistency constraints. The resulting smoothed surface normal  $\hat{\mathbf{N}}_L^{(n)}$  will not fall on the irradiance cone and will hence not satisfy Lambert's law. To overcome this problem, in the second step of the process the smoothed surface normal is rotated onto the nearest location on the irradiance cone. The resulting surface normal, which satisfies Lambert's law, is

$$\mathbf{N}_L^{(n+1)} = \hat{\mathbf{N}}_L^{(n)} \cos(\theta) + (\mathbf{R} \times \hat{\mathbf{N}}_L^{(n)}) \sin(\theta) \quad (1)$$

where the rotation axis and the rotation angle are given by

$$\mathbf{R} = \frac{\hat{\mathbf{N}}_L^{(n)} \times \mathbf{L}}{\|\hat{\mathbf{N}}_L^{(n)} \times \mathbf{L}\|}, \quad \theta = \cos^{-1}(\hat{\mathbf{N}}_L^{(n)} \cdot \mathbf{L}) - \cos^{-1}(E) \quad (2)$$

In Worthington and Hancock's shape-from-shading method, which deals with matte surfaces that are free from specularities, these smoothing and back-projection steps may be interleaved and iterated until convergence, i.e. a stable needle-map is obtained.

### 3 Probabilistic Framework

The aim in this paper is to use a previously reported Bayes-decision scheme for separating the two reflectance modes [20]. We compute the *a posteriori* prob-

abilities of specular or Lambertian reflectance. This is done using the iterated conditional modes algorithm of Besag [2]. Although the method has notoriously poor global optimisation properties, we use it here because it is simple and relatively efficient.

The aim is to label pixels according to the reflectance mode from which they originated. The class identity for the pixel  $(i, j)$  at iteration  $n$  is denoted by  $\omega_{i,j}^{(n)}$ . The class-identity may be drawn from the set  $\Omega = \{S, L\}$  where  $S$  is the specular reflectance label and  $L$  is the Lambertian reflectance label. For each image location, we maintain a specular surface normal and a Lambertian surface normal which satisfy the geometric constraints outlined in Section 2. At iteration  $n$  of the algorithm the currently available estimates of the two surface normals are respectively  $\mathbf{N}_L^{(n)}(i, j)$  and  $\mathbf{N}_S^{(n)}(i, j)$ . In the case of the specular component, the normal direction is in the direction of local specular reflection, and does not change with iteration number. In the case of Lambertian reflectance, the surface normal direction varies with iteration number, but is always projected to be positioned on the irradiance cone.

To develop our decision process, we require two probabilistic modelling ingredients. The first of these are separate probability density functions which can be used to represent the distributions of surface normals for the two reflectance components. We evaluate these densities at the posterior mean surface normal  $\mathbf{M}^{(n)}(i, j)$  computed at iteration  $n$ . The reason for doing this is that the current values of the two normals are guaranteed to satisfy the geometric constraints outlined in Section 2. As a result, they will be associated with vanishing angular error. Accordingly, we let  $q_{i,j}^{(n)}(L) = p(\mathbf{M}^{(n)}(i, j) | \omega_{i,j}^{(n)} = L)$  be the probability distribution for the posterior mean surface normal under the Lambertian reflectance model. Similarly, we let  $q_{i,j}^{(n)}(S) = p(\mathbf{M}^{(n)}(i, j) | \omega_{i,j}^{(n)} = S)$  denote the distribution function for the posterior mean surface normal for the specular reflectance component.

The second probabilistic ingredient is a smoothness prior for the selected surface normal. This component of the model incorporates contextual information. Indexing the surface normals according to their pixel locations, suppose that  $\Gamma_{i,j}^{(n)} = \{\mathbf{M}^{(n)}(k, l) | (k, l) \in G_{i,j}\}$  is the set of posterior mean surface normals in the neighbourhood  $G_{i,j}$  of the pixel  $(i, j)$ . We let  $P_{i,j}^{(n)}(L) = P(\mathbf{N}_L^{(n)}(i, j) | \Gamma_{i,j}^{(n)})$  be the conditional probability (or smoothness prior) of the Lambertian surface normal at the location  $(i, j)$  given the field of surrounding posterior mean surface normals. With these ingredients, then according to the iterated conditional modes, the probability that the pixel  $(i, j)$  belongs to the Lambertian class at iteration  $n$  is

$$P(\omega_{i,j}^{(n)} = L | \mathbf{M}^{(n)}(i, j)) = \frac{q_{i,j}^{(n)}(L)P_{i,j}^{(n)}(L)}{\sum_{A \in \Omega} q_{i,j}^{(n)}(A)P_{i,j}^{(n)}(A)} \quad (3)$$

The probability that the surface normal belongs to the specular class is the complement, These probabilities can be used to separate the two reflectance modes. With these probabilities to hand, we can update the estimate of the



posterior mean surface normal in the following manner

$$\begin{aligned} \mathbf{M}^{(n+1)}(i, j) &= \mathbf{N}_S^{(n)}(i, j)P(\omega_{i,j}^{(n)} = S | \mathbf{M}^{(n)}(i, j)) \\ &+ \mathbf{N}_L^{(n)}(i, j)P(\omega_{i,j}^{(n)} = L | \mathbf{M}^{(n)}(i, j)) \end{aligned} \quad (4)$$

## 4 Probability Distributions

To apply the Bayes framework for the posterior mean surface normal estimation, we require probability distributions for Lambertian and specular reflectance, together with a smoothness prior for the surface normal directions.

**Specular Reflection.** The modelling of specular reflectance and specular highlights has attracted considerable attention in the computer vision and computer graphics communities [23,1,9]. Here we are interested in two approaches to the problem. The first of these is concerned with modelling specular intensities. Here the model of Torrance and Sparrow [23] captures the physics of scattering by the micro-facet structure of a surface. The second approach is to model the angular distribution associated with the reflected light in the proximity of specularities. Here the Beckmann distribution [1] provides a relatively simple model which captures the angular shape of the specular spike. Healey and Binford [9] have an alternative model which can be used to model the distribution of specular intensities for regions of high surface curvature.

We use the Torrance and Sparrow model to distribution of specular intensities. According to this model, the specular intensity is given by

$$I_S^{(n)}(i, j) = \left( K \frac{F}{\pi} \right) \frac{G}{(\mathbf{M}^{(n)} \cdot \mathbf{V})} \frac{D(\alpha)}{(\mathbf{M}^{(n)} \cdot \mathbf{L})} \quad (5)$$

The model is controlled by four terms. The first of these is the Fresnel term which is close to unity, i.e.  $F \cong 1.0$ . Secondly, there is the geometrical attenuation factor

$$G = \min \left[ 1, 2 \frac{(\mathbf{M}^{(n)} \cdot \mathbf{N}_S)(\mathbf{M}^{(n)} \cdot \mathbf{V})}{(\mathbf{V} \cdot \mathbf{N}_S)}, 2 \frac{(\mathbf{M}^{(n)} \cdot \mathbf{N}_S)(\mathbf{M}^{(n)} \cdot \mathbf{L})}{(\mathbf{V} \cdot \mathbf{N}_S)} \right] \quad (6)$$

Thirdly, there is the facet slope function which we model using the Beckmann distribution [1] to model the distribution of the angle  $\alpha = \cos^{-1}(\mathbf{M}^{(n)}(i, j) \cdot \mathbf{N}_S^{(n)}(i, j))$  between the posterior mean surface normal  $\mathbf{M}^{(n)}(i, j)$  and the predicted direction of the specular spike  $\mathbf{N}_S^{(n)}$ . The distribution is

$$D(\alpha) = \frac{1}{\sigma_S^2 \cos^4 \alpha} \exp \left[ - \left( \frac{\tan(\alpha)}{\sigma_S} \right)^2 \right] \quad (7)$$

where  $\sigma_S$  is a parameter which controls the angular shape of the distribution. This distribution can be used to model the shape of both the specular spike and the specular lobe. It makes no attempt to model the distribution of specular intensities.

Fourthly, and finally,  $K$  is a constant which normalizes the equation. We assume that the observed specular intensities follow a Gaussian distribution with variance  $\sigma_S^2$ . Under these assumptions we can write

$$q_{i,j}^{(n)}(S) = \frac{1}{\sqrt{2\pi}\sigma_S} \exp \left[ -\frac{1}{2} \left( \frac{E(i,j) - I_S^{(n)}(i,j)}{\sigma_S} \right)^2 \right] \quad (8)$$

This distribution models the variation in specular intensity resulting from the physical variation in specular reflectance direction.

**Lambertian Reflectance.** Our model of the Lambertian reflectance process assumes that the observed intensity values follow a Gaussian distribution with variance  $\sigma_L^2$ . The mean intensity is  $(\mathbf{M}^{(n)} \cdot \mathbf{L})$ . Under these assumptions we can write

$$q_{i,j}^{(n)}(L) = \frac{1}{\sqrt{2\pi}\sigma_L} \exp \left[ -\frac{1}{2} \left( \frac{E(i,j) - \mathbf{M}^{(n)}(i,j) \cdot \mathbf{L}}{\sigma_L} \right)^2 \right] \quad (9)$$

**Smoothness Prior.** Our model for the surface normal smoothness prior is based on the average value of the inner product of the surface normal at the location  $(i,j)$  with the surrounding field of surface normals. We write

$$P_{i,j}^{(n)}(\Lambda) = \frac{1}{2|G_{i,j}|} \left[ |G_{i,j}| + \sum_{(k,l) \in G_{i,j}} \mathbf{N}_\Lambda^{(n)}(i,j) \cdot \mathbf{M}^{(n)}(k,l) \right] \quad (10)$$

When the posterior mean surface normals from the neighbourhood  $G_{i,j}$  are aligned in the direction of  $\mathbf{N}_\Lambda^{(n)}(i,j)$ , then  $P_{i,j}^{(n)}(\Lambda) = 1$ , the larger the misalignment then the smaller the value of smoothness prior.

## 5 Specularity Subtraction

Having described the Bayes framework and the associated two-mode reflectance model, we are now in a position to perform specularity removal. In this section we describe a shape-from-shading algorithm which leads to images free of specularities.

We commence by initialising the algorithm. The initial Lambertian surface normal  $\mathbf{N}_L^{(0)}(i,j)$  is constrained to lay on the irradiance cone in the direction of the image gradient. The subsequent iterative steps of the algorithm are as follows:

- 1: The field of posterior mean surface normals (initially equal to  $\mathbf{N}_L^{(0)}$ ) is subjected to local smoothing. Here we use the curvature sensitive smoothing method [26]. The smoothed surface normal is denoted by  $\mathbf{M}_R^{(n)}(i,j)$ .

- 2: We update the current estimate of the Lambertian surface normal by projecting the smoothed posterior mean surface normal onto the nearest location on the irradiance cone. This gives us the revised surface normal  $\mathbf{N}_L^{(n)}$ .
- 3: With the  $\mathbf{M}_R^{(n)}(i, j)$  to hand we compute the conditional measurement densities  $q_{i,j}^{(N)}(L)$  and  $q_{i,j}^{(N)}(S)$  for the two reflectance modes. Taking  $\mathbf{M}_R^{(n)}(i, j)$ ,  $\mathbf{N}_L^{(n)}$  and  $\mathbf{N}_S^{(n)}$ , we compute the smoothness priors  $P_{i,j}^{(n)}(L)$  and  $P_{i,j}^{(n)}(S)$ . Then, we compute the updated *a posteriori* probabilities for both reflectance modes.
- 4: Using  $\mathbf{N}_L^{(n)}$  and  $\mathbf{N}_S^{(n)}$  and the updated *a posteriori* probabilities, we compute the new posterior mean surface normal  $\mathbf{M}^{(n+1)}(i, j)$  and we return to step 1.

The steps of the algorithm are summarised in Figure 1b. The posterior mean surface normals delivered by our shape-from-shading algorithm can be used for the purposes of reconstructing the specular intensity component  $I_S$  using the Torrance-Sparrow model given in equation 5. With the reconstructed specular intensity to hand, we can compute the matte reflectance component  $I_M(i, j) = E(i, j) - I_S(i, j)$ .

## 6 Correcting for Limb-Brightening

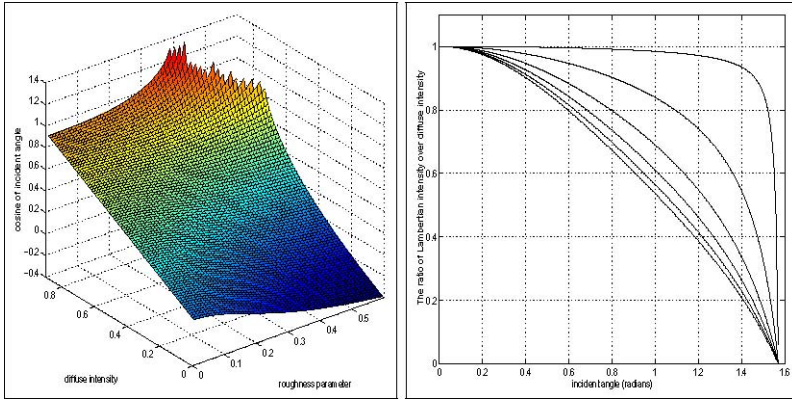
As mentioned earlier, there may also surface brightness anomalies due to rough reflectance from the limbs of objects. Our aim in this section is to show how the Nayar and Oren model can be used to further correct the images obtained by specular subtraction for limb-brightening.

It is well-known that there are a large number of situations where Lambert's law is significantly in error. These include locations near the occluding contour of objects under any illumination conditions. It also applies to situations where the angle of illumination incidence is greater than  $50^\circ$  relative to the viewing direction. Here there will be significant departures from Lambertian reflectance both near the occluding boundary and over a large portion of object area bounded on one side by the shadow boundary with respect to illumination [25]. These non-Lambertian effects are observable for both shiny and rough surfaces.

Oren and Nayar have a qualitative reflectance model for rough surfaces [19]. For a point on a rough surface with illuminant incidence angle  $\theta_i$  and viewing, or reflectance angle,  $\theta_r$  the reflectance functions is

$$L_r(\theta_i, \theta_r, \phi_r - \phi_i; \sigma) = \frac{\rho}{\pi} E_0 \cos(\theta_i) (A + B \max[0, \cos(\phi_r - \phi_i)]) \sin(\alpha) \tan(\beta) \quad (11)$$

where  $A = 1.0 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33}$ ,  $B = 0.45 \frac{\sigma^2}{\sigma^2 + 0.09}$  and  $\alpha = \max[\theta_i, \theta_r]$ ,  $\beta = \min[\theta_i, \theta_r]$ . It is important to note that the model reduces to the Lambertian case when  $\sigma = 0$ . Here, we aim to utilize this model to deduce a corrected Lambertian reflectance image from the matte component delivered by our specular



**Fig. 2.** Plots showing the behaviour of the Oren and Nayar model for rough surfaces.

subtraction method. To do this, we assume that the surface roughness  $\sigma$  is almost constant and the reflectance measurements are obtained in the plane of incidence i.e. ( $\phi_r = \phi_i = 0$ ). We also confine our attention to the case where the angle between the light source and the viewing directions is small, i.e.  $\theta_r = \theta_i = \theta$ . With these two restrictions, we can write  $\cos(\phi_r - \phi_i) = 1$  and  $\alpha = \beta = \theta$ . Hence, the non-specular (or diffuse) intensity predicted by the simplified Oren and Nayar model is

$$I_M(i, j) = A \cos \theta + B \sin^2 \theta \quad (12)$$

Hence, the matte intensity consists of two components. The first of these is a Lambertian component  $A \cos \theta$ . The second is the non-Lambertian component  $B \sin^2 \theta$  which takes on its maximum value where  $\theta = \frac{\pi}{2}$ , i.e. close to the occluding boundary. To perform Lambertian correction, we proceed as follows. At every pixel location, we use Equation (12) to estimate the angle  $\theta$  using the subtracted matte intensity and solving the resulting quadratic equation in  $\cos \theta$ . The solution is

$$\cos \theta = \frac{A \mp \sqrt{A^2 - 4B(I_M(i, j) - B)}}{2B} \quad (13)$$

We take the sign above which results in a value of  $A \cos \theta$  which is closest to the matte intensity  $I_M$  (in the majority of cases this involves taking the solution associated with the minus sign). This hence allows us to reconstruct the corrected Lambertian reflectance image  $I_L = A \cos \theta$ . It also gives us an estimate of the opening angle of the Lambertian reflectance cone. This can then be used in the Worthington and Hancock shape-from-shading scheme which assumes the Lambertian reflectance model to recover improved surface normal estimates.

In Figure 2a we show the Lambertian reflectance  $\cos \theta$  (equation 13) as a function of the roughness parameter  $\sigma$  and the matte intensity  $I_M$ . When the roughness is zero, then the Lambertian and matte intensities are equal to one another. When the roughness increases, then the departures from Lambertian reflectance become more marked.

In Figure 2b we plot the ratio  $\frac{I_L}{I_M}$  as a function of the incidence angle  $\theta$ . The different curves are for different values of the roughness parameter  $\sigma$ . For zero roughness, the ratio is flat, i.e. the reflectance is purely Lambertian. As the roughness increases, then so the value of the ratio decreases by increasing the incidence angle. For normal incidence, the ratio is always unity, i.e. the reflectance is indistinguishable from the Lambertian case, whatever the value of the roughness parameter is.

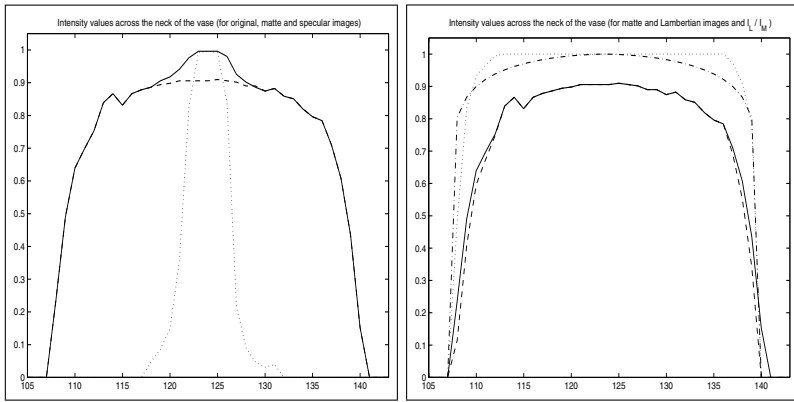
## 7 Experiments

The images used in our experiments have been captured using an Olympus 10E camera. The objects studied are made of white porcelain and are hence shiny. Each object has been imaged under controlled lighting conditions in a darkroom. The objects have been illuminated using a single collimated tungsten light source. The light source direction is recorded at the time the images are captured.

To ground-truth the surface highlight removal process, we have used a pair of polaroid filters. We have placed the first filter between the light source and the illuminated object. The second filter was placed between the illuminated object and the camera. For each object we have collected a pair of images. The first of these is captured when the second filter (i.e. the one between the camera and the object) is rotated until there is maximal extinction of the observed specularities. The second image is obtained when the polaroid is rotated through 90 degrees, i.e. there is minimal extinction of the specularities. We refer to the polarisation conditions of the former image as “uncrossed” and of the latter as “crossed”.

In Figure 4 we show the results obtained for three of the objects used in our study. The objects are a porcelain bear, a porcelain vase and a porcelain urn. The top row of the figure shows the images obtained with uncrossed polaroids while the second row shows the images obtained with crossed polaroids. The third row shows the difference between the crossed and uncrossed polaroid images. The strongest differences occur at two different locations. Firstly, there are the positions of specularities. From the uncrossed polaroid images it is clear that there are several quite small specular reflections across the surface of the bear. The vase has larger specularities on the neck and the centre of the bulb. The urn has a complex pattern of specularities around the handles. From the crossed polaroid images it is clear that most of the specular structure is removed. The second feature in the difference images are the locations of occluding object limbs, where oblique scattering occurs.

In the fourth row of Figure 4, we show the matte images  $I_M$  obtained after specular subtraction using the Torrance and Sparrow model. The fifth row shows the reconstructed specular intensity obtained using the Torrance and Sparrow model, i.e.  $I_S$ . The sixth row shows the difference between the corrected matte images in the fourth row and the uncrossed polaroid images in the top row. Turning our attention to the matte images and the specular images, it is clear that for each of the objects the specular structure is cleanly removed and the matte appearance is close to that obtained with the crossed polaroids. Also,



**Fig. 3.** Intensity plots for different reflectance components across the neck of the vase.

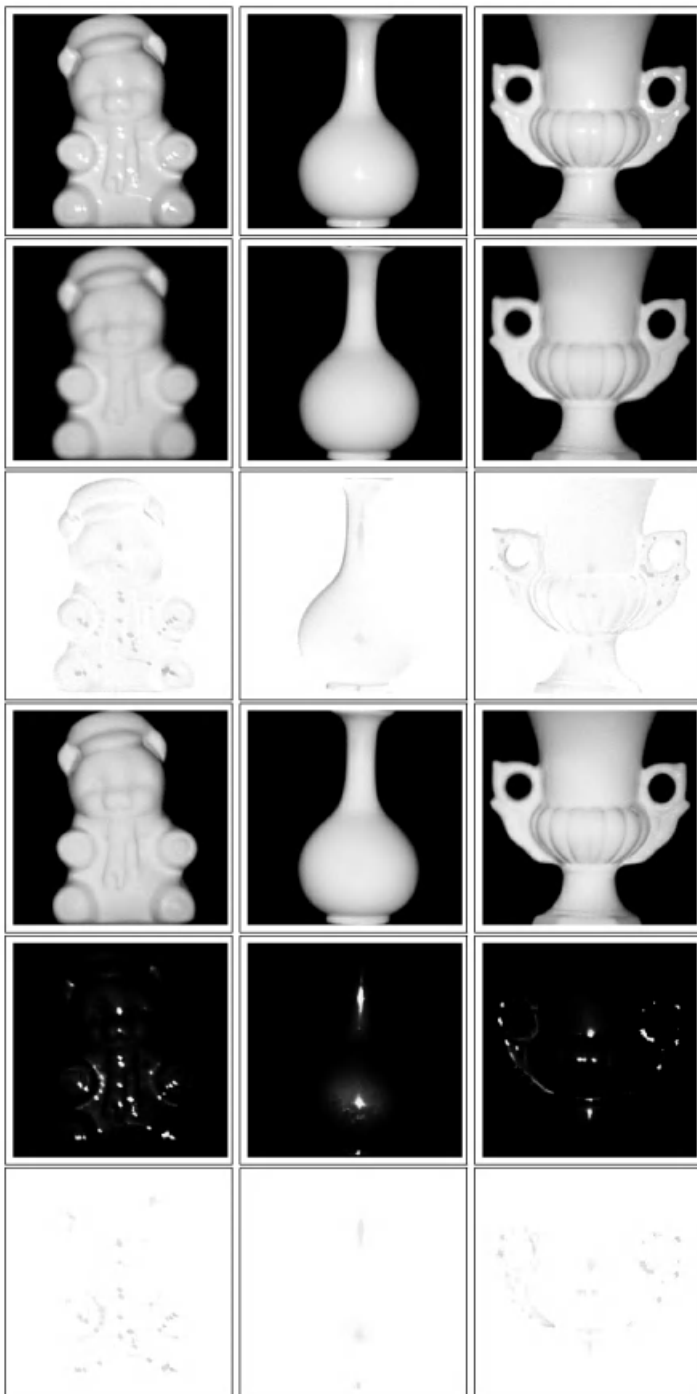
the pattern of specularities obtained in each case corresponds to that obtained by subtracting the crossed polaroid images from the uncrossed polaroid images.

In Figure 5 we investigate the shape information recoverable. The top row shows the Lambertian images after correction for rough limb reflectance using the simplified Oren and Nayar model, i.e.  $I_L$ . The second row shows the difference between the corrected Lambertian images and the uncrossed polaroid images.

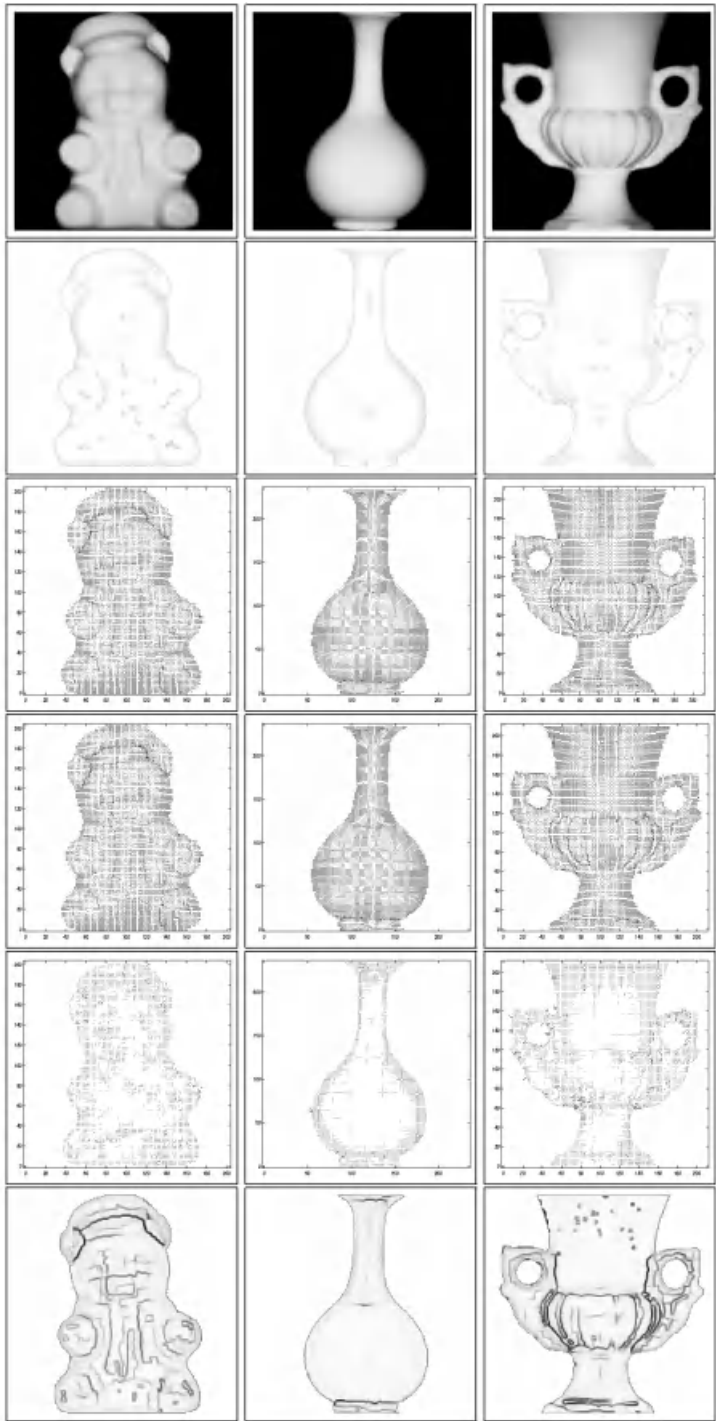
In the third row we show the needle-maps obtained when we apply shape-from-shading to the images obtained only by specular subtraction, i.e.  $I_M$ . In the fourth row of the figure we show the needle maps obtained when the shape-from-shading is applied to the corrected Lambertian images ( $I_L$ ) appearing in the top row of this Figure. The fifth row of Figure 5 shows the difference in needle-map directions for the matte ( $I_M$ ) and Lambertian images ( $I_L$ ). Here the main differences occur at the limbs of the objects.

The sixth row of Figure 5 show the curvedness estimated using the surface normals delivered by the corrected Lambertian images. In the case of the urn the ribbed structure emerges well. The complex surface structure of the bear, in particular the boundaries of the arms and legs, is clearly visible. For the urn the symmetric structure of the neck and the bulb is nicely preserved.

In Figure 6 we show some image reconstructions obtained from the Lambertian image surface normals. The images in the figure are organised into three pairs of rows. In each pair the top row shows the image reconstructions. The lower row shows the differences between the reconstructed images and the subtracted matte images (the fourth row of Figure 4). The first pair of rows show the results obtained using a simple Lambertian model, the second pair are the results obtained with the Oren and Nayar method, and the final pair show the results obtained using Wolff's Fresnel model. The images in the second pair show almost no difference with the matte components. However, the remaining two pairs show stronger patterns of difference. This suggests that the correction process using the Oren and Nayar model has been performed successfully. The reason for this is that the reconstructed images are almost identical to the matte

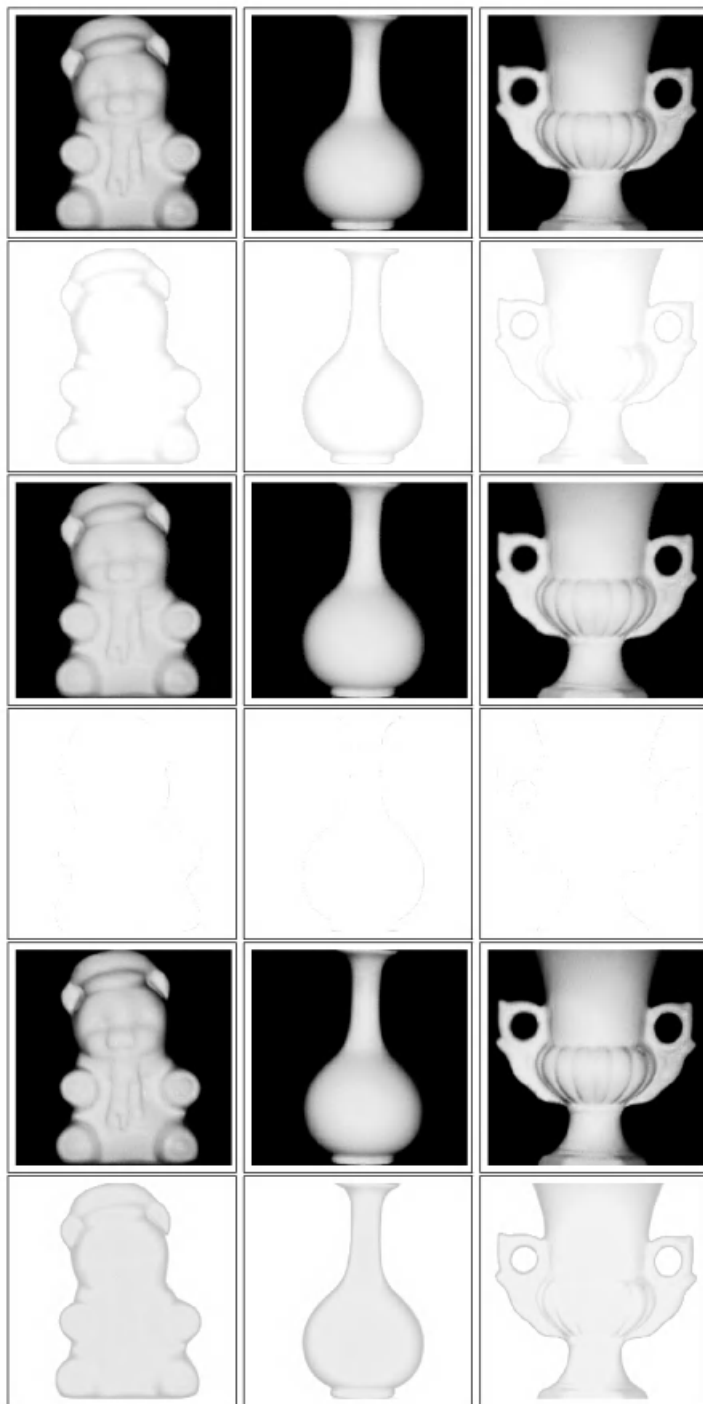


**Fig. 4.** Applying our specular SFS to separate the specular and matte components.



**Fig. 5.** Surface normals obtained by running SFS over matte and Lambertian images.





**Fig. 6.** Reconstructed images using Lambertian, Oren-Nayar and Wolff models.

images. It is also clear that the images reconstructed using the Oren and Nayar model are brighter than the ones reconstructed using the Lambertian model, whereas, the images reconstructed using the Wolff model are darker than both of them.

In Figure 3 we provide some analysis of the different reflectance models used in our experiments. In the left hand panel of the figure, the solid curve is the intensity cross-section along a horizontal line crossing the uncrossed image of the neck of the vase shown in Figure 4. The dashed-curve shows the matte image  $I_M$  resulting from specular subtraction, while the dotted curve is the specular component  $I_S$  reconstructed using the Torrance and Sparrow model. The specularity on the neck is clearly visible as a peak in the solid curve. This peak is cleanly subtracted in the matte (dashed) curve. In the right-hand panel we focus on the corrected Lambertian image. Here the solid curve is the matte reflectance  $I_M$  obtained by specular subtraction. The dashed curve is the corrected Lambertian reflectance  $I_L$ . The differences between the two curves are small except at the limbs of the object. To examine the effect of the model in more detail, the dotted curve shows the ratio of corrected Lambertian and matte reflectance  $\rho = \frac{I_L}{I_M}$ . The ratio drops rapidly towards zero as the limbs are approached. Also shown on the plot as a dash-dot curve is the predicted value of the ratio based on the assumption that the object has a circular cross-section. If  $x$  is the distance from the centre and  $r$  is the radius of the circle, then value of the ratio at a distance  $x$  from the centre is  $\rho(x) = \frac{A\sqrt{1-(\frac{x}{r})^2}}{A\sqrt{1-(\frac{x}{r})^2}+B(\frac{x}{r})^2}$ . This simple model is in reasonable agreement with the empirical data.

## 8 Conclusions

In this paper we have shown how to use shape-from-shading to perform photometric correction of images of shiny objects. Our approach is to use estimated surface normal directions together with reflectance models for specular and rough reflectance to perform specularity removal and rough limb-correction. Specularities are modelled using the Torrance and Sparrow model while the rough limb brightening is modelled using the Oren and Nayar model. We commence by using an iterated conditional modes algorithm to extract surface normals using a mixture of specular and matte reflectance directions. The resulting surface normal directions are used to perform specularity subtraction. Finally, we correct the residual matte reflectance component for rough limb scattering using the Oren and Nayar model. The resulting corrected Lambertian images can be used as input to a conventional shape-from-shading algorithm and result in improved recovery of object-geometry.

## References

1. P. Beckmann and A. Spizzochino, *The Scattering of Electromagnetic Waves from Rough Surfaces*, Pergamon, New York, 1963.
2. J. Besag, "On the statistical analysis of dirty pictures," *J. R. Statis. Soc. Lond. B*, Vol. 48, pp. 259-302, 1986.

3. M. Bichsel and A.P. Pentland, "A Simple Algorithm for Shape from Shading," *CVPR*, pp. 459-465, 1992.
4. A. Blake and H. Bulthoff, "Shape from Specularities: computation and psychophysics," *Phil Trans R. Soc. Lond. B*, Vol. 331, pp. 237-252, 1991.
5. G. Brelstaff and A. Blake, "Detecting Specular Reflection Using Lambertian Constraints," *ICCV*, pp. 297-302, 1988.
6. K. Dana, S. Nayar, B. Van Ginneken and J. Koenderink, "Reflectance and Texture of Real-World Surfaces," *CVPR*, pp. 151-157, 1997.
7. F.P. Ferrie and J. Lagarde, "Curvature Consistency Improves Local Shading Analysis," *ICPR*, Vol. I, pp. 70-76, 1990.
8. R.T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms," *ICCV*, pp. 118-125, 1987.
9. G.H. Healey and T.O. Binford, "Local shape from specularity," *ICCV*, pp. 151-160, 1987.
10. B.K.P. Horn and M.J. Brooks, "The Variational Approach to Shape from Shading," *CVGIP*, Vol. 33, No. 2, pp. 174-208, 1986.
11. R. Kimmel and A.M. Bruckstein, "Tracking Level-sets by Level-sets: A Method for Solving the Shape from Shading Problem," *CVIU*, Vol. 62, No. 1, pp. 47-58, 1995.
12. J.J. Koenderink, A.J. van Doorn and A.M.L. Kappers, "Surface Perception in Pictures," *Perception and Psychophysics*, Vol. 52, No. 5, pp. 487-496, 1992.
13. C.C.J. Kuo and K.M. Lee, "Shape from Shading With a Generalized Reflectance Map Model," *CVIU*, Vol. 67, No. 2, pp. 143-160, 1997.
14. S. Lin and S.W. Lee, "Estimation of Diffuse and Specular Appearance," *ICCV*, pp. 855-860, 1999.
15. S. Magda, D. Kriegman, T. Zickler and P. Belhumeur, "Beyond Lambert: Reconstructing Surfaces with Arbitrary BRDFs," *ICCV*, Vol. 2, pp. 391-399, 2001.
16. S.K. Nayar, K. Ikeuchi and T. Kanade, "Surface Reflection: Physical and Geometrical Perspectives," *PAMI*, Vol. 13, No. 7, pp. 611-634, 1991.
17. S.K. Nayar, X. Fang and T. Boult, "Removal of specularities using color and polarization," *CVPR*, pp. 583-590, 1993.
18. J. Oliensis and P. Dupuis, "A Global Algorithm for Shape from Shading," *CVPR*, pp. 692-701, 1993.
19. M. Oren and S.K. Nayar, "Generalization of the Lambertian Model and Implications for Machine Vision," *IJCV*, vol. 14, No. 3, pp. 227-251, 1995.
20. H. Ragheb, and E.R. Hancock, "Separating Lambertian and Specular Reflectance Components using Iterated Conditional Modes," *BMVC*, pp. 541-552, 2001.
21. P.T. Sander and S.W. Zucker, "Inferring surface trace and differential structure from 3D images," *PAMI*, Vol. 12, No. 9, pp 833-854, 1990.
22. H.D. Tagare and R.J.P. deFigueiredo, "A Theory of Photometric Stereo for a Class of Diffuse Non-Lambertian Surfaces," *PAMI*, Vol. 13, No. 2, pp. 133-151, 1991.
23. K. Torrance and E. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces," *JOSA*, Vol. 57, pp. 1105-1114, 1967.
24. L.B. Wolff, "On The Relative Brightness of Specular and Diffuse Reflection," *CVPR*, pp. 369-376, 1994.
25. L.B. Wolff, S.K. Nayar and M. Oren, "Improved Diffuse Reflection Models for Computer Vision," *IJCV*, Vol. 30, No. 1, pp. 55-71, 1998.
26. P.L. Worthington and E.R. Hancock, "New Constraints on Data-closeness and Needle-map consistency for SFS," *PAMI*, Vol. 21, No. 11, pp. 1250-1267, 1999.
27. R. Zhang, P. Tsai, J.E. Cryer and M. Shah, "Shape from Shading: A Survey," *PAMI*, Vol. 21, No. 8, pp. 690-706, 1999.

# A Reflective Symmetry Descriptor

Michael Kazhdan, Bernard Chazelle, David Dobkin, Adam Finkelstein, and  
Thomas Funkhouser

Princeton University, Princeton NJ 08544, USA

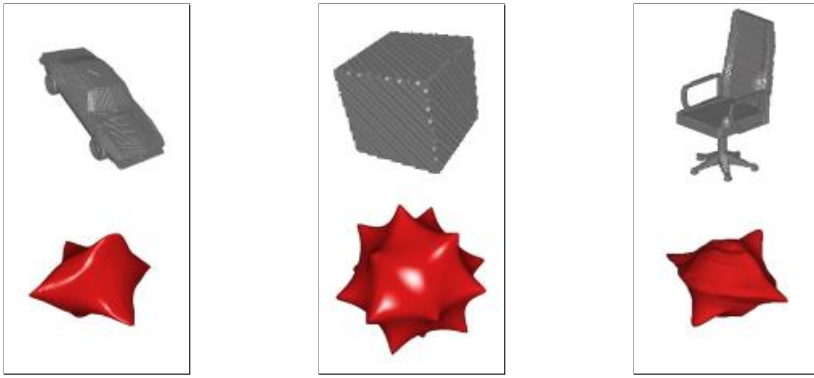
**Abstract.** Computing reflective symmetries of 2D and 3D shapes is a classical problem in computer vision and computational geometry. Most prior work has focused on finding the main axes of symmetry, or determining that none exists. In this paper, we introduce a new *reflective symmetry descriptor* that represents a measure of reflective symmetry for an arbitrary 3D voxel model for all planes through the model's center of mass (even if they are not planes of symmetry). The main benefits of this new shape descriptor are that it is defined over a canonical parameterization (the sphere) and describes global properties of a 3D shape. Using Fourier methods, our algorithm computes the symmetry descriptor in  $O(N^4 \log N)$  time for an  $N \times N \times N$  voxel grid, and computes a multiresolution approximation in  $O(N^3 \log N)$  time. In our initial experiments, we have found the symmetry descriptor to be useful for registration, matching, and classification of shapes.

## 1 Introduction

Detecting symmetry in 3D models is a well studied problem with applications in a large number of areas. For instance, the implicit redundancy in symmetric models is used to guide reconstruction [1,2], axes of symmetry provide a method for defining a coordinate system for models [3], and symmetries are used for shape classification and recognition [4,5].

Despite its intuitive appeal, symmetry has been under-utilized in computer-aided shape analysis. Most previous methods have focused only on discrete detection of symmetries – i.e., classifying a model in terms of its symmetry groups (either a model has a symmetry, or it does not) [2,6,7,8,9,10,11,12,13]. Accordingly, they provide limited information about the overall shape of an object, and they are not very useful for shapes that have no symmetries. In contrast, in the context of shape analysis, we believe that it is just as important to know that a model does not have a particular symmetry, as it is to know that it does.

The objective of our work is to define a continuous measure of reflective symmetry (over any plane for any 3D model) and use it to build a concise shape signature that is useful for registration, matching, and classification of 3D objects. Our approach is to define a *reflective symmetry descriptor* as a 2D function that gives the measure of invariance of a model with respect to reflection about each plane through the model's center of mass. For example, Figure 1 shows a car, cube and chair (top) and their corresponding reflective symmetry descriptors (bottom). The descriptors are drawn by scaling unit vectors on the sphere in proportion to the measure of reflective symmetry about the plane through the center of mass and normal to the vector. Note that the reflective symmetry



**Fig. 1.** A visualization of the reflective symmetry descriptor for a car, a cube, and a chair. The visualization is obtained by scaling unit vectors on the sphere in proportion to the measure of reflective symmetry about the plane through the center of mass, normal to the vector.

descriptor provides a continuous measure of reflective symmetry for all planes through the center of mass, and has peaks corresponding to the planes of symmetry, or near symmetry, of the models. For example, the symmetry descriptor of the chair in Figure 1 has strong peaks corresponding to its left-right symmetry, it has smaller horizontal and vertical peaks corresponding to the seat and back of the chair, but it also has a strong peak corresponding to the plane that reflects the back of the chair into the seat. Thus, in the case of the chair, the reflective symmetry descriptor describes not only the different parts of the chair, but also their spatial relationships.

For shape analysis tasks, the potential advantages of the reflective symmetry descriptor are four-fold. First, it characterizes the global shape of the object, and thus it is well-suited for the matching of whole objects (as is often needed for searching large databases of 3D objects). Second, it is defined over a canonical 2D domain (the sphere), and thus it provides a common parameterization for arbitrary 3D models that can be used for alignment and comparison. Third, it is insensitive to noise and other small perturbations in a 3D model, since each point on the symmetry descriptor represents an integration over the entire volume, and thus similar models which only differ in their fine details have similar symmetry descriptors. Finally, it describes the shape of an object in terms of its symmetry features, which provide distinguishing shape information for many objects (look around your office and consider classifying objects based on their symmetries). This approach is quite different from existing shape descriptors, and thus in addition to being useful on its own, it may be helpful to use in conjunction with other representations.

In this paper, we describe our initial research in defining, computing, and using reflective symmetry descriptors. Specifically, we make the following contributions: (1) we define a new continuous measure for the reflective symmetry of a 3D voxel model with respect to a given plane, (2) we describe efficient algorithms to compute the reflective symmetry measure for all planes through the center of mass of a 3D model, and (3) we present experimental results evaluating the utility of reflective symmetry descriptors for registration and classification of 3D models. In our tests, we find that reflective symmetry

descriptors are more effective than commonly used shape descriptors (e.g., moments [14] and shape distributions[15]) for registering and classifying 3D models.

The remainder of the paper is organized as follows. Section 2 contains a brief review of related work. Next, Section 3 introduces our new measure of reflective symmetry for voxel models, and Section 4 describes an efficient algorithm for computing a shape descriptor based on this measure. Section 5 discusses some of the properties of the reflective symmetry descriptor, while Section 6 presents experimental results acquired during tests in shape registration and classification applications. Finally, Section 7 contains a brief summary of our work and a discussion of topics for future work.

## 2 Related Work

Existing approaches for reflective symmetry detection have mainly focused on finding perfect symmetries of a 2D or 3D model [6,7]. For instance, early work in this area is based on efficient substring matching algorithms (e.g., [16]). However, since substring matching is inherently a binary question, these algorithms can only find perfect symmetries and are highly unstable in the presence of noise and imprecision; thus they are not suitable for most shape registration and matching applications.

In the case of voxel grids, methods for symmetry detection have been proposed using the covariance matrix [11,12], taking advantage of the fact that eigenspaces of the covariance matrix must be invariant under the symmetries of the model. These methods are efficient and work in all dimensions but have the disadvantage that they only work when the eigenspaces of the covariance matrix are all one-dimensional. In the case of the cube, for example, the covariance matrix is a constant multiple of the identity, every vector is an eigenvector, and no candidate axes of symmetry can be determined. Additionally, the covariance matrix can only identify candidate axes and does not determine a measure of symmetry. So, further evaluation needs to be performed to establish the quality of these candidates as axes of symmetry. Methods for symmetry detection in 2D using more complex moments and Fourier decomposition have also been described [8,9,10,13], though their dependence on the ability to represent an image as a function on the complex plane makes them difficult to generalize to three-dimensions.

In the work most similar to ours, Marola [8] presents a method for measuring symmetry invariance of 2D images. However, because of its use of autocorrelation, the method cannot be extended directly to three-dimensional objects. In related work, Zabrodsky, Peleg and Avnir [2] define a continuous *symmetry distance* for point sets in any dimension. Unfortunately, it relies on the ability to first establish point correspondences, which is generally difficult. Additionally, while the method provides a way of computing the symmetry distance for an individual plane of reflection, it does not provide an efficient algorithm for characterizing a shape by its symmetry distances with respect to multiple planes.

Our approach differs from previous work on symmetry detection in that we aim to construct a *shape descriptor* that can be used for registration, matching, and classification of 3D shapes based on their symmetries. The key idea is that the measure of symmetry with respect to any plane is an important feature of an object's shape, even if the plane does not correspond to a reflective symmetry of the shape. By capturing this idea in a

structure defined on a canonical parameterization, we can compare models by comparing their symmetry descriptors. This basis for comparison provides a means for shape registration, matching, and classification. In this respect, our goals are similar to previously described shape descriptors [17,18,19,20], of which some recent examples include spin images [21], harmonic shape images [22], shape contexts [23,24], and Extended Gaussian Images [25]. Similarly, our descriptor is related to several shape representations that characterize symmetries with respect to *local* axes, such as medial axes [26], shock graphs [27], and skeletons [28,29]. However, our reflective symmetry descriptor differs from these structures in that it characterizes global symmetry features of a 3D model, and thus it provides shape information orthogonal to these other descriptors.

In the following sections, we describe our methods for computing the reflective symmetry descriptor. There are two main challenges. First, we must describe a new notion of symmetry distance that can be used to measure the invariance of a 3D voxel model with respect to reflection about any plane. Second, we must develop an algorithm for computing the reflective symmetry descriptor that is more efficient than the brute force  $O(N^5)$ , algorithm for  $N \times N \times N$  voxel grids.

### 3 Defining the Symmetry Distance

The first issue is to define a measure of symmetry for a 3D model with respect to reflection about a plane. While previous work has proposed symmetry measures for 2D images and 3D point sets, we seek such a measure for 3D models based on a solid mathematical framework. This allows us to prove valuable properties of the descriptor.

We define the symmetry distance of a function with respect to a given plane of reflection as the  $L^2$ -distance to the nearest function that is invariant with respect to the reflection. Specifically, we treat a voxel model as a regular sampling of a function and use the  $L^2$ -norm on the space of functions. For a function  $f$  and a reflection  $\gamma$  this translates into the equation:

$$SD(f, \gamma) = \min_{g | \gamma(g)=g} \|f - g\|.$$

Using the facts that the space of functions is an inner product space and that the functions that are invariant to reflection about  $\gamma$  define a vector subspace, it follows that the nearest invariant function  $g$  is precisely the projection of  $f$  onto the subspace of invariant functions. That is, if we define  $\pi_\gamma$  to be the projection onto the space of functions invariant under the action of  $\gamma$  and we define  $\pi_\gamma^\perp$  to be the projection onto the orthogonal subspace then:

$$SD(f, \gamma) = \|f - \pi_\gamma(f)\| = \|\pi_\gamma^\perp(f)\|$$

so that the symmetry distance of  $f$  with respect to  $\gamma$  is the length of the projection of  $f$  onto a subspace of functions indexed by  $\gamma$ .

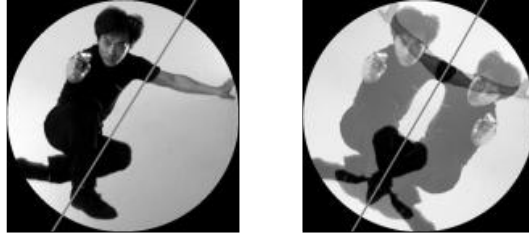
In order to compute an explicit formulation of the projection of  $f$  onto the space of functions invariant under the action of  $\gamma$ , we observe that reflections are orthogonal transformations (that is, they preserve the inner product defined on the space of functions). This lets us apply a theorem from representation theory [30] stating that a projection of a vector onto the subspace invariant under the action of an orthogonal group is the average

of the vector over the different elements in the group. Thus in the case of a function  $f$  and a reflection  $\gamma$  we get:

$$\text{SD}(f, \gamma) = \left\| f - \frac{1}{2}(f + \gamma(f)) \right\| = \left\| \frac{f - \gamma(f)}{2} \right\| \quad (1)$$

so that up to a scale factor the symmetry distance is simply the  $L^2$ -difference between the initial function and its reflection.

As an example, Figure 2 demonstrates this process of projection by averaging. The image on the left shows a picture of Jackie Chan. The image on the right is the closest image that is symmetric with respect to the gray line. It is obtained by averaging the original with its reflection. The  $L^2$ -difference between these two images is the measure of the symmetry of the initial image with respect to reflection about the gray line. Equivalently, according to Equation 1, the symmetry distance is half the  $L^2$ -distance from the original image to its reflection.



**Fig. 2.** An image of Jackie Chan (left) and its projection onto the space of images invariant under reflection through the gray line (right). The image on the right is obtained by averaging the image on the left with its reflection about the gray line.

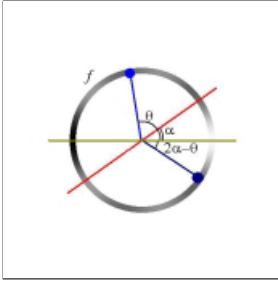
## 4 Computing the Reflective Symmetry Descriptor

The second issue is to build a shape descriptor for a 3D model based on symmetry distances. We define our *reflective symmetry descriptor* as a representation of the symmetry distances for all planes through the model's center of mass. This definition captures global symmetry information and can be parameterized on the domain of a sphere.

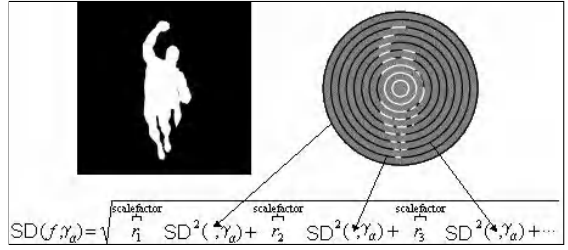
In order to use this reflective symmetry descriptor in practical applications, we must develop efficient algorithms for computing it. A naive algorithm would explicitly compute the symmetry distances in  $O(N^3)$  time for each of the  $O(N^2)$  planes in an  $N \times N \times N$  voxel model, requiring  $O(N^5)$  time overall. Our approach is to leverage the Fast Fourier Transform to reduce the computation time to  $O(N^4 \log N)$  and to use multiresolution methods to provide a good approximation in  $O(N^3 \log N)$ .

We present our method for efficiently computing the reflective symmetry descriptor of a voxel grid in four steps. First, we show how the Fast Fourier Transform can be used to compute the reflective symmetries of a function defined on a circle efficiently (Section 4.1). Second, we show how the case of a function  $f$  defined on the unit disk can be reduced to the case of a function on a circle by decomposing  $f$  into a collection





**Fig. 3.** Reflection about  $\alpha$  maps a point with angle  $\theta$  to the point with angle  $2\alpha - \theta$ .



**Fig. 4.** The reflective symmetry descriptor of a 2D image can be obtained by decomposing the image into concentric circles and computing the reflective symmetry descriptors on each of the circles.

of functions defined on concentric circles (Section 4.2). Third, through a collection of mappings we show how to reduce the question of finding the symmetry descriptor of a function on a sphere to the question of finding the symmetry descriptor for a function on a disk (Section 4.3). Fourth, we show how the reflective symmetry descriptor of a voxel grid can be computed by decomposing the grid into a collection of concentric spheres and applying the methods for symmetry detection of functions defined on a sphere (Section 4.4).

We conclude this section by describing a method for efficiently computing a multiresolution approximation to the descriptor, which provides comparable quality at low resolutions in far less time (Section 4.5).

#### 4.1 Functions on a Circle

In order to define the reflective symmetry descriptor for a function on a circle we would like to compute the symmetry distance for reflections about all lines through the origin efficiently. In particular, for a given function  $f$  on the circle and any reflection  $\gamma$  we would like to compute the measure of invariance of  $f$  with respect to  $\gamma$ . Denoting by  $\gamma_\alpha$  the reflection about the line through the origin with angle  $\alpha$  and using the fact that this reflection maps a point with angle  $\theta$  to the point with angle  $2\alpha - \theta$  (see Figure 3) we can apply Equation 1 to obtain:

$$SD(f, \gamma_\alpha) = \sqrt{\overbrace{\frac{\|f\|^2}{2}}^{L^2\text{-norm}} - \overbrace{\int_0^{2\pi} \frac{f(\theta)f(2\alpha - \theta)}{2} d\theta}^{\text{convolution term}}}$$

This formulation provides an efficient method for computing the reflective symmetry descriptor of a function defined on a unit circle because we can use the Fast Fourier Transform to compute the value of the convolution term for all angles  $\alpha$  in  $O(N \log(N))$  time, where  $N$  represents the number of times  $f$  is sampled on the circle.

## 4.2 Functions on a Disk

As with functions on a circle, the reflective symmetry descriptor of a function on a disk is a mapping that associates to every angle  $\alpha$  the measure of the invariance of the function with respect to the reflection about the line with angle  $\alpha$ . To compute the reflective symmetry descriptor we observe that these reflections fix circles of constant radius, and hence the symmetries of a function defined on a disk can be studied by looking at the restriction of the function to concentric circles. Figure 4 shows a visualization of this process where the image of Superman is decomposed into concentric circles and the reflective symmetry descriptor of the image is computed by combining the reflective symmetry descriptors of the different circular functions.

To make this observation explicit we reparameterize the function  $f(x, y)$  into polar coordinates to get the collection of functions  $\{\tilde{f}_r\}$  with:

$$\tilde{f}_r(\theta) = f(r \cos \theta, r \sin \theta),$$

where  $r \in [0, 1]$  and  $\theta \in [0, 2\pi]$ , and we set  $\gamma_\alpha$  to be the reflection about the line through the origin with angle  $\alpha$ . Using Equation 1 and applying the appropriate change of variables we get:

$$SD(f, \gamma_\alpha) = \sqrt{\int_0^1 SD^2(\tilde{f}_r, \gamma_\alpha) r dr}$$

showing that we can take advantage of the efficient method for computing the reflective symmetry descriptor of a function on the circle to obtain an  $O(N^2 \log N)$  algorithm for computing the reflective symmetry descriptor of an  $N \times N$  image.

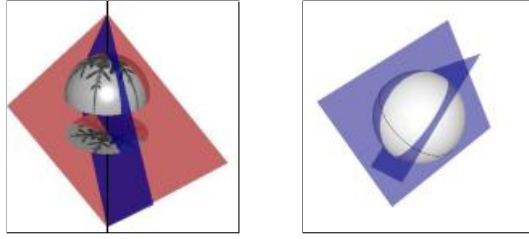
This method is similar to the method presented in the works of Marola and Sun et al. [8,10] in its use of autocorrelation as a tool for reflective symmetry detection. The advantage of our formulation is that it describes the relationship between autocorrelation and an explicit notion of symmetry distance, defined by the  $L^2$  inner-product of the underlying space of functions, and provides a method for generalizing the definition of symmetry distance to 3D.

## 4.3 Functions on a Sphere

The reflective symmetry descriptor of a function defined on the surface of a sphere is a mapping that gives the measure of reflective invariance of a model with respect to reflections about every plane through the origin. To compute the reflective symmetry descriptor of a function on a sphere we fix a North pole and restrict our attention to those planes passing through it. The values of the reflective symmetry descriptor for the restricted set of reflections can be efficiently computed by breaking up the function into its restrictions to the upper and lower hemisphere and projecting each of these restrictions to a disk. Figure 5(left) shows a visualization of this process for the restriction to the upper hemisphere. Note that reflections through planes containing the North pole map the upper hemisphere to itself and correspond to reflections about lines in the projected function.

In particular if we parameterize the sphere in terms of spherical coordinates:

$$\Phi(\phi, \theta) = (\cos \phi, \sin \phi \cos \theta, \sin \phi \sin \theta)$$



**Fig. 5.** To compute the reflective symmetry descriptor of a function defined on the sphere for planes passing through the North pole we observe that these planar reflections correspond to reflections about lines in the projected functions (left). We observe that a great circle must intersect every plane through the origin (right) so letting the North pole vary over a great circle and computing the projection at every step we obtain the symmetry distance for all planes.

with  $\phi \in [0, \pi]$  and  $\theta \in [0, 2\pi]$ , the restriction to the upper hemisphere corresponds to the restriction  $\phi \in [0, \pi/2]$ . Unfolding the restriction of  $f$  to the upper hemisphere along lines of constant latitude gives a function  $\tilde{f}_u$  defined on a disk of radius  $\pi/2$ :

$$\tilde{f}_u(\phi \cos \theta, \phi \sin \theta) = f(\Phi(\phi, \theta)) \sqrt{\frac{\sin \phi}{\phi}}.$$

We can obtain  $\tilde{f}_l$ , the projection of the lower hemisphere, in a similar fashion. Letting  $\gamma_\alpha$  represent both the reflection of the sphere about the plane through the North pole with constant angle of longitude  $\alpha$  and the reflection of the disk about the line with angle  $\alpha$  we get:

$$\text{SD}(f, \gamma_\alpha) = \sqrt{\text{SD}^2(\tilde{f}_u, \gamma_\alpha) + \text{SD}^2(\tilde{f}_l, \gamma_\alpha)}$$

so that with the correct parameterization and scaling of the projections, the symmetry distance for the reflection  $\gamma_\alpha$  can be obtained from the symmetry distances of the projections of  $f$ . (Note that rather than doing a true projection onto the plane perpendicular to the North pole, we actually unfold the hemisphere in terms of its angles of latitude. This allows us to avoid the sampling problems that would otherwise result due to a vanishing Jacobian near the boundary of the disk.)

In order to compute the value of the reflective symmetry descriptor for all planes through the origin, not just those passing through the North pole, we use the fact that if we fix a great circle on the sphere, any plane through the origin must intersect the great circle in at least two points (Figure 5(right)). This allows us to compute the values of the reflective symmetry descriptor for all planes by walking (half of) the great circle and at each point projecting onto a disk to compute the measure of symmetries for those planes containing the current North pole. Since the symmetry descriptor of the projection onto a disk can be computed in  $O(N^2 \log N)$  and since we preform  $O(N)$  such projections, this method gives an  $O(N^3 \log N)$  algorithm for computing the reflective symmetry descriptor of a function on the sphere, sampled at  $O(N^2)$  points.

#### 4.4 Functions on a Voxel Grid

As with a function defined on a sphere, the reflective symmetry descriptor of a voxel model is a function that gives the measure of invariance of the model with respect to reflection about every plane through the origin, where we assume that the model's center of mass has been translated to the origin. As in Section 4.2, we can use the fact that reflections fix lengths to transform the problem of computing the reflective symmetry descriptor of a voxel grid into a problem of computing the reflective symmetry descriptors of a collection of functions defined on a sphere. In particular, if  $f$  is a function defined on the set of points with radius less than or equal to 1 then we can decompose  $f$  into a collection of functions  $\{\tilde{f}_r\}$  where  $\tilde{f}_r$  is a function defined on the unit sphere and  $\tilde{f}_r(v) = f(rv)$ . After changing variables, the measure of symmetry of  $f$  with respect to a reflection  $\gamma$  becomes:

$$\text{SD}(f, \gamma) = \sqrt{\int_0^1 \text{SD}^2(\tilde{f}_r, \gamma) r^2 dr}$$

and we obtain the value of the symmetry descriptor of  $f$  as a combination of the values of the symmetry descriptors of the spherical functions  $\{\tilde{f}_r\}$ , giving a method for computing the reflective symmetry descriptor of an  $N \times N \times N$  model in  $O(N^4 \log N)$ .

#### 4.5 Multiresolution Approximation

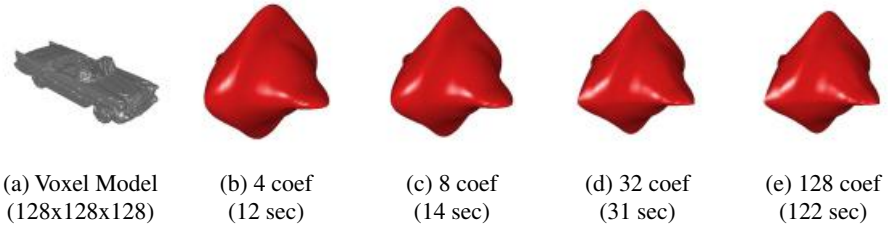
Our algorithm for computing the reflective symmetry descriptor takes  $O(N^4 \log N)$  time at full resolution. However, using Fourier decomposition of the restriction of the function to lines through the origin, we are able to compute a good multiresolution approximation to it in  $O(N^3 \log N)$  time. This approximation is useful in most applications because symmetry describes global features of a model and is apparent even at low resolutions. Given a function  $f$  defined on the set of points with radius less than or equal to 1 we decompose  $f$  into the collection of one-dimensional functions by fixing rays through the origin and considering the restriction of  $f$  to these rays. This gives a collection of functions  $\{\tilde{f}_v\}$ , indexed by unit vectors  $v$ , with  $\tilde{f}_v(t) = f(tv)$  and  $t \in [0, 1]$ . Expanding the functions  $\tilde{f}_v$  in terms of their trigonometric series we get:

$$\tilde{f}_v(t) = a_0(v) + \sum_{k=1}^{\infty} \left( a_k(v) \frac{\cos(2k\pi t)}{\sqrt{2}} + b_k(v) \frac{\sin(2k\pi t)}{\sqrt{2}} \right).$$

The advantage of this decomposition is that the functions  $a_k(v)$  and  $b_k(v)$  are functions defined on the sphere, providing a multiresolution description of the initial function  $f$ . Applying the appropriate change of variables and letting  $\gamma$  denote a reflection about a plane through the origin we get:

$$\text{SD}(f, \gamma) = \sqrt{\text{SD}^2(a_0, \gamma) + \sum_{k=1}^{\infty} (\text{SD}^2(a_k, \gamma) + \text{SD}^2(b_k, \gamma))}.$$

Thus a lower bound approximation to the reflective symmetry descriptor can be obtained in  $O(N^3 \log N)$  time by only using the first few of the functions  $a_k$  and  $b_k$ .



**Fig. 6.** (a) A 1957 Chevrolet model, (b-d) the approximations of its symmetry descriptor using the first 4, 8, and 32 spherical coefficient functions, and (e) the descriptor at full resolution.

The advantage of this multiresolution decomposition is that for binary voxel models we can show that the approximations converge quickly to the true value of the reflective symmetry descriptor at every point. In particular, if  $c$  represents the complexity of the model (i.e. the number of times a line through the origin will enter and exit the shape) then the approximation using only the first  $k$  Fourier coefficient functions differs from the true reflective symmetry descriptor in proportion to  $c/k$ . (This follows from the fact that if  $g(t)$  is the characteristic function of  $c$  disjoint segments contained on the interval  $[0, 2\pi]$  then  $\int \sin(2k\pi t)g(t)dt \leq c/k$ .) This result is demonstrated empirically in Figure 6(b-e), which shows the symmetry descriptor computed for a Chevrolet using 4, 8, 32, and 128 Fourier coefficient functions, respectively. Note that using only the first eight Fourier coefficient functions results in an approximation that is barely distinguishable from the higher resolution versions.

## 5 Additional Properties of the Reflective Symmetry Descriptor

In addition to being a function that is both parameterized over a canonical domain and describes a model in terms of its symmetries, the reflective symmetry descriptor has provable properties valuable for shape analysis:

**Stability:** The reflective symmetry descriptor is stable in the presence of high-frequency noise. To see this, we rewrite the reflective symmetry descriptor of a function  $f$ , defined on a circle, in terms of its Fourier coefficients:

$$SD(f, \gamma_\alpha) = \sqrt{\frac{1}{2} \sum_k (\|a_k\|^2 + a_k^2 e^{i2k\alpha})}.$$

This equation demonstrates that the contribution of different frequencies to the reflective symmetry descriptor depends only on their Fourier coefficients. In contrast, shape descriptors that involve computation of model derivatives, either as normals or gradients [25,10], have the property of amplifying the contribution of high-frequency components, making them unstable in the presence of high-frequency noise.

**Globality:** The differences in the reflective symmetry descriptors of two different models at a single point provides a lower bound for the overall similarity of the two models.

The proof of this bound derives from the fact that the symmetry distance of a function  $f$  with respect to a reflection  $\gamma$  is defined as the length of the projection  $\pi_\gamma^\perp(f)$  (Section 3). Since we know that for any orthogonal projection  $\pi$  and any vectors  $v$  and  $w$  we have  $\|v\| \geq \|\pi(v)\|$  and  $\|v - w\| \geq \|\pi(v) - \pi(w)\|$  it follows that:

$$\|f - g\| \geq \|\pi_\gamma^\perp(f) - \pi_\gamma^\perp(g)\| = |\text{SD}(f, \gamma) - \text{SD}(g, \gamma)|$$

so that the difference in the values of two symmetry descriptors at a single point provides a lower bound for the  $L^2$ -difference of the corresponding models.

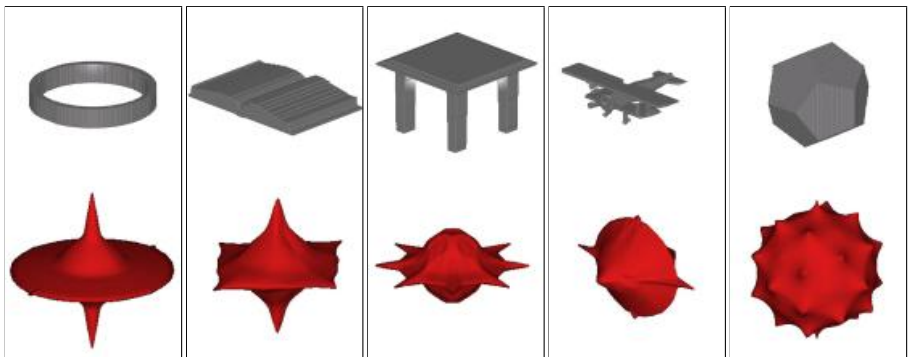
## 6 Results

In this section, we show the reflective symmetry descriptors of a wide variety of models and demonstrate the efficacy of the descriptor as a shape analysis tool by showing how models can be registered and classified using the descriptor.

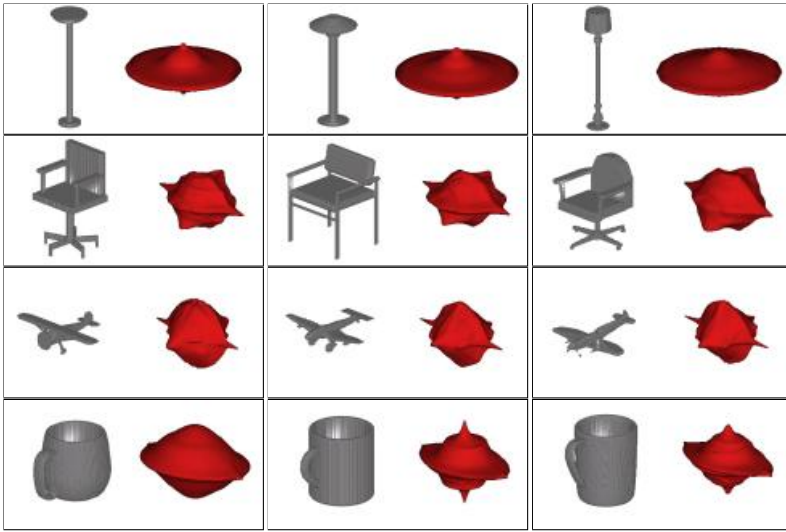
### 6.1 Test Database

Our test database consisted of 90 polygonal models categorized by a third party into 24 different classes. Using a simple rasterization method, the interior of each model was voxelized into a  $128 \times 128 \times 128$  grid.

Figures 7 and 8 show a number of models from the test database with their corresponding reflective symmetry descriptors. The full symmetry descriptor for each model was computed in 122 seconds on an 800 MHz Athlon processor with 512 MB of RAM. Note that the descriptors vary from model to model, with different patterns of undulations and sharp peaks, demonstrating that the symmetry descriptor is a rich function, capable of describing large amounts of information about shape.



**Fig. 7.** A number of models from different classes with their reflective symmetry descriptors, demonstrating the variability and richness of the descriptor.



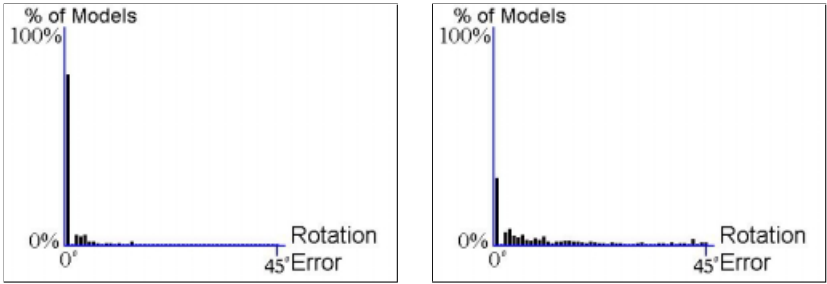
**Fig. 8.** A number of representative models with their corresponding symmetry descriptors. Note that the descriptor remains consistent within a class.

## 6.2 Registration Results

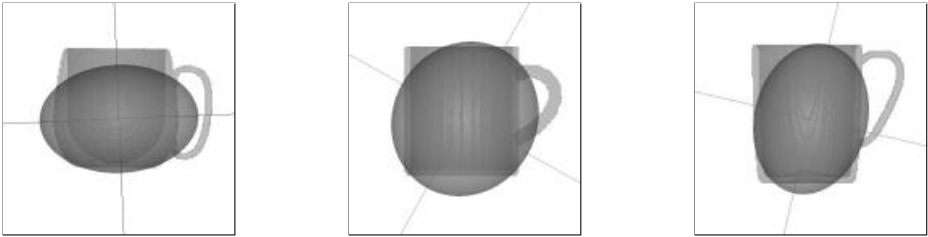
In this section, we test how well the symmetry descriptors of models can be used for registration. To do this we hand-aligned all pairs of models within a class. For each pair of models, we computed the axis of the rotation which brings the two models into alignment and then searched for the rotation along that axis that minimizes the  $L^2$ -distance between the corresponding symmetry descriptors.

Figure 9 compares errors in registration using the reflective symmetry descriptor with errors using principal axis alignment. The graphs show the percentage of pairwise registrations that resulted in a given error, where error is measured as the difference in the rotation angles between the user-specified and the computed rotations. Note that reflective symmetry descriptors register different models within the same class to within 5 degrees of what a human would do in 95% of the tests as opposed to the covariance approach that only registers to within 5 degrees 55% of the time.

These results indicate that registration using the reflective symmetry descriptor does a better job of aligning models than the classic principal axis method. We believe that the trouble with using principal axis alignment is two-fold. First, in the case that an eigenspace of the covariance matrix is more than one-dimensional, a unique eigenvector cannot be determined. Second, the contribution of points to the covariance matrix scales quadratically with their distance from the center, so that small changes in a model that occur far from the center can drastically change the principal axes. Figure 10 demonstrates this by showing the principal axes of three different mugs. Note that changes in the position and shape of the handle, and changes in the cylindrical nature of the mug give rise to principal axes that are differently aligned. The reflective symmetry descriptor, by contrast, remains stable throughout these variances (Figure 8 bottom row.)



**Fig. 9.** A comparison of registration using the reflective symmetry descriptor (left) with registration using principal axes (right). The graphs show the percentage of pairwise registration that resulted in a given rotation error.



**Fig. 10.** A collection of mugs with their principal axes. The figure demonstrates that minor variances within a class can drastically affect the orientation of the principal axes.

6.3 Classification Results

We also evaluated the discriminating power of the reflective symmetry descriptor with respect to the task of object classification. In order to do this efficiently, we generated rotation-invariant signatures for the symmetry descriptor based on the distribution of their values. In particular, we obtained the first eight approximating Fourier coefficient functions (as describes in Section 4.5) and generated histograms for each of their reflective symmetry descriptors. The histograms contained 100 bins, with the  $k$ -th bin containing the measure of the points on the sphere whose corresponding symmetry distance was in the range  $[k/100, (k + 1)/100]$ . We measured model similarity by comparing the obtained histograms using the Earth Mover’s Distance [31].

We performed a sequence of leave-one-out classification experiments for each model based on the measure of model similarity. Table 1 compares the results of model classification using the symmetry distribution with the classification results obtained using higher order moments [14] and shape distributions [15], two other global shape descriptors used for matching and classification. In order to provide a base measure of performance, the table also presents the results when the similarity measure returns a random number. The quality of the classification was measured using three metrics [15]. The *Nearest Neighbor* value is the percentage of models whose closest match belonged to the same class. The *First Tier* and *First Two Tiers* values corresponds to the percentage



of models in the first  $(n-1)$  and  $2(n-1)$  nearest matches that belonged to the same class as the query model, where  $n$  is the class size. They provide measures of classification that normalize for the number of models within a class. Note that reflective symmetry descriptors classify models as well or better than the other shape descriptors for all three classification criteria.

**Table 1.** Comparison of results of the model matching experiment using 4th and 7th order moments, shape distributions and symmetry descriptors.

Comparison Method	Nearest Neighbor	First Tier	First Two Tiers	Time
Random	6%	4%	9%	
Moments (4th Order)	34%	40%	48%	0.1 seconds
Moments (7th Order)	24%	33%	38%	0.25 seconds
Shape Distributions	44%	64%	62%	0.35 seconds
Symmetry Descriptors	52%	69%	71%	.15 seconds

## 7 Conclusion and Future Work

In this paper, we have introduced the *reflective symmetry descriptor*, a function associating a measure of reflective invariance of a 3D model with respect to every plane through the center of mass. It has several desirable properties, including invariance to translation and scale, parameterization over a canonical domain, stability, and globality that make it useful for registration and classification of 3D models. We have shown how to compute it efficiently, and conducted preliminary experiments that show its usefulness for shape registration and classification.

This work suggests a number of questions that we would like to address in future research: (1) Can the symmetry descriptor be used for other shape analysis tasks, such as learning a statistical classifier of shape? (2) Can the multiresolution properties of the descriptor be used to develop more efficient search algorithms, e.g., for registration and recognition? (3) Can other theoretical properties of the descriptor be proven, such as showing when 3D models can have the same descriptor? Answers to these questions will further our understanding of how symmetry defines shape.

## References

1. Mitsumoto, H., Tamura, S., Okazaki, K., Kajimi, N., Fukui, Y.: Reconstruction using mirror images based on a plane symmetry recovery method (1992)
2. Zabrodsky, H., Peleg, S., Avnir, D.: Symmetry as a continuous feature. *IEEE PAMI* **17** (1995) 1154–1156
3. Liu, Y., Rothfus, W., Kanade, T.: Content-based 3d neuroradiologic image retrieval: Preliminary results (1998)
4. Leou, J., Tsai, W.: Automatic rotational symmetry determination for shape analysis. *Pattern Recognition* **20** (1987) 571–582

5. Wolfson, H., Reisfeld, D., Yeshurun, Y.: Robust facial feature detection using symmetry. *Proceedings of the International Conference on Pattern Recognition* (1992) 117–120
6. Atallah, M.J.: On symmetry detection. *IEEE Trans. on Computers* **c-34** (1985) 663–666
7. Wolter, J.D., Woo, T.C., Volz, R.A.: Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer* **1** (1985) 37–48
8. Marola, G.: On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE PAMI* **11** (1989) 104–108
9. Shen, D., Ip, H., Cheung, K., Teoh, E.: Symmetry detection by generalized complex (gc) moments: A close-form solution. *IEEE PAMI* **21** (1999) 466–476
10. Sun, C., Si, D.: Fast reflectional symmetry detection using orientation histograms. *Real-Time Imaging* **5** (1999) 63–74
11. O'Mara, D., Owens, R.: Measuring bilateral symmetry in digital images. *IEEE-TENCON - Digital Signal Processing Applications* (1996)
12. Sun, C., Sherrah, J.: 3-d symmetry detection using the extended Gaussian image. *IEEE PAMI* **19** (1997)
13. Kovese, P.: Symmetry and asymmetry from local phase. *Tenth Australian Joint Conference on Artificial Intelligence* (1997) 2–4
14. Elad, M., Tal, A., Ar, S.: Directed search in a 3d objects database using svm (2000)
15. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3d models with shape distributions. *Shape Matching International* (2001)
16. Knuth, D., J.H. Morris, J., Pratt, V.: Fast pattern matching in strings. *SIAM Journal of Computing* **6** (1977) 323–350
17. Besl, P.J., Jain, R.C.: Three-dimensional object recognition. *Computing Surveys* **17** (1985) 75–145
18. Loncaric, S.: A survey of shape analysis techniques. *Pattern Recognition* **31** (1998) 983–1001
19. Pope, A.R.: Model-based object recognition: A survey of recent research. Technical Report TR-94-04, University of British Columbia (1994)
20. Veltkamp, R.C., Hagedoorn, M.: State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands (1999)
21. Johnson, A., Hebert, M.: Efficient multiple model recognition in cluttered 3-d scenes. *IEEE CVPR* (1998) 671–677
22. Zhang, D., Hebert, M.: Harmonic maps and their applications in surface matching. *IEEE CVPR* **2** (1999)
23. Belongie, S., Malik, J.: Matching with shape contexts. *IEEE Workshop on Content-based access of Image and Video-Libraries* (2000)
24. Mori, G., Belongie, S., Malik, H.: Shape contexts enable efficient retrieval of similar shapes. *CVPR* **1** (2001) 723–730
25. B. Horn, B.: Extended gaussian images. *PIEEE* **72** (1984) 1656–1678
26. Fernández-Vidal, S., Bardin, E., Malandain, G., Damas, S., de la Blanca Capilla, N.: Object representation and comparison inferred from its medial axis. *ICPR* **1** (2000) 712–715
27. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. *IJCV* **35** (1999) 13–32
28. Bloomenthal, J., Lim, C.: Skeletal methods of shape manipulation. *Shape Modeling and Applications* (1999) 44–47
29. Storti, D., Turkiyyah, G., Ganter, M., Lim, C., Stal, D.: Skeleton-based modeling operations on solids. *Symposium on Solid Modeling and Applications* (1997) 141–154
30. Serre, J.: *Linear Representations of Finite Groups*. Springer-Verlag, New York (1977)
31. Rubner, Y., Tomasi, C., Guibas, L.: A metric for distributions with applications to image databases. *IEEE ICCV* (1998)

# Gait Sequence Analysis Using Frieze Patterns<sup>\*</sup>

Yanxi Liu, Robert Collins, and Yanghai Tsin

The Robotics Institute,  
Carnegie Mellon University  
{yanxi,rcollins,ytsin}@cs.cmu.edu  
<http://www.cs.cmu.edu/~yanxi/www/home.html>

**Abstract.** We analyze walking people using a gait sequence representation that bypasses the need for frame-to-frame tracking of body parts. The gait representation maps a video sequence of silhouettes into a pair of two-dimensional spatio-temporal patterns that are near-periodic along the time axis. Mathematically, such patterns are called “frieze” patterns and associated symmetry groups “frieze groups”. With the help of a walking humanoid avatar, we explore variation in gait frieze patterns with respect to viewing angle, and find that the frieze groups of the gait patterns and their canonical tiles enable us to estimate viewing direction of human walking videos. In addition, analysis of periodic patterns allows us to determine the dynamic time warping and affine scaling that aligns two gait sequences from similar viewpoints. We also show how gait alignment can be used to perform human identification and model-based body part segmentation.

## 1 Motivation

Automated visual measurement of human body size and pose is difficult due to nonrigid articulation and occlusion of body parts from many viewpoints. The problem is simplified during gait analysis, since we observe people performing the same activity with certain time period. Although individual gaits vary due to factors such as physical build, body weight, shoe heel height, clothing and the emotional state of the walker, at a coarse level the basic pattern of bipedal motion is the same across healthy adults, and each person’s body passes through the same sequence of canonical poses while walking [6]. We have experimented with a simple, viewpoint-specific spatio-temporal representation of gait. The representation collapses a temporal sequence of body silhouette images into a periodic two-dimensional pattern. This paper explores the use of these frieze patterns for viewing angle determination, human identification, and non-rigid gait sequence alignment.

## 2 Related Work

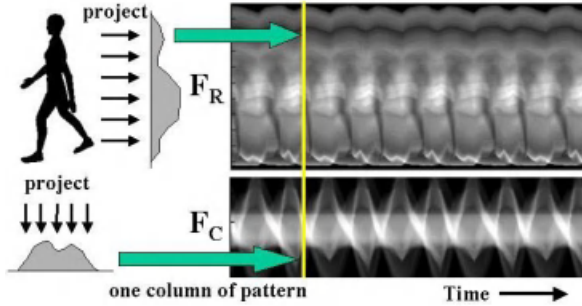
Many approaches to analyzing gait sequences are based on tracking the body as a kinematic linkage. Model-based kinematic tracking of a walking person was

---

<sup>\*</sup> This work is supported in part by ONR N00014-00-1-0915 and NSF # IIS-0099597.

pioneered by Hogg [11], and other influential approaches in this area are [2,4]. These approaches are often brittle, since the human body has many degrees of freedom that cannot be observed well in a 2D image sequence. Our work is more closely related to approaches based on pattern analysis of spatio-temporal representations. Niyogi and Adelson delineate a person's limbs by fitting deformable contours to patterns that emerge from taking spatio-temporal slices of the XYT volume formed from an image sequence [17]. Little and Boyd analyze temporal signals computed from optic flow to determine human identity from gait [14]. Analyzing features over a whole temporal sequence is a powerful method for overcoming noise in individual frames.

Liu and Picard [15] propose to detect periodic motions by treating temporal changes of individual pixels as 1D signals whose frequencies can be extracted. Seitz and Dyer [18] replace the concept of period by the instantaneous period, the duration from the current time instant at which the same pattern reappears. Their representation is effective in studying varying speed cyclic motions and detecting irregularities. Cutler and Davis [5] also measure self-similarity over time to form an evolving 2D pattern. Time-frequency analysis of this pattern summarizes interesting properties of the motion, such as object class and number of objects.



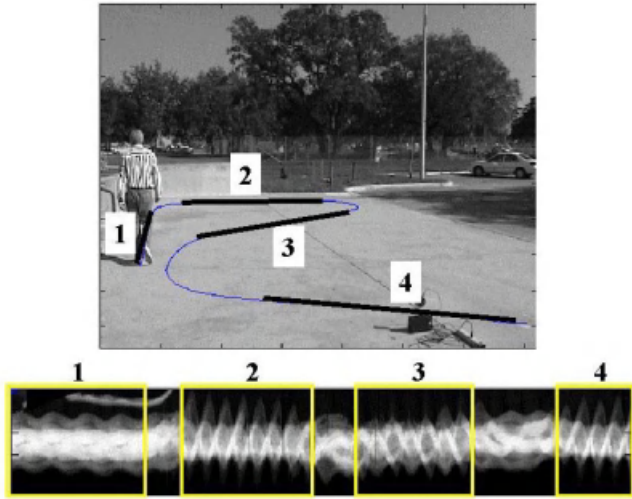
**Fig. 1.** Spatio-temporal gait representations are generated by projecting the body silhouette along its columns ( $F_C$ ) and rows ( $F_R$ ), then stacking these 1D projections over time to form 2D patterns that are periodic along the time dimension. A 2D pattern that repeats along one dimension is called a “frieze” pattern.

### 3 A Spatio-Temporal Gait Representation

Consider a sequence of binary silhouette images  $b(t) \equiv b(x, y, t)$ , indexed spatially by pixel location  $(x, y)$  and temporally by time  $t$ . Form a new 2D image  $F_C(x, t) = \sum_y b(x, y, t)$ , where each column (indexed by time  $t$ ) is the vertical projection (column sum) of silhouette image  $b(t)$ , as shown in Figure 1. Each value  $F_C(x, t)$  is then a count of the number of silhouette pixels that are “on” in

column  $x$  of silhouette image  $b(t)$ . The result is a 2D pattern, formed by stacking column projections together to form a spatio-temporal pattern. A second pattern  $F_R(y, t) = \sum_x b(x, y, t)$  can be constructed by stacking row projections. Since a human gait is periodic with respect to time,  $F_C$  and  $F_R$  are also periodic along the time dimension. A two-dimensional pattern that repeats along one dimension is called a *frieze* pattern in the mathematics and geometry literature, and group theory provides a powerful tool for analyzing such patterns (Section 4.1).

Figure 2 shows the column projection frieze pattern  $F_C$  extracted from a roughly 30 second long sequence of a person walking along a test course. Note the changes in appearance of the frieze pattern as the walking direction changes. In our experiments, body silhouette extraction is achieved by simple background subtraction and thresholding, followed by a 3x3 median filter operator to suppress spurious pixel values. Silhouettes across a gait sequence are automatically aligned by scaling and cropping based on bounding box measurements so that each silhouette is 80 pixels tall, centered within a window 80 pixels wide by 128 pixels high. Background subtraction is a commonly used method for extract-



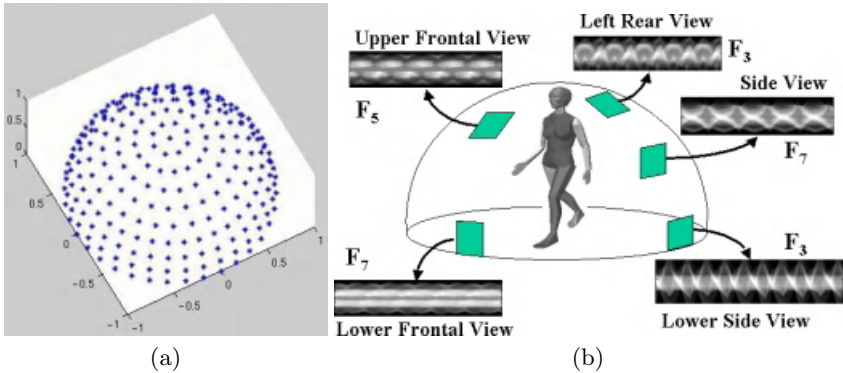
**Fig. 2.** Frieze pattern extracted from a 30 second long walking sequence. Note the changes in appearance of the frieze pattern as the walking direction changes.

ing body silhouettes from a stationary background scene [7,20]. It is difficult for kinematic trackers to automatically identify and fit individual limb positions from such data. This is because background subtraction often yields noisy silhouettes with holes, fragmented boundaries, and extra parts due to background clutter and shadows. However, by distilling a sequence of silhouettes into a periodic pattern that can be smoothed and analyzed using robust signal analysis

techniques, our “holistic” approach to gait analysis is better able to deal with noisy silhouette data.

## 4 Model-Based Gait Analysis

With the aid of a 3D walking humanoid model, we have studied how the spatio-temporal frieze patterns described above vary with respect to camera viewpoint. Our model of human body shape and walking motion is encapsulated in a VRML/H-Anim 1.1 compliant avatar called “Nancy”.<sup>1</sup> Nancy’s 3D polyhedral body parts were generated by a graphics designer, and the gait motion, specified by temporal sequences of interpolated rotations at each joint, is based on the motion sequence of a real person in “The Human Figure in Motion” by Eadweard Muybridge. We have ported Nancy into an open-GL program that generates 2D perspective views of the avatar given a camera position and time step within the gait cycle. Figure 3 illustrates variation of the column projection frieze patterns  $F_C$  defined in Section 3 when Nancy’s gait is seen from different viewing directions. The diversity inspires us to seek an encoding for these different types of frieze patterns in order to determine viewpoint from frieze group type. One candidate for categorizing frieze patterns is by their symmetry groups.



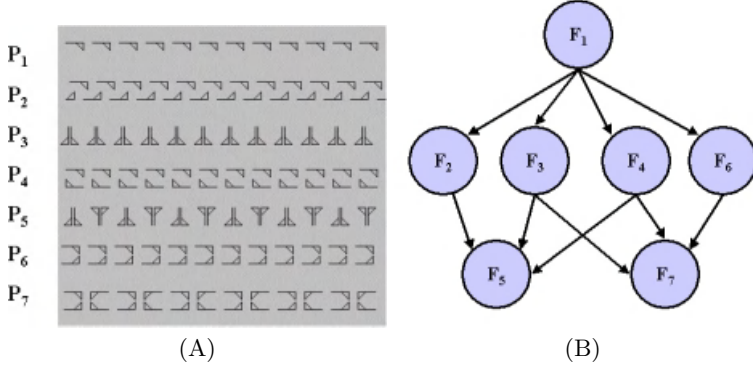
**Fig. 3.** (a) A database of gait sequences is generated from 241 sample viewpoints. The subject is a walking humanoid avatar (motion sequence is from a real person). (b) Some gait patterns of the avatar “Nancy” viewed from different directions.

### 4.1 Frieze Symmetry Groups Classification

Any frieze pattern  $P_i$  in Euclidean space  $R^2$  is associated with a unique symmetry group  $F_i$ , where  $i = 1..7, \forall g \in F_i, g(P_i) = P_i$ . These seven symmetry groups are

<sup>1</sup> ©1997 Cindy Ballreich, 3Name3D / Yglesias, Wallock, Divekar, Inc.

called *frieze groups*, and their properties are summarized in Figure 4 and Table 1. Five different types of symmetries can exist for frieze patterns: (1) translation, (2) 2-fold rotation, (3) horizontal reflection (4) vertical reflection, and (5) glide-reflection. A frieze pattern can be classified into one of the 7 frieze groups based on what combination of these 5 primitive symmetries are present in the pattern [16] (Table 1).



**Fig. 4.** (A) The seven frieze patterns ( $P_1...P_7$ ) in Euclidean space  $R^2$ . (B) The subgroup relationship among the seven frieze symmetry groups ( $F_1...F_7$  in Table 1).  $F_i \rightarrow F_j$  means  $F_i$  is a subgroup of  $F_j$ .

We are interested in classifying imperfect and noise-contaminated frieze patterns generated from real human gaits. There are two important and intertwined computational issues for frieze symmetry group classification: 1) given an imperfect frieze pattern, how to decide whether or not it has certain types of symmetries; and 2) given the symmetry measures for a pattern, how to give each of the seven frieze groups an equal chance to be chosen as the symmetry group of the pattern, since these groups are *not disjoint*. The first issue is addressed by establishing a distance measure between an imperfect periodic pattern and frieze patterns. The second issue is addressed by using geometric AIC [12,13] for symmetry group model selection.

**Distance to the Nearest Frieze Patterns.** We define the symmetry distance (SD) of an approximately periodic pattern  $P$  to the set of all frieze patterns  $\{P_n\}$  with frieze group  $F_n$  as

$$SD_n(P) = \min_{Q \in \{P_n\}} \left\{ \sum_{i=1}^{tN} \left( \frac{p_i - q_i}{s_i} \right)^2 \right\} \quad (1)$$

where  $N$  is the number of pixels in a tile (smallest 2D repeating region),  $t$  is the number of tiles being studied,  $p_i$  and  $q_i$  are intensity values of corresponding

**Table 1.** Symmetries of frieze pattern tiles (N is number of pixels in one tile)

Symmetry Group	translation	2-fold rotation	Horizontal reflection	Vertical reflection	Glide reflection	Degrees of Freedom
F1	yes	no	no	no	no	N
F2	yes	no	no	no	yes	N/2
F3	yes	no	no	yes	no	N/2
F4	yes	yes	no	no	no	N/2
F5	yes	yes	no	yes	yes	N/4
F6	yes	no	yes	no	no	N/2
F7	yes	yes	yes	yes	no	N/4

pixels of pattern  $P$  and  $Q \in \{P_n\}$  respectively, and  $s_i$  is the standard deviation of the frieze pattern at pixel  $i$ . For independent Gaussian noise, the distance  $SD_n$  has a  $\chi^2$  distribution with  $tN$  degrees of freedom.

The symmetry distance measure is defined with respect to a frieze pattern  $Q \in \{P_n\}$  that has the minimal distance to  $P$ . We can show that this pattern  $Q$  can be constructed as follows: (1) For  $t > 1$  and  $n = 1$ ,  $Q$  is the pixel-wise average of all the tiles in  $P$ . (2) For  $t = 1$  and  $n > 1$ ,  $Q = \frac{(\mathcal{O}_n(P)+P)}{2}$ , where  $\mathcal{O}_n(P)$  is the pattern obtained by applying the set of symmetry operations in  $F_n$  to  $P$ . (3) For  $t > 1$  and  $n > 1$ ,  $Q$  is the pixel-wise average of each  $Q$  obtained above. Our definition of frieze pattern symmetry distance in pixel intensity space is analogous to that of Zabrodsky et.al. [21,13] for polygon distance in vertex location space.

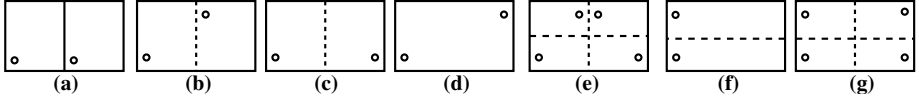
**Geometric AIC for Frieze Group classification.** The frieze symmetry groups form a hierarchical structure (Figure 4B) where frieze group  $F_1$  is a subgroup of all the other groups and so on. For example, a frieze pattern  $P_3$  (with vertical reflection symmetry) is a more general pattern type than  $P_5$  or  $P_7$ , since any  $P_5$  or  $P_7$  frieze with more complicated symmetries also has vertical reflection symmetry. But this implies that the distance of a pattern  $P$  to  $P_3$  is always no greater than the distance to  $P_5$ , since the set of  $P_5$  patterns is a subset of the  $P_3$  patterns. If no care is taken, a symmetry group classification algorithm based on raw symmetry distance scores will always favor  $P_3$  over  $P_5$ . To address this problem, we adopt the concept of Geometric-AIC (G-AIC) proposed by Kanatani [12,13]. Given two possible frieze patterns whose symmetry groups have a subgroup relationship, G-AIC states that we should prefer  $F_m$  over  $F_n$  if

$$\frac{SD_m}{SD_n} < 1 + \frac{2(d_n - d_m)}{r(tN) - d_n} \tag{2}$$

where  $d_m$  and  $d_n$  are the degrees of freedom for frieze patterns of  $F_m$  and  $F_n$  respectively, and  $r$  is the codimension. Since the data space (the intensity space) is



dimension one, and our model space (point in multidimensional intensity space) dimension is 0, the codimension  $r = 1 - 0 = 1$ .



**Fig. 5.** Determining the degrees of freedom of frieze patterns by how many constraints a pixel intensity has to satisfy. The figure shows the corresponding pixels that must have the same intensity values in (a) two tiles of a  $P_1$  pattern; (b)-(g) a tile from frieze pattern  $P_2 \dots P_7$  respectively.

The degrees of freedom (DOF) of a frieze pattern depends on how the intensity of each pixel on the pattern is constrained. For frieze patterns with translation symmetry only, the only constraint for each of the  $tN$  pixels is to have the same intensity value as the pixel  $t$  units to the left. Thus its DOF is  $N$ . On the other hand, pixels on a  $P_3$  pattern have to satisfy a vertical reflection symmetry constraint, and thus half of the pixel intensities need to be the same as the other half. So the DOF of a  $P_3$  pattern is  $N/2$ . The last column of Table 1 and Figure 5 explain the DOFs of the seven frieze groups. In summary, we would prefer to classify a pattern  $P$  as having frieze group  $F_m$  rather than  $F_n$  if

$$\frac{SD_m(P)}{SD_n(P)} < \frac{t}{t-1}, \text{ for } m = 2, 3, 4, 6 \text{ and } n = 1 \quad (3)$$

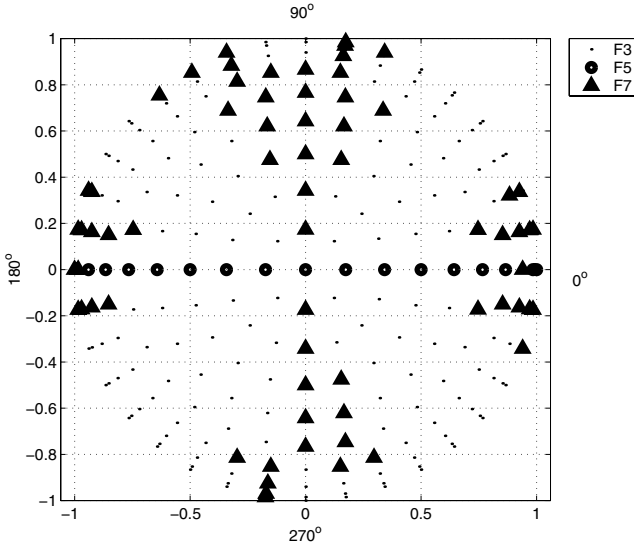
$$\frac{SD_m(P)}{SD_n(P)} < \frac{2t}{2t-1}, \text{ for } m = 5, 7 \text{ and } n = 2, 3, 4, 6 \quad (4)$$

$$\frac{SD_m(P)}{SD_n(P)} < \frac{2t+1}{2t-2}, \text{ for } m = 5, 7 \text{ and } n = 1 \quad (5)$$

## 4.2 View Direction Estimation

To study the effects of viewpoint on human gait appearance we have generated a database of 241 walk sequences, indexed by viewing direction azimuth and elevation, by sampling the view sphere at roughly every 10 degrees (Figure 3). This type of detailed study is perhaps only possible given a generative model, since the cost of gathering such data experimentally would be prohibitive. Figure 6 shows the frieze groups associated with the 241 column projection frieze patterns  $F_C$  generated from Nancy's gait when viewed from these directions. Row projection frieze patterns  $F_R$  exhibit less variation with respect to group structure, and are not used in this section.

The underlying assumption in the current approach is that the distribution of symmetry groups of the gait patterns from different views of the computer model can provide guidance for determining the viewing angle of an observed human subject's gait. Figure 7 shows a comparison among corresponding avatar and two

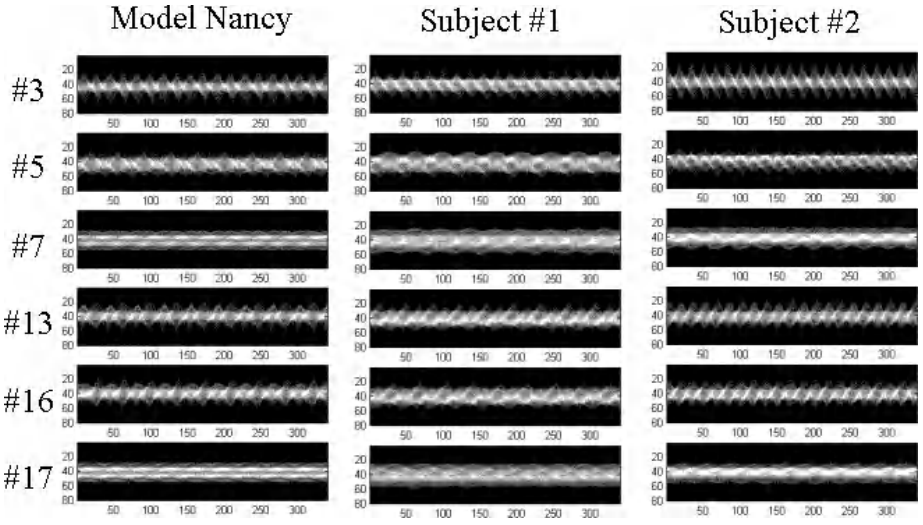


**Fig. 6.** Symmetry group labelings of the frieze patterns of a humanoid avatar viewed from different directions on the upper hemisphere (Figure 3). Each sample point on the hemisphere is projected to the plane of elevation 0 degrees. Since the shape and motion of the avatar is based on a real person, the symmetry group distribution map above is NOT perfectly symmetrical with respect to the  $0^\circ$  (Frontal view) and  $180^\circ$  (Back view) line.

individual human subject (one male, one female) frieze patterns, viewed from six different viewing angles. One can observe that frieze patterns from the same view point share the same frieze symmetry group, and their tiles have a similar appearance. We also observe that the avatar frieze patterns are visually similar to the patterns extracted from gait video of real human subjects. In another word, the gait frieze patterns so computed are more similar across different subjects than across viewing directions.

Given an observed human subject gait pattern  $P = F_C$  (Section 3), we use a moment-based method (Section 5.1) to align the model friezes  $P_i$  from each of the 241 candidate viewing directions to the subject frieze. Applying PCA to a typical tile from  $P$  and taking the non-dominant PCA components that are most sensitive to discriminate pattern variations, the closest  $K$  nearest neighbors are found in this subspace. We used a dual elimination method to decide which angle values from these  $K$  neighbors we can count on. The first condition is that  $P$  and  $P_i$  have the same symmetry group. The second condition is that corresponding pixels of tiles from  $P$  and  $P_i$  must have similar intensities. Results for classifying viewing direction for two human subjects is listed in Table 2.

In this framework we have assumed affine camera models, thus only one viewing direction (specified by azimuth and elevation) is estimated for a ray directed towards the center of the person. However, the data used in the experiment



**Fig. 7.** View of  $F_C$  frieze patterns from six different angles. Left: Avatar Nancy (motion sequence is extracted from a real female person). Middle: human subject # 1 (male). Right: human subject # 2 (female).

**Table 2.** View direction estimation using frieze groups, for two real human subjects viewed from six different cameras

Camera ID	Sym. Group	Ground truth			subj 1 estimate		subj 2 estimate	
		View Dir.	elevation	azimuth	elev	azim	elev	azim
3	F7	L. side	15.4	83.2	50	75	10	80
5	F3	L. front	12.0	37.4	30	160	80	45
7	F5	Frontal	25.0	359.8	20	0	20	0
13	F3	R. back	11.4	234.9	20	200	20	240
16	F3	R. front	11.9	314.5	40	334	10	20
17	F5	Back	26.5	181.4	20	180	20	180

comes from a perspective camera, and due to the proximity of the subject, the difference in viewing ray elevation between their head and feet is roughly 28 degrees. This partly explains why estimation of azimuth angles tends to be more accurate than elevation angles. Furthermore, much more accurate estimations of viewing angles are achieved for frieze patterns with non- $F_3$  groups (Table 2). This can be explained by the multiple possible angle ranges for  $P_3$  patterns (Figure 6). We have dealt with this using majority votes and robust median estimators. Although having the same symmetry group is only a *necessary* condition for two gait patterns to share the same viewing direction, our initial experiments show that this condition yields better estimation accuracy than not using symmetry groups at all. Accuracy would be further improved by using these cues in combination with other constraints, for example geometric constraints on azimuth

and elevation provided by approximate knowledge of ground plane and direction of travel [19].

## 5 Spatio-Temporal Gait Alignment

Consider two gait sequences, represented by two pairs of frieze patterns  $F_C(x, t)$ ,  $F_R(y, t)$  and  $F'_C(x', t')$ ,  $F'_R(y', t')$ . We seek to align the patterns temporally and spatially, as a precursor for further correspondence-based analysis. Temporal alignment of gait sequences amounts to aligning frieze patterns horizontally, thereby determining a mapping between time variables  $t$  and  $t'$ . Spatial alignment means finding a mapping between pixel locations  $(x, y)$  in sequence 1 and  $(x', y')$  in sequence 2. We restrict this to a four parameter affine mapping, and show that it can be found by aligning the corresponding row and column friezes along their vertical dimensions.

Spatio-temporal alignment of two video sequences is typically treated within a framework of 3D space-time volumetric warping [3]. However, representing human activity using line projection frieze patterns collapses the problem down to 2D spatial pattern alignment. Temporal alignment of these frieze patterns is further simplified by the periodic nature of the patterns themselves, allowing us to use simple periodic signal analysis in place of expensive dynamic time warping procedures [8].

### 5.1 Moment-Based Gait Alignment

It is well known that the first and second moments of two binary silhouettes can be used to determine an affine transformation that coarsely aligns them, and that some of the moments of a silhouette image can be computed from its row and column projections [1]. This forms the basis of our gait alignment method.

First, we generalize the concept of moments of a binary image to cover a time series of moments computed from a sequence of binary images. Define a *moment sequence* as  $m_{ij}(t) = \sum_x \sum_y x^i y^j b(x, y, t)$ , which is a sequence of single-frame binary silhouette moments, indexed by time. Note that  $m_{00}(t)$  is just the area of the binary silhouette over time, while  $\bar{x}(t) \equiv m_{10}(t)/m_{00}(t)$  and  $\bar{y}(t) \equiv m_{01}(t)/m_{00}(t)$  are the coordinates of the silhouette centroid over time. Similarly, define a *central moment sequence* as  $\mu_{ij}(t) = \sum_x \sum_y (x - \bar{x}(t))^i (y - \bar{y}(t))^j b(x, y, t)$ , which is a sequence of moments measured after translating each silhouette so that its centroid is at the origin. The second central moments measure the spread of silhouette pixels about the centroid, and can be used to derive the principal axis of the silhouette shape.

Since we are summarizing each sequence of silhouettes with frieze patterns, we are concerned only with moments that can be computed from row and column projections. For example, consider silhouette area

$$m_{00}(t) = \sum_x \sum_y b(x, y, t) = \sum_x \left( \sum_y b(x, y, t) \right) = \sum_x F_C(x, t)$$

which can thus be computed from the frieze pattern as well as the original silhouette sequence. Any moment sequence  $m_{ij}(t)$  or central moment sequence  $\mu_{ij}(t)$  with either  $i$  or  $j$  (or both) equal to zero can be computed from frieze patterns  $F_C(t)$  and  $F_R(t)$ . In the present case, we will use  $m_{00}(t)$ ,  $m_{10}(t)$ ,  $m_{01}(t)$ ,  $\mu_{20}(t)$ , and  $\mu_{02}(t)$ . Note that the second central moment  $\mu_{11}(t)$  can not be determined from the two frieze patterns, and we will therefore not be able to adjust skew or principle axis rotation when aligning silhouette shapes using friezes alone.

We now present an algorithm for moment-based gait alignment. To a first approximation, the temporal alignment between the two periodic gait sequences can be represented as  $t' = \rho t + \phi$ , where  $\rho$  corrects for the relative stride frequency and  $\phi$  corrects for the relative phase difference (position within a stride). The average stride frequency of each gait sequence is found by taking signal  $m_{00}(t)$ , “whitening” it by subtracting its mean and dividing by its standard deviation, then autocorrelating to find peaks occurring at a fundamental frequency. From some viewpoints this is the stride frequency, and from others it is half the stride frequency (e.g. a bipedal gait viewed from the side looks self-similar halfway through a full stride). Whether the autocorrelation of  $m_{00}(t)$  yields peaks at half the stride frequency is viewpoint dependent, and can be calibrated using the walking avatar model. Let  $f$  and  $f'$  denote the average frequencies of the two gait sequences, computed from  $m_{00}$  of sequence 1 and  $m'_{00}$  of sequence 2. Then  $\rho = f'/f$ . To determine the relative phase, we crop a subsequence of temporal length  $f$  from  $m_{00}(t)$ , expand or contract it by  $\rho$ , then correlate with  $m'_{00}$ . The average lag of prominent peaks of the correlation result determines the relative phase. There may be a two-fold ambiguity in the phase from those viewpoints for which the autocorrelation of  $m_{00}$  yields peaks at half the stride frequency. For people close to the camera, the perspective effects are usually enough to uniquely determine the phase. For people far away, however, it can be difficult to distinguish between left foot forward or right foot forward on the basis of silhouette moment information alone.

After determining the temporal mapping between  $t$  and  $t'$ , we now align the frieze patterns spatially. Given the moments that we can compute from frieze patterns, we determine the two translations and two scale factors that relate  $(x, y)$  and  $(x', y')$  for corresponding time steps in the two sequences. Dropping the time variables from the notation, this affine transformation is found to be

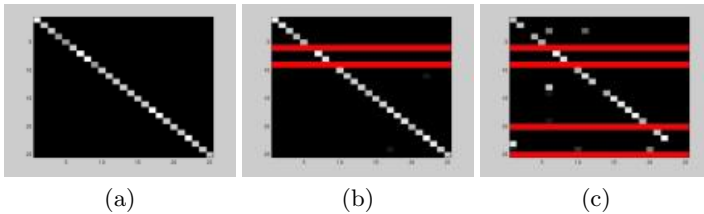
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{\mu'_{20} m_{00}}{\mu_{20} m'_{00}}} & 0 \\ 0 & \sqrt{\frac{\mu'_{02} m_{00}}{\mu_{02} m'_{00}}} \end{bmatrix} \begin{bmatrix} x - m_{10}/m_{00} \\ y - m_{01}/m_{00} \end{bmatrix} + \begin{bmatrix} m'_{10}/m'_{00} \\ m'_{01}/m'_{00} \end{bmatrix}$$

Whether to allow both scale factors to vary independently for each time step, to enforce their ratio to be constant, to compute a temporal average for each, or other variations depends on the application and on the amount of noise one can expect in the underlying silhouette data.

## 5.2 Applications of Gait Alignment

We illustrate the utility of moment-based frieze alignment with two applications. The first involves comparing frieze tiles to classify a walking person’s identity given a prior training set of gait data. The second application concerns matching a walking humanoid model to gait silhouette data from a real person, in order to locate specific body parts in each frame.

**Human Identification.** Given a dataset of gait sequences collected from one camera viewpoint, we want to analyze a new sequence to determine which person it is. Our approach is to create row and column silhouette projection friezes for each sequence, warp them all temporally to a canonical frequency and phase using the first half of the above alignment procedure, then cut out several tiles corresponding to individual strides from each sequence. These aligned frieze tiles are compared using normalized correlation, and subject classification is performed by nearest neighbor matching on correlation scores. This approach implicitly captures biometric shape cues such as body height/width ratio, body-part proportions, stride length and amount of arm swing.

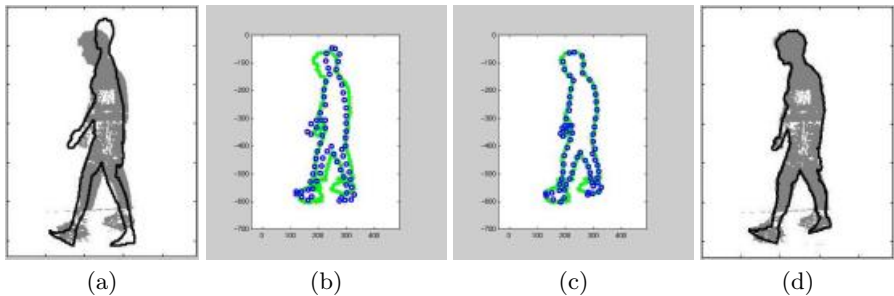


**Fig. 8.** Confusion matrices for nearest neighbor classification of 25 human subjects using gait frieze pattern tiles. (a) Result from training and testing on non-overlapping slow walking gait sequences. Classification rate is 100%. (b) Training on slow walk, testing on fast walk. Classification rate is 100%. (c) Training on slow walk, testing on walking carrying a ball (to inhibit arm swing). Classification rate is 81%. Blank rows in (b) and (c) denote subjects for which there is no corresponding test data available.

To test this approach, we use the CMU MoBo database [9], which contains motion sequences of 25 subjects walking on a treadmill. Each subject is recorded performing four different types of walking: slow walk, fast walk, inclined walk, and slow walk holding a ball (to inhibit arm swing). Figure 8 shows results achieved for side views, for gait combinations slow-slow, slow-fast and slow-ball. For the slow-slow experiment, the gallery consisted of tiles from the first five seconds of each subject’s slow walk gait sequence, and the probe set consists of tiles from the last five seconds of the same sequences. For both slow-fast and slow-ball, the classification algorithm is trained on all tiles from the slow walk sequences, and tested on all tiles from the other two gait sequences. We see that

the results are quite good, even across different gait types. Although the match similarity metric is simple normalized correlation, each tile succinctly represents both spatial and temporal information from an entire stride subsequence.

**Model-Based Body Part Analysis.** Assume that we know the camera viewpoint, and have rendered a walking humanoid model from that viewpoint. We now have a sequence of model body silhouettes that can be matched against a real gait sequence. After spatio-temporal gait alignment, the temporal pairing of each frame of the data sequence with a corresponding model frame is known, along with a four parameter affine transformation that aligns those two binary silhouettes. Thus, for each frame, we can project the model silhouette contour onto the data silhouette image. A sample frame showing an overlaid model contour found through automatic gait sequence alignment is shown in Figure 9A. The aligned model contour does not exactly coincide with the person's body

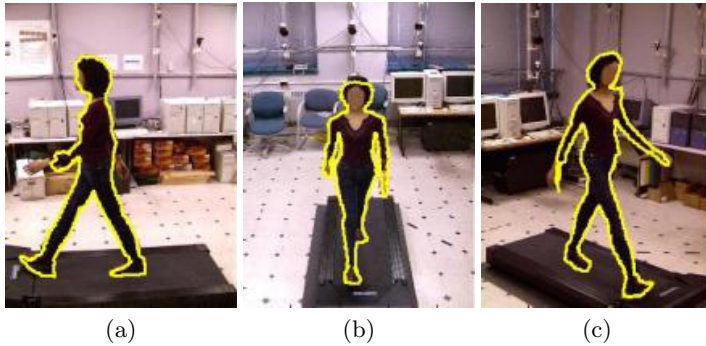


**Fig. 9.** (a) Moment-based alignment of model and data silhouettes. (b) Sampled points from model and data silhouette contours. (c) Results of non-rigid thin-plate spline alignment of the two sets of sample points. (d) Model silhouette warped by thin-plate spline transform, overlaid on data silhouette.

outline due to a variety of factors, including differences in body shape and joint angle kinematics between the avatar and the human being (e.g. body proportions and amount of arm swing), as well as small differences in camera perspective between the model and data viewpoints. However, note that the overall temporal and spatial alignment is quite good, in the sense that the aligned model tells us what body parts should be visible, and roughly where they should appear in the image. More importantly, we know which body parts are occluded and should not be considered for further analysis in this frame.

To illustrate what can potentially be done given this initial alignment between model and data silhouettes, we uniformly sample points from along each silhouette contour and use a program for non-rigid point matching to compute a thin-plate spline transformation between them [10]. Figure 9 shows, from left to right, the initial model contour alignment, the two sampled point sets, the

resulting point sets after warping non-rigidly by a thin-plate spline, and the new warped model contour overlaid over the data silhouette. The agreement between contours is now much improved. The success of the non-rigid point matcher in this case is due in large part to the accuracy of the model silhouette topology, as determined by moment-based alignment of gait frieze patterns. More examples are shown in Figure 10. Model-based gait analysis using frieze patterns offers an efficient alternative to kinematic body part tracking for determining the location of individual body parts in each frame of a gait sequence.



**Fig. 10.** Spatially and temporally aligned model silhouette overlaid on original image for three views taken simultaneously by synchronized cameras. We plan to use results like these for further model-based body part analysis, ultimately leading to 3D body reconstruction and motion capture of a walking human.

## 6 Summary

We have presented a periodic pattern representation for analyzing gait sequences. Silhouette row and column projections are stacked over time to form frieze patterns that can be analyzed using the mathematical theory of symmetry groups. With the help of a walking humanoid avatar, we have studied the correlation between the seven frieze symmetry groups and gait viewing direction, and have developed practical techniques for classifying imperfect frieze patterns. Our future work will explore methods for more efficient and accurate viewpoint estimation from frieze patterns, and extend our mathematical methods for imperfect pattern analysis to patterns that are periodic along two dimensions. We have also presented a moment-based method for aligning frieze gait patterns both temporally and spatially. The method has applications in determining human identity from gait biometrics, and it provides an efficient alternative to frame-by-frame tracking approaches for locating and delineating body parts.



## References

1. B.K.P.Horn. *Robot Vision*. MIT Press, 1986.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 8–15, 1998.
3. Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *IEEE Computer Vision and Pattern Recognition*, pages II:682–689, 2000.
4. T.J. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages II:239–245, 1999.
5. R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis and applications. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 22(8):781–796, August 2000.
6. E.Ayyappa. Normal human locomotion, part 1: Basic concepts and terminology. In *Journal of Prosthetics and Orthotics*, volume 9(1), pages 10–17. The American Academy of Orthotists and Prosthetists, 1997.
7. A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision*, pages 751–767, 2000.
8. M.A. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *International Journal of Computer Vision*, 38(1):59–73, June 2000.
9. R. Gross and J. Shi. The CMU motion of body (MoBo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, 2001.
10. H.Chui and A.Rangarajan. A new algorithm for non-rigid point matching. *IEEE Computer Vision and Pattern Recognition*, pages 44–51, 2000.
11. D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.
12. K. Kanatani. *Statistical Optimization for Geometric Computation : Theory and Practice*. North-Holland, 1996.
13. K. Kanatani. Comments on "Symmetry as a Continuous Feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):246–247, 1997.
14. J.J. Little and J.E. Boyd. Recognizing people by their gait: The shape of motion. *Videre*, 1(2), 1998.
15. F. Liu and R. W Picard. Finding periodicity in space and time. In *IEEE International Conference on Computer Vision (ICCV)*, 1998.
16. Y. Liu and R. T. Collins. A Computational Model for Repeated Pattern Perception using Frieze and Wallpaper Groups. In *Computer Vision and Pattern Recognition Conference (CVPR'00)*, pages 537–544, Los Alamitos, CA, June 2000. IEEE Computer Society Press. ([http://www.ri.cmu.edu/pubs/pub\\_3302.html](http://www.ri.cmu.edu/pubs/pub_3302.html)).
17. S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 469–474, 1994.
18. S.M. Seitz and C.R. Dyer. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25:1–23, 1997.
19. T.N. Tan, G.D. Sullivan, and K.D. Baker. Recognizing objects on the ground-plane. *Image and Vision Computing*, 12(3):164–172, April 1994.
20. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision*, pages 255–261, 1999.
21. H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1154–1165, December 1995.

# Feature-Preserving Medial Axis Noise Removal

Roger Tam and Wolfgang Heidrich

Department of Computer Science,  
University of British Columbia  
{rtam|heidrich}@cs.ubc.ca

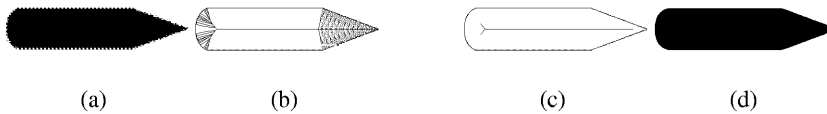
**Abstract.** This paper presents a novel technique for medial axis noise removal. The method introduced removes the branches generated by noise on an object's boundary without losing the fine features that are often altered or destroyed by current pruning methods. The algorithm consists of an intuitive threshold-based pruning process, followed by an automatic feature reconstruction phase that effectively recovers lost details without reintroducing noise. The result is a technique that is robust and easy to use. Tests show that the method works well on a variety of objects with significant differences in shape complexity, topology and noise characteristics.

## 1 Background and Motivation

The medial axis [1] is a useful shape representation for many applications in computer vision and computer graphics. The primary drawback of the medial axis is that it is very sensitive to minor perturbations of the object's boundary, such as that caused by discretization, segmentation errors, image noise, and so forth. The goal of most medial axis *pruning* techniques is the removal of branches associated with these artifacts, resulting typically in a much cleaner and more usable medial axis. In addition, the denoised axis can then be used to reconstruct a smoother version of the original object.

Most current medial axis pruning algorithms suffer from the problem that when excess branches are removed, other branches that correspond to fine but perceptually significant features of the object are excessively shortened. This is primarily due to the fact that most pruning methods use a global significance measure (*e.g.*, feature size or frequency) to discern between data and noise. Unfortunately, for most measures there is a significant overlap between what is considered noise and data, and when the noise is removed some data is taken with it. Figure 1 provides a very simple, motivating example. The unprocessed medial axis (Fig. 1b) has many spurious branches, largely because of discretization artifacts. Figure 1c shows the typical result of pruning with a global threshold. In this case the significance measure is noise size. The result is that all of the branches associated with noise are gone, but the remaining branches are also shortened, causing the tip of the pencil to become rounded (Fig. 1d).

So far, proposed solutions to address this issue have proven inadequate. Attempts to overcome the noise/data overlap problem by developing more complicated global measures frequently result in a fuzzy relationship between parameter values and how they correspond to changes in object features, thereby making the estimation of an appropriate threshold more difficult. Another general approach is to recover lost details

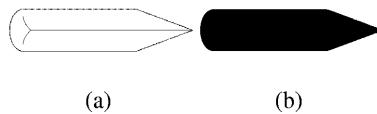


**Fig. 1.** Pencil (a) Original object (b) Unpruned axis (c) Typical pruned axis (d) Object reconstructed from typical pruned axis

by *unpruning* the remaining branches after the noisy branches are removed. Current algorithms using this approach typically do not work well because they depend on a global threshold for the unpruning process as well, thereby subjecting it to the same overlap problem.

In this paper, we present a novel approach for medial axis denoising that removes unwanted artifacts while preserving fine features, such as sharp corners and thin limbs. Our method first prunes the axis by using an intuitive global threshold based on noise size, then *automatically reconstructs* the fine features by extending the remaining branches. The use of a simple pruning method based on a physically meaningful parameter followed by an effective feature reconstruction process makes the technique robust and easy to use. The user can determine an appropriate threshold simply by estimating the size of the noise he or she wants to remove and in most cases, rough estimates are adequate because the reconstruction process can automatically correct many errors caused by overly aggressive pruning.

Our feature reconstruction algorithm extends each branch by using local shape information and does not depend on a global threshold. This process localizes the discernment between data and noise to the feature level which significantly reduces the overlap problem. As demonstrated in our results, this localization allows each branch to be extended to an appropriate length and in the correct direction so that each feature is reconstructed accurately without reintroducing noise. Figure 2a shows the medial axis of the pencil after processing with our algorithm; the noise is gone and the tip of the pencil is still sharp.



**Fig. 2.** Pencil (a) Axis processed by our denoising algorithm (b) Our reconstructed object

Although simple in its design, we can show that our technique works well for removing artifacts of various sizes and characteristics from objects of arbitrary shape complexity and topology. We have tested our algorithm on a wide variety of data, a number of examples of which are included in this paper to demonstrate the effectiveness of our method.

## 2 Related Work

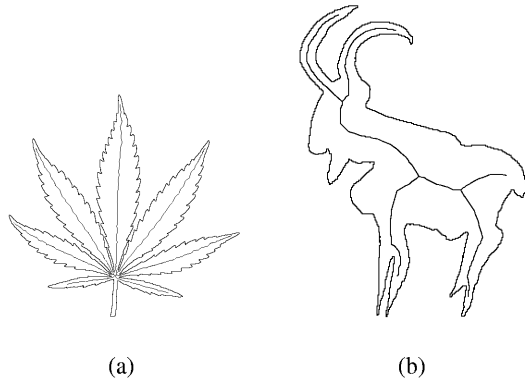
Our method for the construction of the medial axis is one of a number of algorithms that use the Voronoi graph of a set of sample points regularly spaced along the object's boundary to form an approximation of the skeleton. The main idea of such algorithms, examples of which include [2,3,4], is to first compute the Voronoi graph of the points, then extract a subgraph to form the skeleton. For example, the subgraph can be extracted by taking only the Voronoi vertices that are inside the boundary of the object.

Given a model or an image of an object, there are two main approaches for producing a clean medial axis. The first approach performs some form of preprocessing on the image or model before computation of the medial axis. Such preprocessing usually consists of blurring (*e.g.*, [5]) or boundary smoothing (*e.g.*, [6]) of the original object to reduce spurious branches. Blurring and smoothing techniques can result in undesirable structural changes to the medial axis [7,8]. In addition, these operations typically use a global scale measure (*e.g.*, size of smoothing kernel) to filter out noise, and smaller object features are often altered or destroyed during preprocessing.

The other main approach is to start with the complete axis and prune the branches using some heuristic (*e.g.*, [3,9]). The general idea is to have a significance measure that assigns an importance value to each branch. During pruning, this value is compared to a user-given threshold to determine how much of each branch gets cut. With an ideal significance measure and threshold, only the parts of the axis associated with noise would be removed, and the rest of the axis would remain unaltered. However, for currently available measures there is usually an overlap between data and noise; a threshold value that completely removes the branches associated with noise will usually shorten the remaining branches as well, often to an undesirable degree. Thus, finding a good threshold value often requires striking a delicate balance between noise removal and feature preservation. In addition, the complexity of some measures makes them seem ad hoc and adds to the difficulty of finding an appropriate threshold. In some cases, even multiple parameters are required (*e.g.*, [10]). To overcome the difficulties in estimating parameter values, a completely automatic method for threshold selection is proposed in [7,11]. The method is able to determine an appropriate value for many shapes, but there are instances in which the algorithm strongly oversegments the shape, resulting in large missing features. Figures 3a and 3b, generated with the algorithm from [7], show two examples. In Fig. 3a, the stem of the leaf is missing from the axis; in Fig. 3b, two of the goat's legs are among the larger features not represented.

A number of researchers have proposed methods that utilize a postprocess to recover small details lost by pruning. Such methods add an unpruning process that extends the branches that remain after pruning (*e.g.*, [8,11]). The typical approach is to use the same significance measure as used for pruning and simply apply a different threshold to extend the branches. This approach forces the user to select two thresholds, and still the problem of overlap between data and noise is not solved. This frequently results in some branches being overextended (*i.e.*, noise is reintroduced) while others are still too short.

Our algorithm is designed to address the problems described above, and consists of the following two main processes:



**Fig. 3.** Ogniewicz's (a) Leaf (b) Goat

1. A threshold-based pruning technique with a single parameter and a simple significance measure that gives the user intuitive control.
2. An automatic feature reconstruction process that extends each branch using local shape information and does not depend on a global threshold.

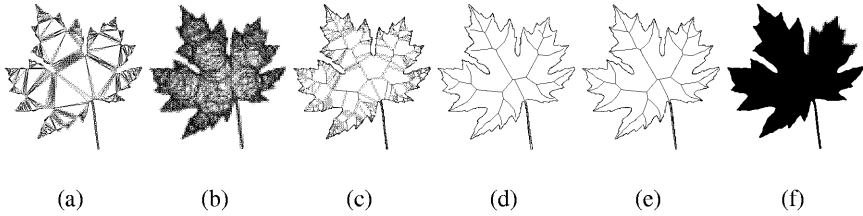
The result is an algorithm that gives the user the freedom to select a threshold that completely removes all noise while providing a reliable feature reconstruction process which brings back the right amount of detail at each branch. This technique gives the user some control, so that large features are not accidentally removed, but hides the more complex data/noise discernment algorithm inside an automatic process so that the user is not burdened with a complicated significance measure.

It should be noted that some pruning methods, such as [11], are hierarchical in nature and can produce results at multiple levels of detail. Thus, at coarser levels, the loss of fine features is considered acceptable, even appropriate. In contrast, our algorithm is designed to remove artifacts of a given size, while preserving as much detail in the rest of the object as possible. However, at finer levels of detail, the goals of the algorithms are essentially the same.

### 3 Methodology

Given the boundary points of an object, the main steps of our algorithm for medial axis noise removal are as follows:

1. Construct the medial axis from the boundary points (Figs. 4a-c).
2. Prune the spurious branches (Fig. 4d) by using a user-determined global threshold.
3. Extend the remaining branches to recover small details (Fig. 4e) by using a local measure of shape smoothness to distinguish between data and noise.
4. Reconstruct the object with the clean medial axis (Fig. 4f).



**Fig. 4.** Maple leaf example (a) Delaunay triangulation (b) Union of Circles (c) Complete medial axis (d) Pruned medial axis (e) Pruned medial axis with details recovered (f) Denoised shape

### 3.1 Medial Axis Construction

Like most Voronoi-based methods for medial axis construction, our method assumes that the sample points are spaced with sufficient density along the boundary of the object. Our method for the computation of the medial axis from a boundary point set can be divided into three main steps:

1. Compute the Delaunay triangulation of the point set and discard any triangles that are outside of the object (Fig. 4a).
2. Compute the circumscribing circle of each remaining triangle (Fig. 4b). This set of circles is referred to as a *Union of Circles (UoC)* [12].
3. Construct the medial axis by connecting the centres of the circles (Fig. 4c).

We represent the medial axis as a directed graph<sup>1</sup> whose root node is the centre of the largest circle in the UoC. Construction of the graph begins by creating a line segment (called an *axial segment*) between the root node and each of its neighbours (two circles are *neighbours* if their corresponding triangles share an edge). The root node is the *parent* node and the neighbours are *child* nodes. This process is then repeated with the neighbours as parent nodes until all circles in the UoC are linked. The result is a medial axis that can be traversed recursively by starting at the largest circle and following the child nodes until they reach the boundary of the object. With this construction method, each node in the axis has a corresponding circle in the UoC, which in turn has a corresponding triangle in the Delaunay triangulation.

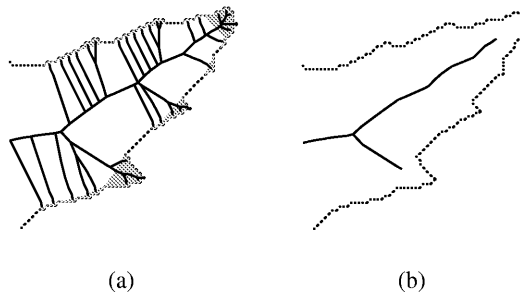
In this paper, a node that has no children is referred to as an *end node*. A node that has more than one child is called a *branch node*. A *branch* is defined as any chain of nodes that has a single branch node at the beginning and an end node at the end.

### 3.2 Area-Based Pruning

The purpose of the pruning process is to remove parts of the medial axis that are associated with noise on the object's boundary. A significance measure is needed for determining

<sup>1</sup> For objects of genus zero, the graph is naturally a tree. For objects with holes, we break each cycle by imposing an appropriate breakpoint in the loop. This is only for the purposes of traversing the graph without running into infinite loops. In order to preserve the topology of the original object, cycles are never pruned.

whether a feature of the object should be considered as noise. In the context of pruning, a feature can be defined as the set of triangles associated with any subtree of the axis graph. Our significance measure assigns an importance value to a feature based on the surface area it covers. An example is shown in Fig. 5a, which shows a part of the maple leaf from Fig. 4. In this figure, the shaded regions are features that are smaller in area than the user-given significance threshold. The significance value of a feature can be determined by summing the areas of all triangles in the subtree associated with that feature. Any subtree that has a value below the threshold is pruned. Each feature can be seen as being *supported* by the branches of the subtree, so when the subtree is pruned, the feature is eliminated.



**Fig. 5.** Area-Based pruning (a) Unpruned axis (b) Pruned axis

Our significance measure has the following two main advantages:

- The pruning is guaranteed not to disconnect the graph, because a parent always has a higher significance value than its child.
- Area is a simple and intuitive significance measure and an appropriate threshold can be estimated via a typically straightforward analysis of the data acquisition method. Even when knowledge of the acquisition method is insufficient, a suitable value can be found by visual inspection of the data more easily than most heuristics-based measures that have a less direct physical meaning.

The result of the pruning process is that all noise below the threshold size is eliminated. With an appropriate threshold, the only branches that remain are associated with significant features of the object. As mentioned, a side effect of pruning with a fixed global threshold is that the remaining branches are typically shorter than they should be and fine but important details are often lost.

### 3.2.1 Noise Model and Threshold Selection

Like practically all medial axis pruning methods, our noise model focuses on artifacts in the form of relatively small protrusions from a larger body. Pruning techniques are the most effective when applied to this type of noise, called *additive noise*, because they work by removing and shortening branches. A common example of additive noise are the artifacts originating from the dark current in CCD cameras. We define two conditions that must be satisfied in order for a feature to be classified as noise:

1. The size of the feature is smaller than the user-determined threshold. This is the condition used in the pruning phase.
2. The feature is not *smoothly connected* to the rest of the object. The transition into a protrusion is considered smooth if the abruptness of the narrowing does not exceed the changes in width in the parts of the object leading up to the feature. This condition is used in the feature reconstruction phase and is defined more precisely in the next section.

To select an appropriate threshold for pruning, consideration must be given to the data and application at hand. If the noise characteristics are known, an estimate of the artifact size can be made, and selection of the threshold is relatively simple. Otherwise, the value can be set by visual inspection of the data. Our implementation is such that the threshold can be set as an absolute size or as a percentage of the total area of the object. In most of our examples, the value is determined interactively. The feature reconstruction process, as described in the next section, is robust enough to allow a fairly imprecise threshold selection, and a range of appropriate values exist for most objects.

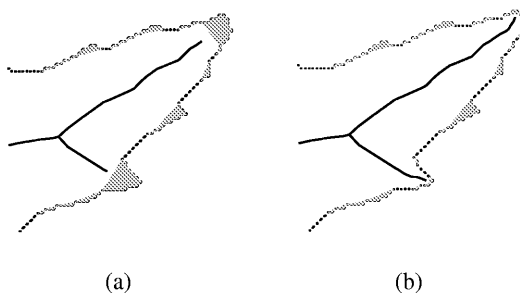
The most important guiding principle in selecting an appropriate threshold is to ensure that each significant protrusion in the object has a single supporting branch. The reason for this is best described by Leyton's Symmetry-Curvature Duality Theorem [13]:

**Theorem 1 (Symmetry-Curvature Duality).** *Any section of curve, that has one and only one curvature extremum, has one and only one symmetry axis. This axis is forced to terminate at the extremum itself.*

Figure 5b shows an example of how each significant feature is supported by a single branch. The result of pruning should be that each significant extremum in the border has a single remaining branch.

### 3.3 Feature Reconstruction

The purpose of the feature reconstruction process is to recover significant parts of the object that have been pruned because they fall below the size threshold. For example,

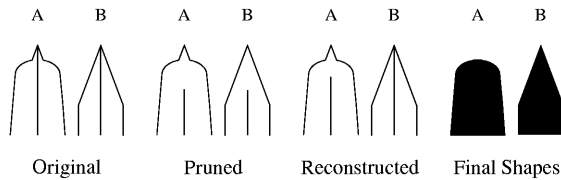


**Fig. 6.** Feature reconstruction (a) Before reconstruction (b) After reconstruction



Fig. 6 shows part of the maple leaf before and after reconstruction. The shaded areas show features that would be removed. The axis in Fig. 6b clearly represents the shape of the object better than the axis in Fig. 6a.

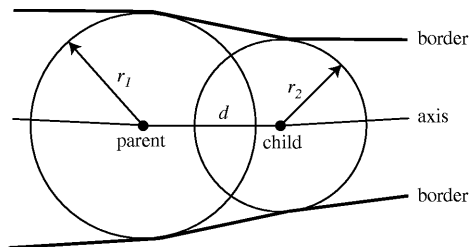
Our reconstruction algorithm works on one branch at a time, and its main idea is to use the shape information present in the remaining branches and circles to calculate local smoothness constraints that determine how far each branch can be extended to recover fine features without reintroducing noise. In this scheme, what is classified as noise varies from branch to branch. Figure 7 shows an example in which the branch in one feature (B) is extended further than in another (A), even though the tips of the features have the same angle and both branches reach the boundary in the unpruned axis. In this case, the small sharp point in Feature A is regarded as noise because it falls below the size threshold and violates the local smoothness constraints.



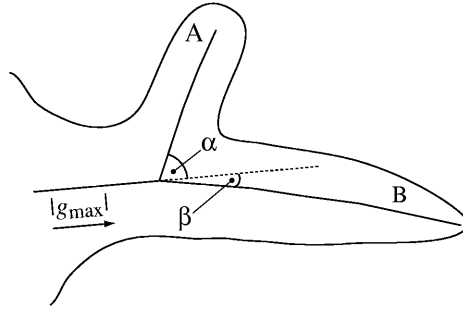
**Fig. 7.** Effect of smoothness constraints on feature reconstruction

The measure of smoothness that we use is termed the *axial gradient*, which measures the change in the width of the object per unit length of the axis. Each axial segment has an axial gradient value that is mathematically defined as the signed difference in radius between the child circle and the parent circle, divided by the Euclidean distance between the two nodes. Figure 8 illustrates this definition.

The algorithm works by starting at the root node and following the axis until it reaches the current end node of the branch in question, while keeping track of the greatest absolute axial gradient value ( $|g_{\max}|$ ) encountered along the path. This value is used to determine how far the branch can be extended. The reasoning is that if a feature at the end of the branch is below the threshold size and is marked by a narrowing that is more abrupt than any other change in width along the path, then the feature is most likely noise. Because  $|g_{\max}|$  is calculated individually for each branch, the data/noise overlap problem associated with global thresholds is significantly reduced.



**Fig. 8.** Axial gradient between two nodes ( $g = \frac{r_2 - r_1}{d}$ )



**Fig. 9.** Axial gradient at a branch node. Because  $\beta$  is smaller than  $\alpha$ ,  $|g_{\max}|$  is more relevant to Branch B than Branch A.

When calculating  $|g_{\max}|$  along a path, special consideration must be given to branch nodes, because the degree of *continuity* of a feature across a branch node depends strongly on the branching angles at these nodes. Figure 9 shows an example in which one feature (B) has stronger continuity across a branch node than another (A). In this case, the shape information along the path before the branch node is more relevant to Branch B than Branch A. For any given path, the branching angle at a branch node is a good indicator of how much of the maximum gradient encountered before the node should be “carried over” past the branch node. Intuitively, an angle of 0 degrees (maximum continuity) should impose no change to the maximum gradient, whereas an angle of 90 degrees or greater (no continuity) should cause  $|g_{\max}|$  to become 0 at the branch node. For an angle between 0 and 90, we use this formula:

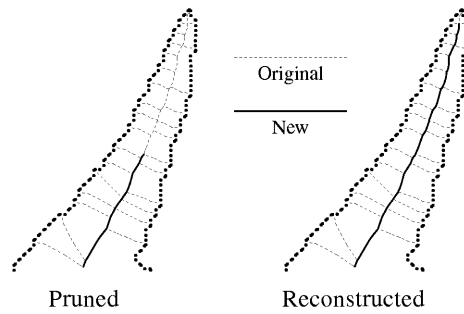
$$g_{\max}(\theta) = |g_{\max}| \times \frac{90 - \theta}{90}$$

where  $|g_{\max}|$  is maximum axial gradient before the branch node and  $g_{\max}(\theta)$  is the maximum axial gradient at the branch node for the segment with angle  $\theta$ .

Once  $|g_{\max}|$  for a given path has been calculated, the next step is to extend the branch at the end of the path. Two main issues need to be considered at this point. First, if there is more than one direction for possible extension (*i.e.*, the current end node is a branch node in the original axis), a decision needs to be made to determine which segment to follow. The second issue is how far to extend the branch.

The first issue is addressed in consideration of the second part of Leyton’s theorem, which says that the symmetry axis of a feature should terminate at the extremum of that feature. Given that the objects we are considering have many minor extrema due to noise, we need to distinguish these from the extrema associated with significant features. Again, we use the axial gradient for this purpose. If there is more than one path for possible extension, the segment with the lowest axial gradient is chosen. This method is essentially a greedy algorithm for finding the smoothest path. As shown in the Results section, this gives a high probability of reaching the correct extremum. Figure 10 illustrates an example.

The second issue of how far to extend the branch is addressed by comparing the absolute value of the axial gradient of the chosen candidate segment ( $|g_{\text{cand}}|$ ) with the



**Fig. 10.** Branch extension. At each branch node, the segment with the lowest axial gradient is chosen, which extends the branch toward the appropriate extremum.

$|g_{\max}|$  of the branch. If  $|g_{\text{cand}}| \leq |g_{\max}|$ , the segment is added to the end of the branch. Note that because our noise model defines noise as small protrusions from a larger body, a feature can be classified as noise only if  $g_{\text{cand}} < 0$ . Segments are added until  $g_{\text{cand}} < 0$  and  $|g_{\text{cand}}| > |g_{\max}|$ , or there are no more candidate segments.

As demonstrated by the examples in the Results section, our branch extension algorithm is very effective in automatically reconstructing features to their appropriate degree of sharpness without reintroducing noise.

### 3.4 Shape Reconstruction

An advantage to using the UoC method for medial axis construction is that there is a one-to-one correspondence between the nodes of the axis and the set of circles. To reveal the final object shape after pruning and feature reconstruction, we simply take the union of the circles associated with the new axis. The triangles associated with the new axis can be used efficiently for applying error metrics based on area.

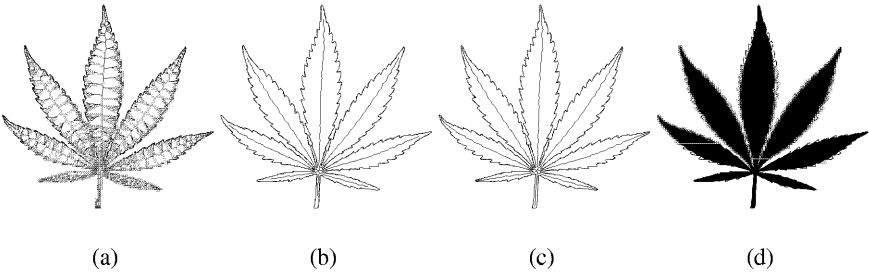
## 4 Results

We have tested our algorithm on many objects with various amounts of noise. Our data sources include map data, medical images, aerial photographs and specially designed/chosen test models. In this section, a number of examples are used to illustrate the effectiveness of our method. Possible extensions to overcome some of the current limitations of our algorithm are also discussed. The examples include four synthetic objects and one object from an MRI image. The characteristics of the objects are shown in Table 1. All thresholds ( $t$ ) are specified as a percentage of the total area of the original object.

Figures 11 (Leaf) and 12 (Goat) are good examples of how the algorithm can reconstruct fine features after removing unwanted artifacts. In the Leaf, the small variations along the border are removed, resulting in a much smoother shape. However, the fine features such the tips of the leaflets and the thin stem are nicely reconstructed. This example can be compared to the result by Ogniewicz in Fig. 3a, where the stem is missing

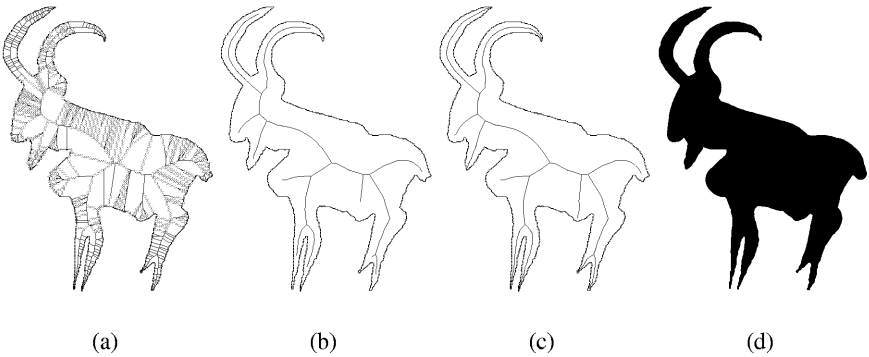
**Table 1.** Characteristics of our test objects

Object	Figure	Dimensions	Threshold ( $t$ )
Leaf	11	$479 \times 462$	0.3%
Goat	12	$254 \times 344$	0.4%
Lizard	13	$443 \times 446$	2.0%
Brain	14	$255 \times 293$	0.1%
Rectangle	15	$339 \times 238$	2.0%

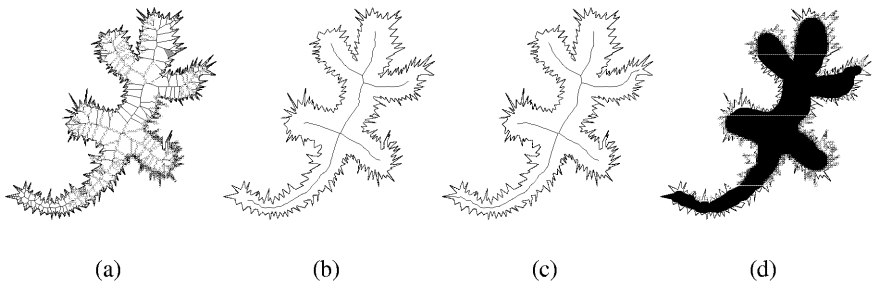


**Fig. 11.** Leaf ( $479 \times 462$ ,  $t = 0.3\%$ ) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

completely. The Goat shows how the branch extension method can use local shape information to reconstruct features to differing degrees of sharpness where appropriate. For the sharper features, such as the horns, goatee and legs, the branches are extended to the tips. For the more rounded features, such as the mouth, chest, belly and tail, the branches are extended enough to fully reconstruct the features, but not so far as to reintroduce noise.



**Fig. 12.** Goat ( $254 \times 344$ ,  $t = 0.4\%$ ) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape

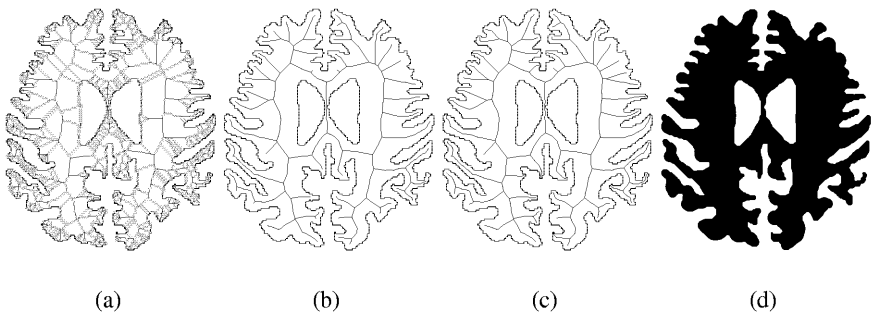


**Fig. 13.** Lizard ( $443 \times 446$ ,  $t = 2.0\%$ ) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

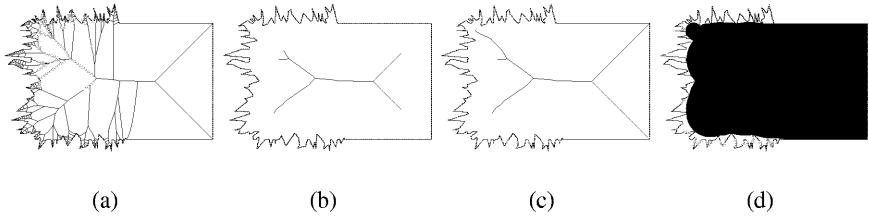
Figure 13 (Lizard) is an example where the “noise” artifacts are quite large. Our algorithm still results in a nicely denoised shape in this case. In the head and the four legs, the reconstructed branches do not extend into any of the spurious spikes along the border. The shape of the tail causes its supporting branch to be extended to the tip. It is somewhat debatable whether the spike at the end should be considered as a significant feature or noise, but in this case its inclusion seems appropriate.

Figure 14 (Brain) shows an object from an MRI image. The noise in this case is a combination of image noise, segmentation errors and discretization artifacts. The shape of this object is significantly more complex than in the other examples. This object is also of a different topology in that it has two holes. Our algorithm is able to effectively remove the various types of noise from all areas of the object.

Figure 15 (Rectangle) is an example of an object that has a heterogeneous distribution of noise. In this case, the left side of the rectangle is very noisy, whereas the right side is clean. Again, the reconstruction algorithm is able to perform well, extending branches to their appropriate lengths so that the corners on the right side are sharp, while the artifacts on the left side are removed.



**Fig. 14.** Brain ( $255 \times 293$ ,  $t = 0.1\%$ ) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape

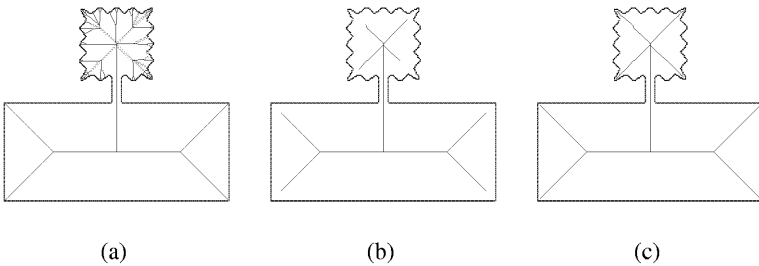


**Fig. 15.** Rectangle ( $339 \times 238$ ,  $t = 2.0\%$ ) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

#### 4.1 Extensions

We are considering a number of extensions to our algorithm to overcome some of the current limitations:

- Our use of the maximum axial gradient as a local smoothness constraint inherently assumes that the axial gradient does not vary greatly within a single feature, at least not relative to the gradient of any noise at the end of the branch. Although such variations are not encountered frequently, they certainly can exist. Figure 16 shows an example of such a situation. In this case, a thin *neck*, or local narrowing, in the object causes the branches in the noisy square to be overextended into the corners. A possible solution to this problem is to impose a limit on the length of the path used for computing the maximum gradient. For example, instead of only having one root node from which to start, we can break the graph down into subgraphs using features such as necks to do the division. Although theoretically straightforward, this has not been implemented at the time of writing of this paper.



**Fig. 16.** Narrow neck example (a) Original axis (b) Pruned axis (c) Axis after feature reconstruction. The branches are extended too far.

- The branch extension algorithm may not work well in the case of a very short branch with a large branching angle, because there would be very few axial segments and, therefore, a very limited amount of local shape information with which to calculate an appropriate maximum axial gradient. Increasing the sampling density

of the boundary points may be a possible solution, but this would increase the computational costs considerably.

- Although our tests show it to be largely effective, the greedy algorithm for finding the smoothest path for branch extension is not *guaranteed* to reach the correct extremum. A solution would be to search further into the tree before making a path selection.

## 5 Summary and Future Work

We have presented a novel algorithm for medial axis noise removal. Our algorithm consists of a threshold-based pruning method that uses a simple significance measure, followed by an automatic feature reconstruction process that extends the remaining branches to recover fine features without reintroducing noise. Our method is easy to use and our results show that the approach has strong potential to be viable in practical applications.

There are a number of areas, in addition to the extensions described in Sect. 4.1, that we intend to include as part of our future research:

- Although in our experience the method presented is robust with respect to changes in the user-selected threshold (Sec. 3.2), a more formal evaluation of the algorithm's sensitivity should be done.
- We intend to further investigate the axial gradient approach by testing alternate formulations, such as using the average gradient with a tolerance instead of the maximum gradient.
- We would like to develop a method for automatically determining the pruning threshold. Although our current technique makes threshold selection relatively easy, an automatic method would be useful for processing large data sets. To achieve this goal, a more precise noise model may be required (*e.g.*, [10]).
- We would like to extend the method to 3D.

**Acknowledgments.** We would like to thank Robert Ogniewicz for making his pruning program and test data publicly available. This research is supported by the Natural Sciences and Engineering Research Council, British Columbia Advanced Systems Institute, and the University of British Columbia. This paper is dedicated to the memory of Alain Fournier.

## References

1. Blum, H., Nagel, R.: Shape description using weighted symmetric axis features. *Pattern Recognition* **10** (1978) 167–180
2. Attali, D., Montanvert, A.: Semicontinuous skeletons of 2D and 3D shapes. In: *Proceedings of the International Workshop on Visual Form, Capri, World Scientific (1994)* 32–41
3. Brandt, J., Algazi, V.: Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding* **55** (1992) 329–337

4. Ogniewicz, R., Ilg, M.: Voronoi skeletons: Theory and applications. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (1992) 63–69
5. Pizer, S., Oliver, W., Bloomberg, S.: Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Trans. Pattern Analysis and Machine Intelligence* **9** (1987) 505–511
6. Mokhtarian, F., Mackworth, A.: A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence* **14** (1992) 789–805
7. Ogniewicz, R.: Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory (1995)
8. Shaked, D., Bruckstein, A.: Pruning medial axes. *Computer Vision and Image Understanding* **69** (1998) 156–169
9. Attali, D., Sanniti di Baja, G., E., T.: Pruning discrete and semicontinuous skeletons. In De Floriani, C., Braccini, C., Vernazza, G., eds.: *Lecture Notes in Computer Science, Image Analysis and Processing*. Volume 974. Springer-Verlag (1995) 488–493
10. Attali, D., Montanvert, A.: Modeling noise for a better simplification of skeletons. In: Proc. of the International Conference on Image Processing. Volume III., Lausanne, Switzerland (1996) 13–16
11. Ogniewicz, R.: Skeleton-space: A multiscale shape description combining region and boundary information. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Seattle, WA (1994) 746–751
12. Ranjan, V., Fournier, A.: Matching and interpolation of shapes using unions of circles. *Computer Graphics Forum (Proceedings of Eurographics '96)* **15** (1996) 35–42
13. Leyton, M.: Shape and causal-history. In Arcelli, C., Cordella, L., Sanniti di Baja, G., eds.: *Visual Form: Analysis and Recognition*. Plenum Press (1992) 379–388



# Hierarchical Shape Modeling for Automatic Face Localization

Ce Liu<sup>1</sup>, Heung-Yeung Shum<sup>1</sup>, and Changshui Zhang<sup>2</sup>

<sup>1</sup> Visual Computing Group, Microsoft Research Asia, Beijing 100080, China

<sup>2</sup> Department of Automation, Tsinghua University, Beijing 100084, China  
`lce@msrchina.research.microsoft.com`

**Abstract.** Many approaches have been proposed to locate faces in an image. There are, however, two problems in previous facial shape models using feature points. First, the dimension of the solution space is too big since a large number of key points are needed to model a face. Second, the local features associated with the key points are assumed to be independent. Therefore, previous approaches require good initialization (which is often done manually), and may generate inaccurate localization. To automatically locate faces, we propose a novel hierarchical shape model (HSM) or multi-resolution shape models corresponding to a Gaussian pyramid of the face image. The coarsest shape model can be quickly located in the lowest resolution image. The located coarse model is then used to guide the search for a finer face model in the higher resolution image. Moreover, we devise a Global and Local (GL) distribution to learn the likelihood of the joint distribution of facial features. A novel hierarchical data-driven Markov chain Monte Carlo (HDDMCMC) approach is proposed to achieve the global optimum of face localization. Experimental results demonstrate that our algorithm produces accurate localization results quickly, bypassing the need for good initialization.

## 1 Introduction

Face detection and face localization have been challenging problems in computer vision and machine perception. *Face detection*, for example, explores possible locations of faces from an input image, and *face localization* accurately locates the facial shape and parts, often from an initialized model. *Appearance models* have been successfully used for face detection, where typically a square region with an elliptic mask is used to represent a face image. Based on a large amount of positive (face) and negative (non-face) samples, machine learning techniques such as PCA [13], neural networks [9,11], support vector machines [6], wavelets [10] and decision trees [14], are always used to learn the separating manifold of faces and non-faces. By verifying patterns in a shifting window, the position of a face can be derived.

However, an appearance model alone is not flexible enough to model shape deformations and pose or orientation variations. Shape models, in particular deformable shape models such as deformable template matching [15] and graph

matching [4], have been used for face localization, *i.e.*, finding accurate facial shape and parts. A good example is the active shape model (ASM) [1] where a Bayesian approach using a mixture of Gaussians is adopted. The prior of shape and the likelihood of local features given each point are separately learnt, and Bayesian inference is chosen to obtain the maximum a *posteriori* (MAP) solution. They also developed and improved active appearance models (AAM) [2] to locate faces.

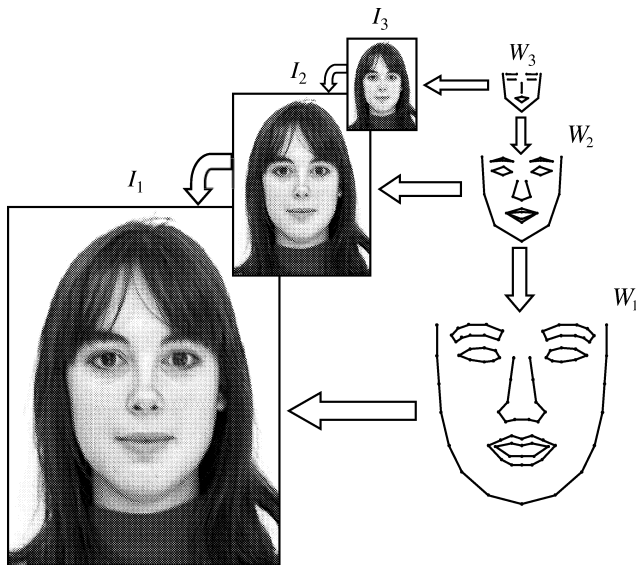
There are, however, two problems with previous shape models for locating faces. First, face localization is not automatic because of the huge solution space of shape and position. Typically we have to use a large number of feature points (*e.g.*, around 80 in [1]) to represent faces. Good initialization must be provided so that the optimization with MAP would converge to the global optimum. Often manual initialization is required. Second, even if a good initialization is provided, the localization results may not be accurate because the likelihood for local features was not modeled properly in previous approaches. For instance, distributions for features are assumed to be independent in the ASM model.

To address these two problems, we propose a *hierarchical shape model* (HSM) for automatic face localization. Multiple levels of shapes (with feature points) from coarse to fine, are employed to represent faces in a face image pyramid from low-resolution to high-resolution. First, the coarsest shape model is located in the lowest resolution image. Then we can gradually infer a finer shape in a higher-resolution image from the located coarse shape in lower-resolution image. Therefore, the uncertainty of the solution space is significantly reduced. Our system can automatically find the face shape and location robustly and quickly.

In HSM, we model two types of priors: single-level distribution and conditional distribution of a lower level given its higher level. Both of them are modeled and learnt by a mixture of Gaussians. A key idea in HSM is the likelihood modeling. The local image patterns associated with the feature points are NOT assumed independent, but conditionally independent with respect to a hidden variable. Specifically, we propose a novel *global and local* (GL) distribution to model the joint distribution, also with a mixture of Gaussians. In addition, we need to learn the data driven proposal density to guess the location of face based on local image evidence.

To pursue global convergence of the solution, we employ a hierarchical data driven Markov chain Monte Carlo (HDDMCMC) [12] method to explore the solution space effectively. It is not only globally optimal compared with traditional gradient descent methods, but also efficient compared with common Monte Carlo methods. All the distributions in HSM are modeled with Gaussian mixtures, which can be reliably learnt by a reversible jump Markov chain Monte Carlo method [8,3].

This paper is organized as follows. Section 2 introduces the framework of HSM, including its formulation, Bayesian inference and the main concept of HDDMCMC. The sampling details of HDDMCMC are introduced in Section 3. Section 4 talks about how to model four types of distributions in HSM and



**Fig. 1.** Illustration of hierarchical shape model with three levels. Left is a Gaussian pyramid of a face image. Right is the hierarchical shapes explaining the image in corresponding levels.

briefly introduces the learning strategy. Experiments are shown in Section 5. Section 6 summarizes this article.

## 2 Hierarchical Shape Model

### 2.1 Hierarchical Modeling for Facial Shape

Feature points used in face shape models may have semantic meanings. For example, we usually choose corner and edge points of eyes, eyebrows, nose, mouth and face contour to model a face. Let  $W = \{(x_i, y_i), i = 1, \dots, n\}$  denote the shape, where  $(x_i, y_i)$  is the  $i$ th key point and  $n$  is the number of key points. Let  $I$  denote the image containing the face. The task of face localization is to infer  $W$  from  $I$ .

A *hierarchical shape model* (HSM) has multiple levels,  $W = \{W_l, l = 1, \dots, L\}$ , where  $W_1$  is the finest level of shape. The number of feature points in  $W_l$  is  $n_l$  and the  $j$ th feature point of  $W_l$  is denoted as  $W_l^{(j)}$ . Each feature point in coarse levels ( $W_2, \dots, W_L$ ) is generated as the weighted sum of chosen feature points in  $W_1$ . In practice, we choose  $n_{l+1}$  to be approximately half of  $n_l$ . A three-level HSM is shown in Fig 1. Let the Gaussian pyramid of image  $I$  be  $\{I_1, \dots, I_L\}$ . Then the correspondence is established between shape domain  $W_l$  and image domain  $I_l$ . The most important property of HSM is that significant semantic information is preserved across levels. As shown in Fig. 1, eyes, mouth, face contour are all modeled even in the coarsest level.

## 2.2 Bayesian Inference in Hierarchical Shape Model

Our task is to infer  $\mathcal{W} = \{W_1, \dots, W_L\}$  from image  $I_1$ :

$$\begin{aligned}
 \mathcal{W}^* &= \arg \max_{\mathcal{W}} p(\mathcal{W}|I_1) = \arg \max_{\mathcal{W}} p(W_1, \dots, W_L|I_1) \\
 &= \arg \max_{\mathcal{W}} p(W_L|I_1) \prod_{l=1}^{L-1} p(W_l|W_{l+1}, \dots, W_L, I_1) \\
 &= \arg \max_{\mathcal{W}} p(W_L|I_L) \prod_{l=1}^{L-1} p(W_l|W_{l+1}, I_l). \tag{1}
 \end{aligned}$$

$p(W_L|I_1) = p(W_L|I_L)$  because the information of  $I_L$  is enough to determine  $W_L$ , and so on to get  $p(W_l|W_{l+1}, I_l)$ . Obviously given  $I_l$  and  $W_{l+1}$ ,  $W_l$  only depends on  $I_l$ . We may have

$$\mathcal{W}^* = \arg \max_{\mathcal{W}} \prod_{l=1}^L p(W_l|I_l) = \arg \max_{\mathcal{W}} \prod_{l=1}^L p(I_l|W_l)p(W_l), \tag{2}$$

which is equivalent to

$$W_l^* = \arg \max_{W_l} p(I_l|W_l)p(W_l), l = L, \dots, 1. \tag{3}$$

In HSM, we shall gradually optimize  $W_l^*$  in Eqn.(3).

We decompose shape model  $W_l$  into two parts: the *external* parameters including centroid  $Z_l$ , scale  $s_l$  and orientation  $\theta_l$ , and the *internal* parameters  $w_l$ . With a linear transition matrix  $T_{(s,\theta)}$  to scale and rotate the shape  $w_l$ , we get

$$W_l = T_{(s,\theta)}w_l + Z_l, \quad T_{(s,\theta)} = s \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{4}$$

It is reasonable to assume that the external and internal parameters are independent

$$p(W_l) = p(w_l)p(Z_l)p(s_l)p(\theta_l). \tag{5}$$

## 2.3 Hierarchical Data-Driven Markov-Chain Monte Carlo

The Markov chain Monte Carlo (MCMC) method is a tool to sample high-dimensional distributions. It can be used in optimization if the objective function itself is a pdf or could be converted to a pdf. Particularly if the objective function is very complex with multiple peaks, MCMC has the good property of global convergence because it ensures the Markov chain to reach the global optimum with a certain probability. The inefficiency of MCMC could be improved by data driven MCMC (DDMCMC)[12]. The traditional MCMC method randomly walks through the parameter space while DDMCMC employs some heuristics from data to guide the walks. In HSM, we should devise the salient proposal density including both the heuristics given by the localization result on the higher level, and local cues directly from the image. This leads to a hierarchical

DDMCMC or HDDMCMC, which starts at the top level and propagates the optimal solution from higher level to lower level.

At the top level, the optimal  $W_L$  of  $I_L$  is determined by Metropolis-Hastings sampling. The Markov chain  $\{W_L(t)\}$  to sample  $p(W_L|I_L)$  is driven by the transition probability at time  $t$

$$\alpha = \min\left\{1, \frac{p(W'_L|I_L)q(W_L(t); W'_L, I_L)}{p(W_L(t)|I_L)q(W'_L; W_L(t), I_L)}\right\} \quad (6)$$

where  $W'_L$  is sampled from proposal density  $q(W'_L; W_L(t), I_L)$  and it is accepted as  $W_L(t+1)$  with probability  $\alpha$ . The proposal density has two components, the shape prior  $p(W_L)$  and *data-driven* part, or local hints from image  $I_L$  to the  $j$ th feature point  $q(W_L^{(j)}; W_L^{(j)}(t), I_L)$ . The optimal  $W'_L$  is selected from the samples  $\{W_L(t)\}$  with maximum a posteriori (MAP)  $p(W'_L|I_L)$ .

The next is to find the optimal  $W'_l$  from the higher level  $W_{l+1}^*$ . The sampling strategy is slightly different from Eqn.(6) because the localization  $W_{l+1}^*$  will guide the Markov chain in proposal density by

$$\alpha = \min\left\{1, \frac{p(W'_l|I_l)q(W_l(t); W'_l, I_l, W_{l+1}^*)}{p(W_l(t)|I_l)q(W'_l; W_l(t), I_l, W_{l+1}^*)}\right\} \quad (7)$$

where the proposal density  $q(W'_l; W_l(t), I_l, W_{l+1}^*)$  relies on  $W_{l+1}^*$  as well. The proposal density again includes two parts, shape prior propagation  $p(W_l|W_{l+1}^*)$  and local hints  $q(W_l^{(j)}; W_l^{(j)}(t), I_l)$  from  $I_l$ .

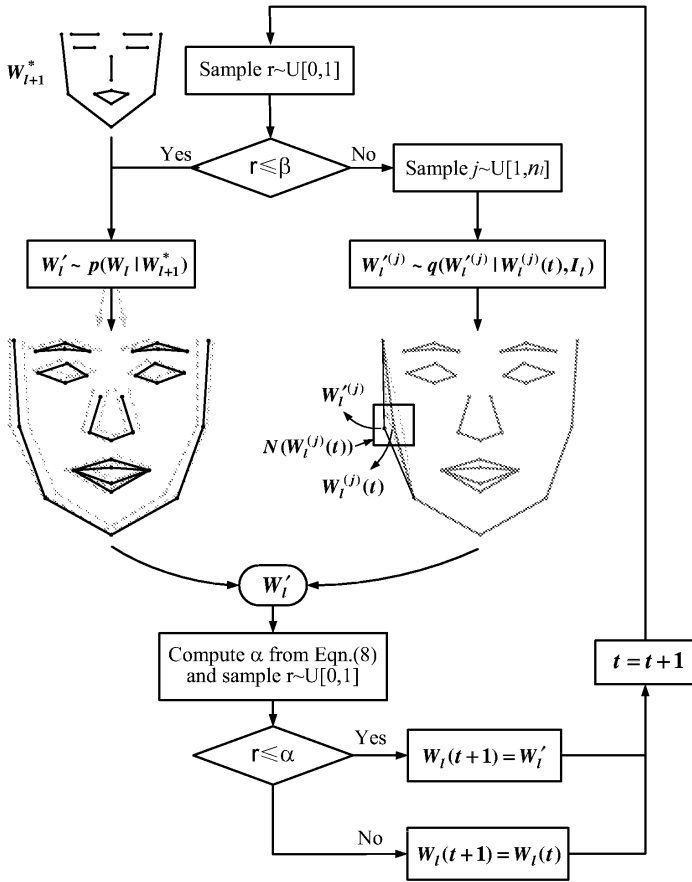
The hierarchical DDMCMC method (Eqn.(6) and (7)) is globally optimal since we shall finally sample the finest posterior  $p(W_1|I_1)$ . The information of higher level shape  $W_{l+1}^*$  propagates to lower level  $W_l$  via proposal density, and guides the Markov chain exploring the solution space  $p(W_l|I_l)$ . Thus, the error of  $W_{l+1}^*$  does NOT propagate to  $W_l$ . Since the entropy of  $p(W_l|W_{l+1}^*, I_l)$  is much smaller than  $p(W_l|I_l)$ , the Markov chain will hardly walk to those unlikely samples. Therefore, HDDMCMC is much more efficient than conventional DDMCMC that directly samples  $W_1$  from input image  $I_1$ .

### 3 Sampling Hierarchical DDMCMC

In this section, we explain the details of Metropolis-Hastings sampling Eqn.(6) and (7). Since the propagation from higher level to lower level is more general than sampling the top level, we focus on Eqn.(7). We shall also discuss how to give good initializations in the top level.

#### 3.1 Sampling Strategy

The basic task of MCMC is to sample a target density  $p(x)$ , but in most cases it is very difficult to directly sample  $p(x)$ . Therefore a proposal density  $q(x'; x(t))$



**Fig. 2.** The flowchart of hierarchical DDMCMC in HSM. We use the propagation form  $W_3$  to  $W_2$  as an example. The left branch is sampling the conditional prior from higher level result, while the right branch is sampling the position of each feature point.

is designed so that it is easy to draw samples  $x' \sim q(x'; x(t))$ . In our HSM, the proposal density  $q(W'_t; W_t(t), I_t, W_{t+1}^*)$  could be decomposed to

$$q(W'_t; W_t(t), I_t, W_{t+1}^*) = \beta p(W'_t | W_{t+1}^*) + (1 - \beta) q(W'_t; W_t(t), I_t), \quad (8)$$

where  $\beta$  is the probability of choosing *prior propagation* process  $p(W'_t | W_{t+1}^*)$  and  $1 - \beta$  is the probability of choosing *data-driven* process  $q(W'_t; W_t(t), I_t)$ , or sampling the feature points directly from the image.

Sampling the first part of Eqn.(8),  $p(W'_t | W_{t+1}^*)$ , is high-dimensional and non-trivial. But if we model the joint distribution  $p(W_t, W_{t+1})$  by a mixture of Gaussians, then the conditional density  $p(W_t | W_{t+1})$  is also a mixture of Gaussians that can be derived from  $p(W_t, W_{t+1})$ . At the top level without any prior propagation, we simply sample  $p(W_L)$  which is also modeled as a mixture of Gaussians.

To make it plausible to sample the second part of Eqn.(8), we design an individual proposal  $q(W'_l; W_l(t), I_l)$  for each feature point  $W_l^{(j)}$ ,  $j = 1, \dots, n_l$ :

$$q(W'_l; W_l(t), I_l) = \prod_{j=1}^{n_l} q(W_l^{(j)'}; W_l^{(j)}(t), I_l). \quad (9)$$

We may use Gibbs sampling to simply flip the position of one feature point at each time. Suppose the  $j$ th feature point is chosen, then after sampling,  $W'_l$  differs from  $W_l(t)$  only at  $W_l^{(j)'}(t)$ . Sampling  $q(W_l^{(j)'} | W_l^{(j)}(t), I_l)$  means that we should find a better position for the  $j$ th feature point, merely considering the local likelihood. Let  $\Gamma_{(x,y)} \subset I_l$  denote a  $5 \times 5$  image patch centered at  $(x, y)$  and  $N(W_l^{(j)})$  be a neighborhood, e.g. a  $7 \times 7$  region centered at  $W_l^{(j)}$ . We merely take into account the possible positions of  $W_l^{(j)'}$  in the neighbor  $N(W_l^{(j)})$

$$q(W_l^{(j)'}; W_l^{(j)}(t), I_l) = \frac{P(W_l^{(j)'} = (x, y) | \Gamma_{(x,y)})}{\sum_{(x,y) \in N(W_l^{(j)}(t))} P(W_l^{(j)'} = (x, y) | \Gamma_{(x,y)})}, \quad (10)$$

where  $P(W_l^{(j)'} = (x, y) | \Gamma_{(x,y)})$  is the probability of the  $j$ th feature point lying at position  $(x, y)$  given the local image pattern  $\Gamma_{(x,y)}$ . Thus it is easy to draw a new sample  $W_l^{(j)'}$  via Eqn.(10). We define a *salient map*  $p(W_l^{(j)} | I_l)$  as

$$p(W_l^{(j)} | I_l) = \frac{P(W_l^{(j)} = (x, y) | \Gamma_{(x,y)})}{\sum_{(x,y) \in I_l} P(W_l^{(j)} = (x, y) | \Gamma_{(x,y)})}, \quad (11)$$

to denote the distribution of the  $j$ th feature point at each position of image  $I_l$  according to local likelihood only. Eqn.(10) may be rewritten as

$$q(W_l^{(j)'}; W_l^{(j)}(t), I_l) = \frac{p(W_l^{(j)'} | I_l)}{\sum_{W_l^{(j)'} \in N(W_l^{(j)}(t))} p(W_l^{(j)'} | I_l)}. \quad (12)$$

Before the sampling process, we pre-compute the salient maps for all feature points such that it is very fast to draw proposals.

### 3.2 Initialization by Generalized Hough Transform

Although HDDMCMC is insensitive to initializations, good initializations always help searching algorithms both in efficiency and accuracy. In HSM, the initialization is given in the top level to initialize  $W_L(0)$  in  $I_L$ . Since the dimension of  $W_L$  is fairly high, we first give an estimate of the global parameters  $\{Z_L(0), s_L(0), \theta_L(0)\}$ , and then estimate the position of each key point.

Suppose the lattice of image  $I_L$  is  $\Psi$ . The proposal density of  $Z_L$  associated with  $s_L$  and  $\theta_L$  is

$$\begin{aligned}
q(Z_L, s_L, \theta_L | I_L) &= \sum_{j=1}^{n_L} \sum_{W_L^{(j)} \in \Psi} p(Z_L, s_L, \theta_L, W_L^{(j)} | I_L) \\
&= \sum_{j=1}^{n_L} \sum_{W_L^{(j)} \in \Psi} p(Z_L, s_L, \theta_L | W_L^{(j)}) p(W_L^{(j)} | I_L), \tag{13}
\end{aligned}$$

where the salient map  $p(W_L^{(j)} | I_L)$  generates a hypothesis of the positions of each feature point, and then the feature point would propagate the hypothesis to the 4D global parameter space by  $p(Z_L, s_L, \theta_L | W_L^{(j)})$ . This is in fact a *generalized Hough transform* (GHT). The initialization of the outer parameters  $\{Z_L(0), s_L(0), \theta_L(0)\}$  is sampled from Eqn.(13), and the key points most likely to be connected to  $Z_L(0)$  via  $s_L(0)$  and  $\theta_L(0)$  are chosen to initialize  $W_L(0)$ .

## 4 Distribution Modeling and Learning

In the previous section we introduced the statistical framework of HSM and its four distributions. Overall, in HDDMCMC, there are basically two densities, *i.e.* conditional prior  $p(W_l | W_{l+1})$  and salient map  $p(W_l^{(j)} | I_l)$  for us to draw proposals, and other two densities, prior  $p(W_l)$  and likelihood  $p(I_l | W_l)$  to evaluate the posterior. In this section we design different strategies to model them. We show that all of them can be decomposed to a mixture of Gaussians model, which could be reliably learnt by reversible jump Markov chain Monte Carlo method.

### 4.1 Prior $p(W_l)$

From prior decomposition Eqn.(4) and independence assumption Eqn.(5), we model the distribution of *external* and *internal* parameters separately. The prior distribution of the position  $Z_l$  is uniform and omitted. The priors of scale  $s_l$  and orientation  $\theta_l$  are modeled by Gaussians

$$p(s_l) = \frac{1}{\sqrt{2\pi}\sigma_{s_l}} \exp\left\{-\frac{(s - \mu_{s_l})^2}{2\sigma_{s_l}^2}\right\}, \quad p(\theta_l) = \frac{1}{\sqrt{2\pi}\sigma_{\theta_l}} \exp\left\{-\frac{(\theta - \mu_{\theta_l})^2}{2\sigma_{\theta_l}^2}\right\}, \tag{14}$$

where  $\mu_{s_l}$ ,  $\mu_{\theta_l}$  and  $\sigma_{s_l}$ ,  $\sigma_{\theta_l}$  are the means and variances of  $s_l$  and  $\theta_l$  respectively. They can be easily estimated from training samples.

To model the position, scale and orientation irrelevant shape  $p(w_l)$  is non-trivial due to its high dimensionality  $2n_l$ . Here we apply principal components analysis (PCA) to reduce the dimension and obtain the principal components  $h_l$  ( $\dim(h_l) < \dim(w_l)$ )

$$h_l = B_{w_l}^T (w_l - \mu_{w_l}), \quad W_l = B_{w_l} h_l + \mu_{w_l}, \tag{15}$$

where  $\mu_{w_l}$  is the mean of  $p(w_l)$ , and each column vector of  $B_{w_l}$  is the eigenvector of the covariance matrix of  $p(w_l)$ . Since  $h_l$  can approximate  $w_l$  very well with much lower dimension, we may learn  $p(h_l)$  rather than  $p(w_l)$ . We model  $p(h_l)$  with the Gaussian mixture



$$p(h_l) = \sum_{i=1}^{K_l} \alpha_l^{(i)} G(h_l; \mu_l^{(i)}, \Sigma_l^{(i)}), \quad (16)$$

where  $G(h_l; \mu_l^{(i)}, \Sigma_l^{(i)})$  is a Gaussian distribution with mean  $\mu_l^{(i)}$  and covariance  $\Sigma_l^{(i)}$ .  $\alpha_l^{(i)}$  is the corresponding weight such that  $\sum_{i=1}^{K_l} \alpha_i = 1$  and  $\alpha_l^{(i)} > 0, \forall i$ .  $K_l$  is the number of Gaussian kernels.

#### 4.2 Conditional Prior $p(W_l|W_{l+1})$

The conditional density  $p(W_l|W_{l+1})$  plays an essential role in HSM and HDDM-CMC because the localization of higher level  $W_{l+1}^*$  will propagate down via it. Similar to Eqn.(5) we may have

$$\begin{aligned} p(W_l|W_{l+1}) &= p(w_l, Z_l, s_l, \theta_l|w_{l+1}, Z_{l+1}, s_{l+1}, \theta_{l+1}) \\ &= p(w_l|w_{l+1})p(Z_l|Z_{l+1})p(s_l|s_{l+1})p(\theta_l|\theta_{l+1}). \end{aligned} \quad (17)$$

The conditional distributions  $p(Z_l|Z_{l+1}), p(s_l|s_{l+1}), p(\theta_l|\theta_{l+1})$  are all modeled as 1D or 2D Gaussians, *e.g.*,  $p(s_l|s_{l+1}) \propto \exp\{-(s_l - s_{l+1})^2/\lambda_{s_l}\}$  where  $\lambda_{s_l}$  scales the variance of  $s_l$ .

We, however, take a two-step approach to modeling  $p(w_l|w_{l+1})$ . We first learn the joint distribution  $p(w_l, w_{l+1})$  with a Gaussian mixture model. Then the conditional prior  $p(w_l|w_{l+1})$  has a closed form distribution, directly computed by  $p(w_l, w_{l+1})$  with parameters controlled by  $w_{l+1}$ . We again use PCA to reduce dimensions and in fact model  $p(w_l|w_{l+1})$  by  $p(h_l|h_{l+1})$ .

#### 4.3 Likelihood $p(I_l|W_l)$

To evaluate the likelihood of an image given the shape in HSM, we only need to take into account the pixels nearby each feature point. Let  $\Gamma_{W_l^{(j)}} \subset I_l$  denote a  $5 \times 5$  square patch around the  $j$ th feature point  $W_l^{(j)}$ . Then we have

$$p(I_l|W_l) = p(\Gamma_{W_l^{(1)}}, \dots, \Gamma_{W_l^{(n_l)}}). \quad (18)$$

Directly modeling the above joint distribution is difficult. This is why previous shape models (*e.g.*, [1]) assumed independent distributions, *i.e.*

$$p(\Gamma_{W_l^{(1)}}, \dots, \Gamma_{W_l^{(n_l)}}) = \prod_{j=1}^{n_l} p(\Gamma_{W_l^{(j)}}). \quad (19)$$

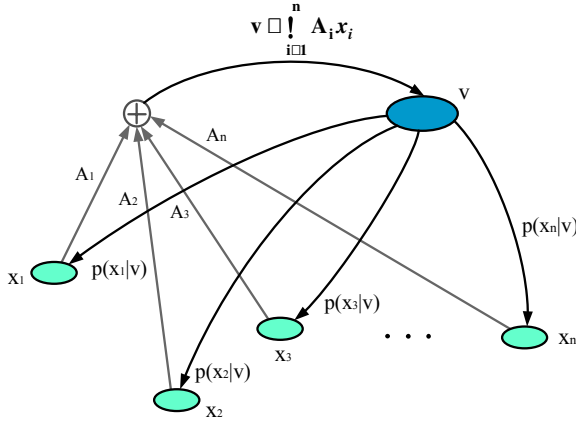
However, this assumption is an oversimplification for the likelihood. For example, what the left corner of the left eye looks like definitely depends on the appearance of the left corner of the left eyebrow.

We now introduce *global and local* (GL) distributions to model likelihood.

**Definition 1.** *The joint distribution of random variable  $X = \{x_1, \dots, x_n\}$  is called a GL distribution if*

$$p(X) = \int p(v)p(x_1, \dots, x_n|v)dv = \int p(v) \prod_{i=1}^n p(x_i|v)dv, \quad (20)$$

where  $v = f(X)$  is the hidden variable of  $X$ .



**Fig. 3.** Illustration of a GL distribution when the hidden variable is chosen as the principal components. The hidden variable  $v$  is determined by  $\{x_i\}$ , but it also controls each  $x_i$ .  $\{x_i\}$ s are never independent because  $x_j$  would affect  $x_i$  via  $v$ .

An intuitive explanation of the GL distribution is that each random variable  $x_1, \dots, x_n$  is conditionally independent with respect to the hidden variable  $v$ , and its distribution  $p(v)$  captures the global properties of  $X$ . Therefore each random variable is not independent because they are connected by the hidden variable, and meanwhile not too correlated because the conditional densities  $p(x_i|v), i = 1, \dots, n$  may be different. What we should do for GL is to select hidden variable  $v = f(X)$  and do the integration.

**Theorem 1.** Let  $v = AX$ , where  $A$  is the principal components of  $X$  and  $\dim(v) \ll \dim(X)$ . Assume  $p(v)$  and  $p(x_i|v)$  ( $i = 1, \dots, n$ ) to be continuous functions with finite optimums. The GL distribution can be approximated by

$$p(X) \approx \lambda p(AX) \prod_{i=1}^n p(x_i|AX), \quad (21)$$

where  $\lambda$  is a constant.

**Proof.** Since matrix  $A$  is the principal components of  $X$ , for a particular  $X$  and a small  $\varepsilon$  there exists a small neighborhood  $N_v(X) = \{p(X|v) > \varepsilon\}$ . Since the integration of  $p(v) \prod_{i=1}^n p(x_i|v)$  in the whole set is  $p(X) < \infty$ , the integration can be approximated in  $N_v(X)$ . The volume of the neighborhood  $N_v(X)$  exists and is assumed to be  $\delta$  due to the condition of  $p(v)$  and  $p(x_i|v)$  ( $i = 1, \dots, n$ ). According to the mid-value theorem, there exists  $\xi \in N_v(X)$  such that

$$\begin{aligned} p(X) &= \int p(v) \prod_{i=1}^n p(x_i|v) dv \\ &\approx \int_{N_v(X)} p(v) \prod_{i=1}^n p(x_i|v) dv \\ &= \delta p(\xi) \prod_{i=1}^n p(x_i|\xi). \end{aligned} \quad (22)$$

The point  $v = AX$  must lie at the center of  $N_v(x)$  because  $X \approx A^T v$  and the conditional density  $p(X|AX)$  is fairly high. Since both  $\xi$  and  $AX$  lie in the

very small neighbor  $N_v(X)$ , we may also have  $\xi \approx AX$ . This naturally leads to Eqn.(21).  $\square$

Theorem 1 gives us an approximation to evaluate the GL distribution by PCA. The hidden variable lies in the eigenspace of the observed data which captures the global correspondence as illustrated in Fig. 3. From another point of view, the distribution of the hidden variable  $p(v)$  in eigenspace approximates the observed one, and the approximation error is compensated by the local densities  $p(x_i|v)$ .

When applying a GL distribution to modeling likelihood Eqn.(18), the dimension of  $\Gamma_{W_l^{(j)}}$  is 25, still too high. We again employ PCA to reduce the dimension of  $\Gamma_{W_l^{(j)}}$  to  $u_l^{(j)}$ . And  $v_l$  is the hidden variable or principal components of  $\{u_l^{(1)}, \dots, u_l^{(n_l)}\}$ . Thus the likelihood is approximated by

$$p(I_l|W_l) \approx p(v_l) \prod_{j=1}^{n_l} p(u_l^{(j)}|v_l) = p^{-(n_l-1)}(v_l) \prod_{j=1}^{n_l} p(u_l^{(j)}, v_l). \quad (23)$$

Both  $p(v_l)$  and  $p(u_l^{(j)}, v_l)$  are assumed mixture of Gaussians.

#### 4.4 Salient Map $p(W_l^{(j)}|I_l)$

From the definition of salient map Eqn.(11), the probability  $P(W_l^{(j)} = (x, y)|\Gamma_{(x,y)})$  is essential. Based on Bayesian law we may have

$$P(W_l^{(j)} = (x, y)|\Gamma_{(x,y)}) \propto p(\Gamma_{W_l^{(j)}})p(W_l^{(j)}). \quad (24)$$

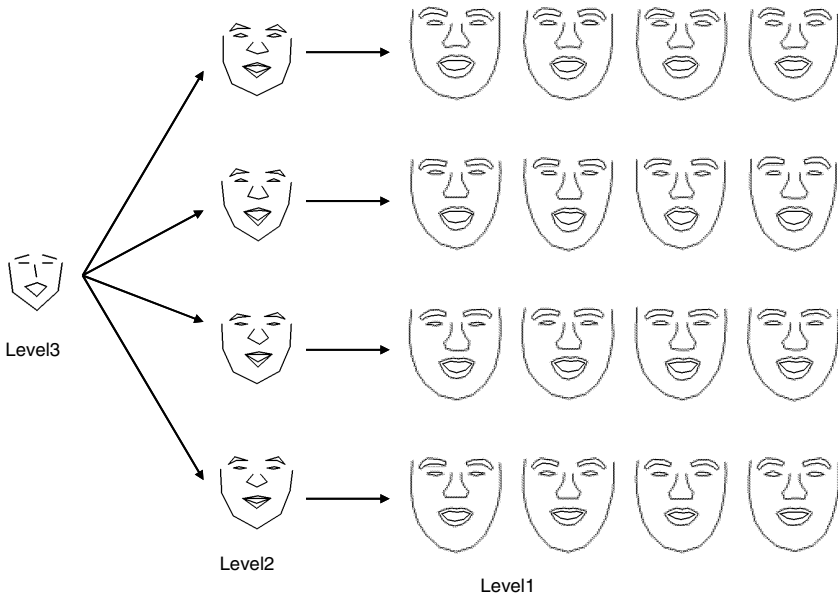
$p(W_l^{(j)})$  is the prior distribution of the  $j$ th feature point, *e.g.*, the left eye would not lie at the upper-right of the image.  $p(\Gamma_{W_l^{(j)}})$  is just the independent component of Eqn.(19). We also apply PCA to reduce  $\Gamma_{W_l^{(j)}}$  and learn a Gaussian mixture model in the reduced space.

#### 4.5 Learning Gaussian Mixture by Reversible Jump MCMC

We have so far modeled all the distributions in HSM as a mixture of Gaussians because of its flexibility in fitting arbitrary distributions. A traditional algorithm of learning Gaussian mixture model is Expectation-Maximization (EM), which needs as input the kernel number and often gets stuck in local minimums. To solve this problem, we formulate the objective function under a MAP criterion instead of MLE, with prior that restricts the number of Gaussian kernels based on the *minimum description length* (MDL) criterion. Let  $\{Y_1, \dots, Y_m\}$  be observed examples. The number of Gaussian kernels is  $k$ , and the parameter of the  $i$ th kernel is  $\alpha_i, \mu_i$  and  $\Sigma_i$ . Let  $\theta_k = \{\alpha_i, \mu_i, \Sigma_i\}_1^k$ . The Gaussian mixture model is learnt via

$$(k^*, \theta_k^*) = \arg \max_{k, \theta_k} p(k) \prod_{j=1}^m p(Y_j; \theta_k) \quad (25)$$

where  $p(k) \propto \exp\{-\lambda k \log k\}$  is the prior of the kernel number, and



**Fig. 4.** A sampling tree of face prior shape from the top to the bottom level. For each parent node  $w_{l+1}$ , four child nodes are randomly sampled from  $p(w_l|w_{l+1})$ .

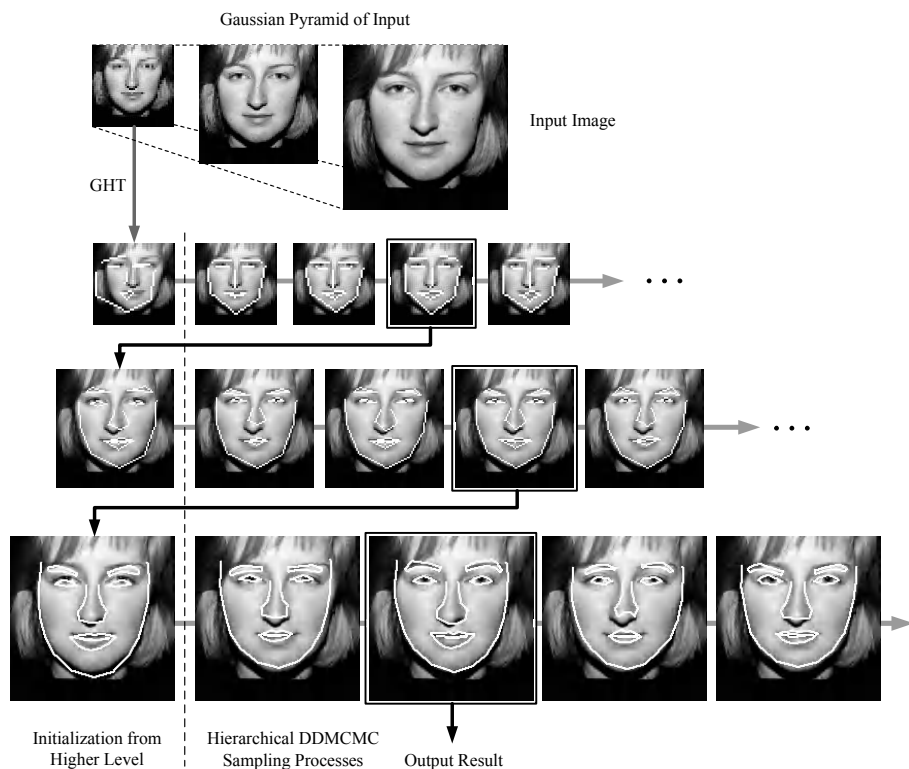
$$p(Y_i; \theta_k) = \sum_{j=1}^k \alpha_j G(Y_i; \mu_j, \Sigma_j).$$

Then a reversible jump Markov chain Monte Carlo is developed to explore varying probability spaces, with the guarantee of global convergence [8,3]. There are three processes in the reversible jump MCMC: *diffusion* to explore the same space, *split* to divide one Gaussian kernel to two, and *merge* to combine two kernels to one. So the sampler may randomly walk to samples with different kernel numbers. The learning by reversible jump MCMC is robust, efficient and reliable.

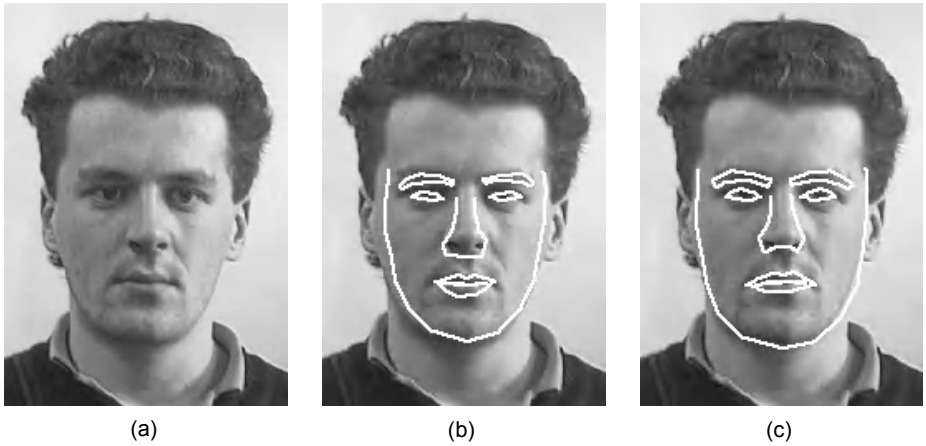
## 5 Experimental Results

Our experiments are conducted with a large number of frontal face images in the FERET data set [7], AR data set [5] and other collections, with different races and varying illuminations. We have selected 721 images as training data and use others for testing. We also collected some face images with complex backgrounds and lightings to test the robustness of our algorithm. Each training image is normalized to the same scale and manually labelled with 83 key points, including the most semantically important feature points such as the corners of eyes, mouth and face contour. These samples form the training set of shape  $\{W_1(i), i = 1, \dots, N\}$ . Then we design  $\{W_2(i), i = 1, \dots, N\}$  and  $\{W_3(i), i = 1, \dots, N\}$  with 34 and 19 feature points, respectively.

Once the three levels of shape samples and their corresponding Gaussian pyramid are generated, we employ reversible jump Markov chain Monte Carlo to learn the four elementary distributions, *i.e.*, single level prior  $p(W_l)$ , conditional prior  $p(W_l|W_{l+1})$ , likelihood  $p(I_l|W_l)$  and salient map  $p(W_l^{(j)}|I_l)$ . To justify the reliance of our learning algorithm, we build a sampling tree of hierarchical facial prior shape. The root node of this tree is the coarsest shape  $w_3$  sampled from  $p(w_3)$ . Then for each parent node in the tree, *e.g.*  $w_{l+1}(j)$ , we may get four child nodes  $\{w_{lj}(1), w_{lj}(2), w_{lj}(3), w_{lj}(4)\}$  randomly sampled from  $p(w_l|w_{l+1}(j))$ , as shown in Fig 4. The samples generated in this hierarchical shape tree demonstrate the reliability of both the conditional density modeling via Gaussian mixture and learning by reversible jump MCMC. For example, we observe that the root sample  $w_3$  seems to have a smile, and so do four child samples at level 2 and sixteen samples at level 1. Obviously the magnitude of a smile differs from level to level. Going from level 2 to level 1, we also observe that the difference between the four children at level 1 generated by the same parent node at level 2 is much smaller than that between those four nodes on layer



**Fig. 5.** The flowchart of gradually locating a face from low-resolution to high-resolution in HSM. In this display, the pyramid does not go up by 2 (the size of the image at higher level is more than a half of that at lower level), but in experiment it does.



**Fig. 6.** Comparison between likelihood assumptions. (a) Input face image. (b) Localization result with independence assumption to local features. (c) Localization result with GL distribution for local features.

2, reflecting the fact that the finer level model represents the higher frequency information.

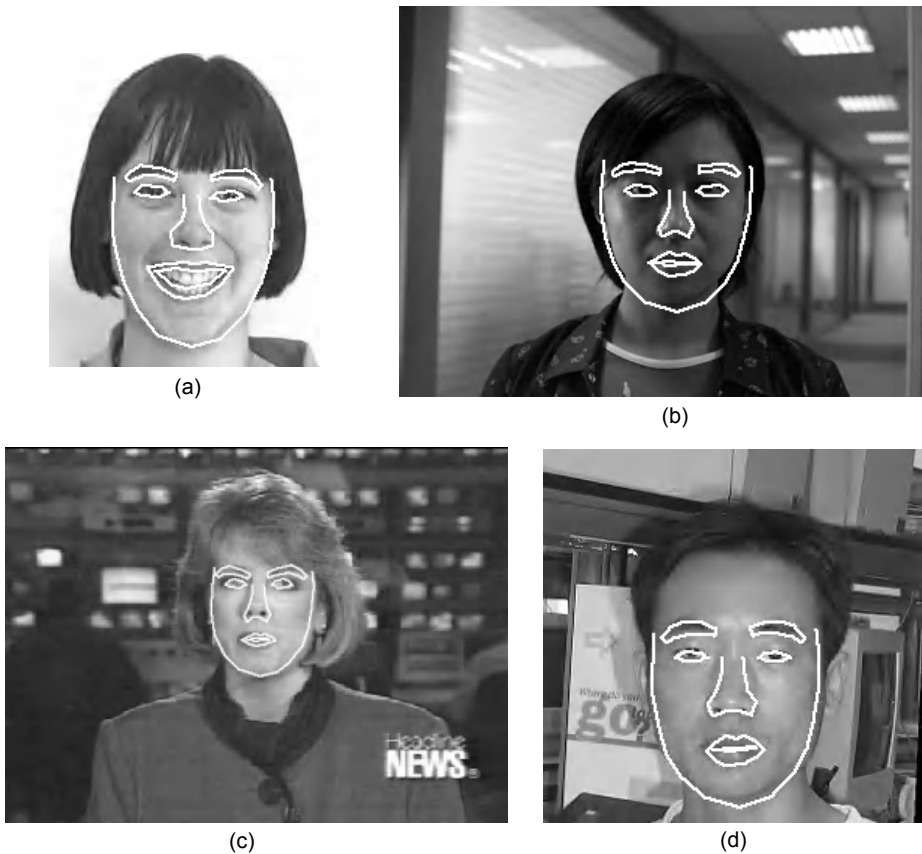
Once all the distributions are learnt (there are 280 distributions to learn), we employ HDDMCMC to locate a face in input image. We use a simple example to illustrate our algorithm in Fig. 5. First we build the Gaussian pyramid of the input image, and do the generalized Hough transform (GHT) to get the initialization in the top level. Then the sampler of HDDMCMC draws random samples from the posterior, and the optimal solution is achieved in the samples by MAP criterion. By sampling the conditional face prior densities as shown in Fig. 4, the optimal solution from higher level generates the initialization at the lower level. This process is propagated to the finest level until the global optimal solution is obtained. In each hierarchical sampling process, we have found that 2000 samples are sufficient. In our experiment, we observe that the initialization from higher level is usually close to the ground truth. Therefore, our algorithm runs very fast, taking 0.5s, 1.5s and 6s to output the face shape from coarse to fine for an image of size  $128 \times 128$ .

We designed an experiment to demonstrate the importance of the GL distribution. We select a face image with significant side illumination and run two HDDMCMC algorithms with the only difference that the local likelihood of each feature point is independent or not<sup>1</sup>. The results are listed in Fig. 6, where (b) and (c) are the results of the independent local likelihood and GL distribution, respectively. It can be observed that the face contour is localized much more accurately in (c) than in (b). The global property of the likelihood plays an

<sup>1</sup> An interesting comparison could be between HSM with GL distribution and ASM (with independent local likelihood). It is, however, fair to compare HSM with GL distribution and with independent local likelihood.

important role in face localization, which is appropriately modeled in the GL distribution.

Finally we test our algorithm on some challenging face images, shown in Fig. 7. There are typically four cases: (a) intensive expression, (b) unusual lighting condition, (c) noisy and low-quality image and (d) face very different from the training data. Overall the results are satisfactory. It is interesting to note that in (a) the bottom lip is mismatched to the bottom teeth. This is the drawback of the shape model which merely takes into account the local image patterns associated with the feature points. Note that no Asian faces are used in our training data, yet we obtain good localization results in (b) and (d). Despite the poor lighting condition in (b), our algorithm is able to generalized the learnt distributions and obtain a good localization result. Because image (c) has very low resolution, we up-sample it and still obtain good localization without false alarm.



**Fig. 7.** The results of HSM in face localization with challenging conditions. (a) Intensive expression. (b) Unusual lighting. (c) Noisy and low-quality image. (d) Appearance that is very different from the training data.

## 6 Summary

In this paper, we build a hierarchical shape model for faces and employ HDDMCMC to automatically locating a face in an image. In this way, two major problems in previous shape models, *i.e.*, huge solution space and rather inaccurate model for likelihoods, are addressed. Even though MCMC is well known for its inefficiency, the HDDMCMC runs very fast because (a) it proceeds from coarse to fine with solution space sharply reduced and (b) salient proposal densities integrating both top-down and bottom-up processes are designed to guide the Markov chain. We model the joint distribution of local likelihoods via global and local (GL) distributions to reserve the global correspondence and the local details of local features associated with the key points. Our experimental results indicate that both modeling and learning of the distributions in HSM are accurate and robust.

A large part of our work focuses on how to deal with high dimensional distributions. The key idea in our approach is to simplify a complex correspondence by introducing hidden variable. We have also found principal components analysis and reversible jump MCMC are effective in linear dimensionality reduction and density learning in HSM. In fact, the GL distribution can be applied to general vision problems.

## References

1. T.Cootes and C.Taylor. Statistical Models of Appearance for Computer Vision. Technical report, University of Manchester, 2000.
2. T.Cootes and C.Taylor. Constrained Active Appearance Models. *In Proceedings of the 8th ICCV*, July, 2001.
3. P.Green. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, vol. 82, pp. 711-732, 1995.
4. T.Leung, M.Burl, and P.Perona. Finding Faces in Cluttered Scenes using Random Labeled Graph Matching. *In Proceedings of the 5th ICCV*, June, 1995.
5. A.Martinez and R.Benavente. The AR Face Database. CVC Technical report, No. 24, June 1998.
6. E.Osuna, R.Freund, and F.Girosi. Training Support Vector Machine: An Application To Face Detection. *In Proceedings of CVPR'97*, pages 130-136, 1997.
7. P.Philips, H.Moon, P.Pauss, and S.Rivzvi. The FERET Evaluation Methodology for Face Recognition Algorithms. *In Proceedings of CVPR'97*, pp.137-143, 1997.
8. S.Roberts, C.Holmes, and D.Denison. Minimum-Entropy Data Partitioning Using Reversible Jump Markov Chain Monte Carlo. *IEEE Transactions on PAMI*, 23(8):909-914, August, 2001.
9. H.Rowley, S.Baluja, and T.Kanade. Neural Network-Based Face Detection. *IEEE Transactions on PAMI*, 20(1), January 1998.
10. H.Schneiderman and T.Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. *In Proceedings of the 7th ICCV*, May, 2000.
11. K.Sung and T.Poggio. Example-based Learning for View-based Human Face Detection. *IEEE Transactions on PAMI*, 20(1):39-51, 1998.
12. Z.Tu and S.Zhu. Image Segmentation by Data Driven Markov Chain Monte Carlo. *In Proceedings of the 8th ICCV*, July, 2001.



13. M.Turk and A.Pentland. Eigenface for Recognition. *Journal of Cognitive Neurosciences*, pages 71-86, 1991.
14. P.Viola and M.Jones. Robust Real-time Face Detection. *In Proceedings of the 8th ICCV*, July, 2001.
15. A.Yuille, P.Hallinan, and D.Cohen. Feature Extraction from Faces using Deformable Templates. *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99-111, 1992.

# Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data

Slobodan Ilic and Pascal Fua\*

Computer Graphics Laboratory(LIG)  
EPFL, 1015 Lausanne, Switzerland  
{Slobodan.Ilic, Pascal.Fua}@epfl.ch

**Abstract.** Free-form deformations (FFD) constitute an important geometric shape modification method that has been extensively investigated for computer animation and geometric modelling. In this work, we show that FFDs are also very effective to fit deformable models to the kind of noisy 3-D data that vision algorithms such as stereo tend to produce.

We advocate the use of Dirichlet Free Form Deformation (DFFD) instead of more conventional FFDs because they give us the ability to place control points at arbitrary locations rather than on a regular lattice, and thus much greater flexibility. We tested our approach on stereo data acquired from monocular video-sequences and show that it can be successfully used to reconstruct a complex object such as the whole head, including the neck and the ears, as opposed to the face only.

## 1 Introduction

Free-form deformations (FFDs) constitute an important approach to geometric shape modification that has been extensively investigated for computer animation and geometric modelling [15,3,9,2,13]. In the vision community, they have also been used to fit parametric models to medical data [17,1] or animation masks to semi-automatically extracted silhouette data [12]. In this work, we show that FFDs also are also very effective to fit deformable surface models to the kind of noisy 3-D data that vision algorithms such as stereo tend to produce.

The initial FFD approach [15] and all subsequent ones involve embedding the object model into a volume whose shape can be changed by moving a number of control points. Here, we take the embedded object to be a triangulated mesh and the embedding guarantees that each model vertex is influenced by the deformation of the control grid.

In this work, we advocate the use of Dirichlet Free Form Deformations (DFFDs) [13] because, unlike more conventional FFDs, they do not require the control points to lay on a regular rectangular grid. This is achieved by replacing the typical rectangular local coordinates by generalized natural neighbor coordinates, also known as Sibson coordinates[16]. That property give us the ability to place control points at arbitrary locations—that is, on the object, inside of it or outside—rather than on a regular lattice, and thus much greater flexibility. In particular, some of the control points can be important feature points that must be controlled in a specific way.

---

\* This work was supported in part by the Swiss National Science Foundation under contract 21-57075.99

This flexibility of DFFDs has been put to good use in earlier face-modelling work [12]. That approach, however, takes silhouettes extracted from orthogonal images as input and does not allow for potential errors in the data. By contrast, we expect our input data to be both noisy and possibly incomplete. We have therefore developed a least-squares adjustment framework that lets us impose appropriate regularization constraints on the control mesh and obtain good fitting results even when using poor quality data.

We chose to demonstrate and evaluate our technique mainly in the context of head-modelling: More specifically, we use a technique we developed in earlier work [7] to compute motion parameters for heads seen in uncalibrated video-sequences and compute disparity maps from consecutive image pairs. We derive 3-D point clouds from these maps and fit a complete head model, including face, ears and neck, to them. Given the relative absence of texture one typically finds on faces and the uncalibrated nature of the sequences, the point clouds cannot be expected to be error-free and, yet, we obtain realistic models, whose geometry is correct and for which we can compute good texture maps by averaging the gray levels in all the images of a sequence.

We concentrate on head-modelling to demonstrate our technique because it is the application for which all the tools required to perform the complete reconstruction are available to us. We will, however, argue that the approach is generic and can be applied to any task for which deformable facetized models exist. In particular, we will show that we can also use our approach for high-resolution modelling of the human ear. We therefore view our contribution as the integration of a powerful shape deformation tool into a robust least-squares framework.

In the remainder of this paper, we first describe DFFDs in more detail. We then introduce our least-squares optimization framework and, finally, present results using both calibrated and uncalibrated image sequences.

## 2 Dirichlet Free Form Deformation(DFFD)

FFDs and their many extensions, such as rational FFD, extended FFD, direct FFD, NURBS based FFD and many others [15,3,2,8,11], are well known in the Computer Graphics community. They are used to globally deform a shape by embedding it into a volume controlled by a small number of control points. The volume's shape is then deformed by moving the control points and the motion of the shape points is obtained by interpolating that of the control points.

In the original FFD, the control points must be placed on a regular lattice, which severely limits the range of allowable deformations that can be modeled. Most FFD extensions aim at overcoming this limitation, often by using a more sophisticated interpolant, but without addressing the basic problem that there is little flexibility in the positioning of the control points. By contrast, DFFDs [13] remove the requirement for regularly spaced control points that is the main conceptual geometric limitation of FFDs by replacing rectangular local coordinates by generalized natural neighbor coordinates, also known as Sibson coordinates, and using a generalized interpolant [4]. This idea comes from the data visualization community that relies on data interpolation and, thus, heavily depends on local coordinates.

## 2.1 Sibson Coordinates

All FFDs can be viewed as a data interpolation scheme where the interpolating function is a 3-D function specifying point displacements. The displacement is known at the control points and we want to interpolate it to the points of the object to deform. To this end, DFFDs replace standard rectangular or barycentric coordinates that constrain the control grid's shape by Sibson coordinates [16]: Given a set of points  $P = \{P_0, P_1, \dots, P_n\}$ , these coordinates let us express any point within the convex hull of  $P$  as a linear combination of its *natural neighbors* within  $P$ . To find them, we compute a 3-D Delaunay triangulation and its Voronoi dual in which the cells are polyhedra.

A given point  $p$  may sit inside several several Delaunay spheres—that is, spheres circumscribed around Delaunay tetrahedra—and the vertices of all these tetrahedra are taken to be the so-called natural neighbors. In other words,  $p$  is inside the sphere of influence of its neighbours. Let this set of natural neighbors be  $Q_k \subset P$ , where  $k$  is number of points influencing  $p$ . The elements of  $Q_k$  are the natural neighbors of  $p$  and their influence is expressed by the Sibson coordinates  $u_i$  such that:

$$p = \sum_{i=0}^k u_i P_i, P_i \in Q_k$$

with  $\sum_{i=0}^k u_i = 1$  and  $u_i > 0$ .

These coordinates are “natural” in the sense that points in  $P$  that are closer to the point  $p$  have greater influence on it, that is, the corresponding  $u_i$  is larger. They are computed using the partitioning of the space induced by the Voronoi diagram which is the dual of the Delaunay triangulation.

## 2.2 Introducing Deformations

As will be discussed above, our goal is to deform a *surface triangulation* using the vertices of a much sparser *control triangulation*, as our control points. We therefore take the control points to the vertices of the control triangulation complemented by the corners of the surface triangulation's bounding box, so as to guarantee that the whole object is contained in their convex hull.

Let  $P$  be this set of all control points and  $P_o$  the set of vertices of the surface triangulation. For each point  $p \in P_o$ , we find its natural neighbors  $Q_k \subset P$  and the corresponding Sibson coordinates. This is referred to as freezing the control mesh to the object. Once computed, Sibson coordinates do not need to be changed when the object is deformed. When we move some of the control points from the set  $Q_k$ , the displacement of the model points is computed as follows:

$$\Delta p_o = \sum_{i=0}^k \Delta P_i u_i \quad (1)$$

where  $\Delta P_i$  is displacements of control point from  $P_i \in Q_k, i = 0, \dots, k$ . Finally, new object point position is computed as:

$$p'_o = p_o + \Delta p_o, \quad (2)$$

In short, the deformations are local and defined by the natural neighbors, which helps to improve the flexibility of the approach and the realism in the final results.

### 3 Least Squares Framework for DFFD Fitting

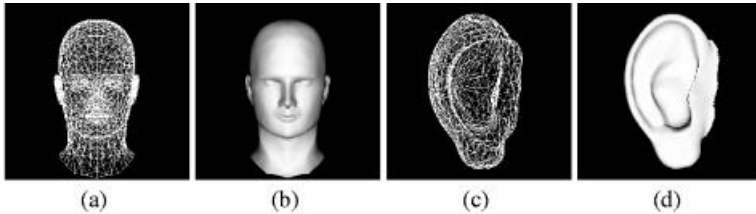
In this section, we introduce the framework we have developed to fit *surface models* such as the ones of Fig. 1 to noisy image data. Our goal is to deform the surface—without changing its topology, that is the connectivity of its vertices—so that it conforms to the image data. In this work data is made of 3-D points computed using stereo. In standard least-squares fashion, for each point  $\mathbf{x}_i$ , we write an observation equation of the form  $d(\mathbf{x}_i, S) = obs_i + \epsilon_i$ , where  $S$  is a state vector that defines the shape of the surface,  $d$  is the distance from the point to the surface and  $\epsilon_i$  is the deviation from the model. In practice  $d(\mathbf{x}, S)$  is taken to be the orthonormal distance of  $\mathbf{x}$  to the closest surface triangulation facet. This results in  $nobs$  such observations forming a vector

$$F(S) = [..., d(\mathbf{x}_i, S) - obs_i, ...]_{1 \leq i \leq nobs}^t \quad (3)$$

that we minimize in the least squares sense by minimizing its square norm

$$\chi^2 = 1/2 \|F(S)\|^2 .$$

In theory we could take the parameter vector  $S$  to be the vector of all  $x, y$ , and  $z$  coordinates of the surface triangulation. However, because the image data is very noisy, we would have to impose a very strong regularization constraint. For example, we have tried to treat the surface triangulation as finite element mesh. Due to its great irregularity and its large number of vertices, we have found the fitting process to be very brittle and the smoothing coefficients difficult to adjust. This is why we chose to use the DFFD deformation approach instead.

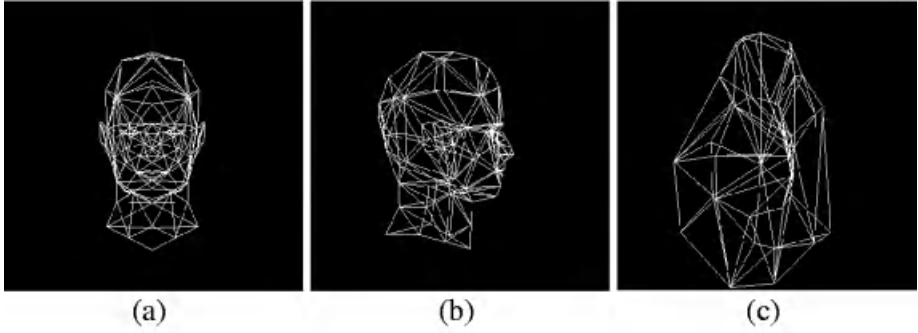


**Fig. 1.** (a,b) Complete head and neck animation model, shown as a wireframe and a shaded surface. (c,d) generic ear model

#### 3.1 DFFD Parametrization

We therefore introduce *control triangulations* such as the ones of Fig. 2. Their vertices are points located at characteristic places on the human head or ear and defining their rough shapes and serve as DFFD control points. Some of these control points also are vertices of the surface model, while other are simply close to it and either inside or outside of it. This ability to place the control points is unique to DFFDs as compared to all other kinds of FFDs. The control triangulation facets will be used to introduce the regularization constraint discussed below. We tried to use several levels of resolutions

of control meshes, but we found that increasing the number of control points does not influence final results of the deformation. This means that keeping low number of the control points, as we did, greatly saves time for computation.



**Fig. 2.** Control triangulations for head (face (a) and profile (b)) and ear (c).

In our scheme, we take the state vector  $S$  to be the vector of 3-D displacements of the DFFD control points, which is very natural using the DFFD formalism: As discussed in Section 2.2, we first freeze the control mesh to the model vertices. This means that for each vertex on the model we compute the influence of certain subset of control points. Those influences are expressed in terms of the Sibson coordinates from Section 2.1 and allows us to express displacement of every model vertex as the linear combination of displacements of control points which influence them.

### 3.2 3D Observations

We use several sets of stereo pairs or triplets from the sequence of images of a given object as our input data such as those of Fig. 3(a,b,c). We then ran a simple correlation-based algorithm [5] to compute a disparity map for each pair or triplet and by turning each valid disparity value into a 3-D point. This resulted in a large cloud of 3-D points that form an extremely noisy and irregular sampling of the underlying global 3-D surface. To reduce the size of the cloud and begin eliminating outliers, we robustly fitted local surface patches to the raw 3-D points [6]. We then fed the centers of those patches, shown in Fig. 3(d), as input to our surface fitting algorithm.

The center of each patch can then be treated as an attractor. The easiest way to handle this is to model it as a spring attached to the mesh vertex closest to it. This, however, is inadequate if one wishes to use facets that are large enough so that attracting the vertices, as opposed to the surface point closest to the attractor, would cause unwarranted deformations of the mesh. This is especially important when using a sparse set of attractors. In our implementation, this is achieved by writing the observation equation as:

$$d_i^a = 0 + \epsilon_i \quad (4)$$

where  $d_i^a$  is the orthogonal distance of the attractor to the closest facet, whose nominal value is zero. It can be computed as a function of the  $x$ ,  $y$ , and  $z$  coordinates of the vertices of the facet closest to the attractor.

Because some of the observations, derived on the way explained above, may be spurious, we weigh them to eliminate outliers. Weighting is done as the preprocessing step, before the real fitting is started. In each iteration after fitting is done, we recompute the attachments and also recompute the observation weight  $w_i$  and take it to be inversely proportional to the initial distance  $d_i$  of the data point to the surface triangulation. More specifically we compute  $w_i$  weight of the  $obs_i$  as:

$$w_i = \exp\left(\frac{d_i}{\bar{d}_i}\right), 1 \leq i \leq n \quad (5)$$

where  $\bar{d}_i$  is the median value of the  $d_i$ . In effect, we use  $\bar{d}_i$  as an estimate of the noise variance and we discount the influence of points that are more than a few standard deviations away.

### 3.3 2D Observations

In our optimization framework besides 3D stereo observations we introduce also 2D observations. For each vertex  $(x_i, y_i, z_i)$  of the *surface triangulation* whose 2D projection in image  $j$  is  $(u_i^j, v_i^j)$  is known, we can write two observation equations:

$$P_{ru}(x_i, y_i, z_i) = u_i^j + \epsilon_i^u$$

$$P_{rv}(x_i, y_i, z_i) = v_i^j + \epsilon_i^v$$

where  $P_{ru}$  and  $P_{rv}$  stand for the projection in  $u$  and  $v$ . In this way we do not need the explicit 3D position of these feature points, only their 2D image location.

### 3.4 Regularization

Because there are both noise and potential gaps in the image data, we found it necessary to introduce a regularization term comparable the one proposed in [1]. Since we start with a generic model, we expect the deformation between the initial shape and the original one to be smooth. This can be effectively enforced by preventing deformations at neighboring vertices of the control mesh to be too different. If the control points formed a continuous surface parametrized in terms of two parameters  $u$  and  $v$ , a natural choice would therefore be to take this term to be

$$\mathcal{E}_D = \sum_{s \in x, y, z} \mathcal{E}_{Ds} \quad (6)$$

$$\mathcal{E}_{Ds} = \iint \left( \frac{\partial}{\partial u} \delta_s(u, v) \right)^2 + \left( \frac{\partial}{\partial v} \delta_s(u, v) \right)^2 du dv ,$$

where  $\delta_s(u, v)$  stands for the displacement along the  $x, y$  or  $z$  coordinates at each point of this control surface. In fact, the control surface is a triangulated one and we only have deformation values at the vertices. We are therefore use a finite element approach to compute  $\mathcal{E}_{Ds}$ .

**Stiffness Matrix for  $C^0$  Triangular Elements.** We write the  $\mathcal{E}_{D_s}$  term of Eq. 6 as

$$\mathcal{E}_{D_s} = \lambda/2 \sum_{1 \leq j \leq n} \mathcal{E}_{D_s}^j$$

where  $\mathcal{E}_{D_s}^j$  represents a summation over a facet  $j$  and  $\lambda$  is a regularization coefficient. In fact, we only know the deformations  $s_1^j, s_2^j$  and  $s_3^j$  at the vertices of the facet and we treat it as a  $C^0$  triangular element. Over this specific facet, we write

$$\delta_s^j(u, v) = (1 - u - v)s_1^j + us_2^j + vs_3^j \quad (7)$$

where  $u, v \in [0, 1]$  and  $u + v < 1$ . It is then easy to show that  $\mathcal{E}_{D_s}^j$  is the quadratic term

$$[s_1 s_2 s_3] K_s^j \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix},$$

where  $K_s^j$  is a  $3 \times 3$  symmetric matrix that only depends on the shape of the triangle and, therefore, does not change during the optimization. These matrices can be summed into a global *stiffness matrix*  $K_s$  so that  $\mathcal{E}_{D_s}$  becomes

$$s^t K_s s$$

where  $s$  stands for the vector of displacements at each vertex in one of the three coordinates. By summing these three terms, we obtain the final quadratic form or our complete regularization term

$$\mathcal{E}_D(S) = \frac{\lambda}{2} S^t K S \quad (8)$$

where  $S$  is the complete state vector.

**Incorporating the Stiffness Matrix into the Least-Squares Framework.** We use the Levenberg-Marquardt algorithm [14] to iteratively minimize the square-norm of the observation vector  $F(S)$  of Eq. 3. At each iteration, given the current state  $S$ , the algorithm attempts to find a step  $dS$  that minimizes

$$\chi^2(S + dS) = 1/2 \|F(S + dS)\|^2 = 1/2 F(S + dS)^t F(S + dS) . \quad (9)$$

At the minimum, we should have

$$\begin{aligned} 0 &= \frac{\partial \chi^2}{\partial dS} \\ &= A^t F(S + dS) \\ &\approx A^t (F(S) + AdS) , \end{aligned}$$

where  $A$  is the jacobian of  $F$ .  $dS$  is therefore taken to be the solution of

$$A^t AdS = -AF(S) . \quad (10)$$



Adding a regularization term means that instead of minimizing simply the  $\chi^2$  term of Eq. 9, we minimize

$$\chi^2(S) + \mathcal{E}_D(S) = \frac{1}{2} \|F(S + dS)\|^2 + \frac{\lambda}{2} (S + dS)^t K (S + dS) . \quad (11)$$

At each iteration, we therefore solve

$$\begin{aligned} 0 &= \frac{\partial \chi^2}{\partial dS} + \lambda K (S + dS) \\ &\approx A^t (F(S) + AdS) + \lambda K (S + dS) . \end{aligned}$$

$dS$  therefore becomes the solution of

$$(A^t A + \lambda K) dS = -A^t F(S) - \lambda K S . \quad (12)$$

Note that solving Eq. 10 or 12 involves the same amount of computation so that our regularization scheme adds very little to the computational complexity of the algorithm. Note also that the proposed optimization scheme is a semi-implicit one very similar to the one proposed for the original active contours [10] and that greatly improves the convergence properties of the algorithm.

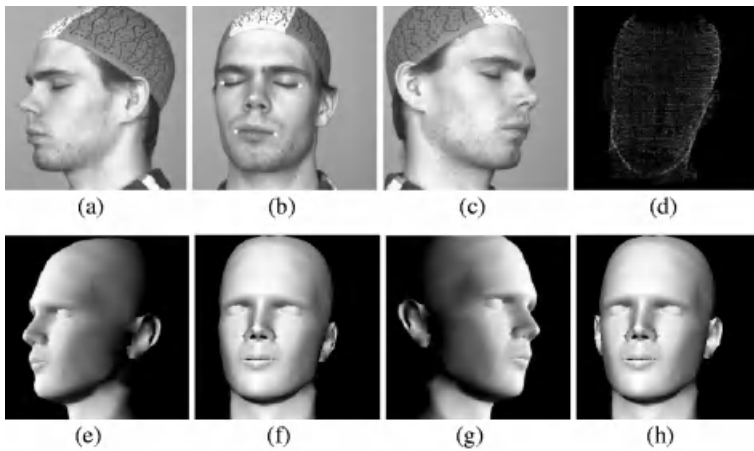
## 4 Results

We demonstrate and evaluate our technique mainly in the context of complete head, that is including face, ears and neck, from calibrated and uncalibrated video sequences. We show its flexibility by also using it to fit a generic ear model using a stereo-pair.

### 4.1 Calibrated Video Sequence

We first illustrate the effectiveness of our approach using relatively clean stereo data. We use the sequence of forty 512x512 images where some are depicted by the first row of Fig. 3. They were acquired with a video camera over a period of a few seconds by turning around the subject who was trying to stand still. Camera models were later computed using standard photogrammetric techniques at the Institute for Geodesy and Photogrammetry, ETH-Zürich. The centers of the local surface patches fitted to the raw stereo data shown on Fig. 3(d) are used as an input to our surface fitting algorithm.

We initialized the model by manually picking the five 2-D points overlaid in Fig. 3(b). We used them to compute a 4x4 rotation-translation matrix  $Rt$  such that five specific 3-D points on the generic model—outside corners of the eyes, corners of the mouth and tip of the nose—once multiplied by this matrix project as close as possible to the hand-picked 2-D location. Because these points are not coplanar, this guarantees that, when we multiply the generic model by this  $Rt$  matrix, we obtain an initial head model that is roughly aligned with the point cloud. We use it as the surface model that we deform using our DFFD approach, yielding the results shown in Fig. 3(e,f,g). In this case we did not use additional 2D observations provided manually. The resulting shape corresponds to the real head shape except around the ears that stick out more from the real head than from the reconstructed model. The reason for this behavior is because of well



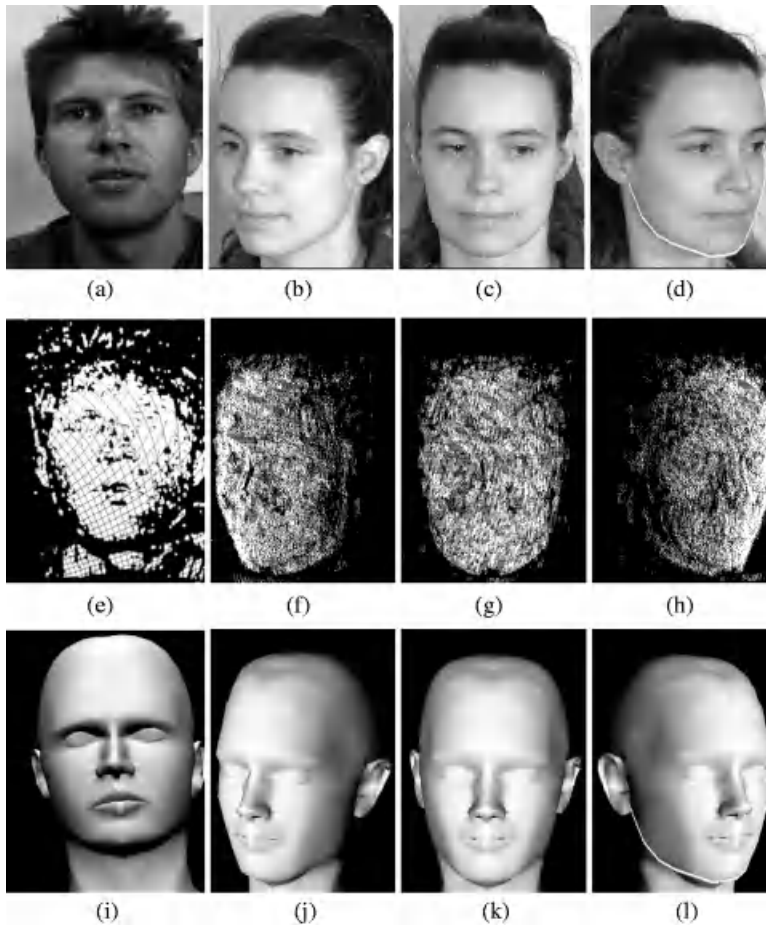
**Fig. 3.** Calibrated video sequence: (a,b,c) Three images chosen from the calibrated video sequence, courtesy of IGP, ETH Zürich. (d) Centers of local surface patches fitted to the raw stereo data. (e,f,g) Automatically obtained shaded model after fitting to the stereo data and projected using the same perspective transform as that of the images. (h) Shaded model after interactive correction of the ears.

known fact that minimizing orthonormal distance of data points to the closest surface triangulation facet may have difficulty in deforming the model into concave objects [18]. More accurate deformation can be obtained when 2D projected observation are included in the objective function what is used in the examples of uncalibrated video sequence Fig. 4. One of the advantages of DFFD is that this can be fixed manually very quickly and very intuitively by moving one control point per ear to produce the model of Fig. 3(h).

## 4.2 Uncalibrated Video Sequence

Fig. 4 depicts two examples of reconstruction from uncalibrated images. First row shows images from different image sequences: one image from the stereo pair of images Fig. 4(a) and three frames from other uncalibrated video sequence. In both cases, we had no calibration information about the camera or its motion. We therefore used a model-driven bundle-adjustment technique [7] to compute the relative motion and, thus, register the images. We then used the same technique as before [6] to derive the clouds of 3-D points depicted by Fig. 4(e,f,g,h).

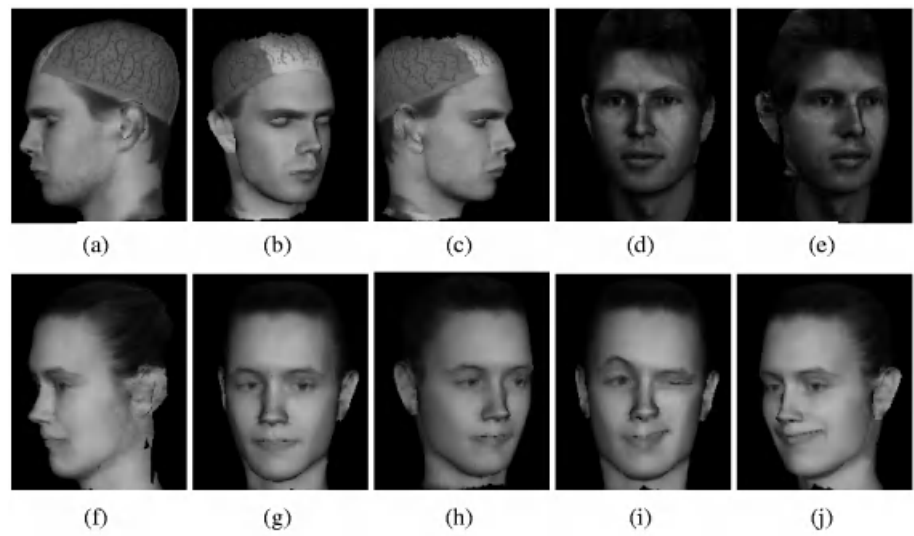
Because we used fewer images and an automated self-calibration procedure as opposed to a sophisticated manual one, the resulting cloud of 3-D points is much noisier and harder to fit. Shaded models obtained after the fitting are depicted on Fig. 4(i,j,k,l). Notice they shaded models are shown in the same projection as the corresponding images on Fig. 4(a,b,c,d). In Fig. 4(d) and (l) we overlay on both the original image and the shaded projection of the mask the outlines of the face. Note that they correspond to the outlines predicted from the recovered 3-D geometry, thus indicating a good fit. These models can also be reused to resynthesize textured images such as the ones of Fig. 5.



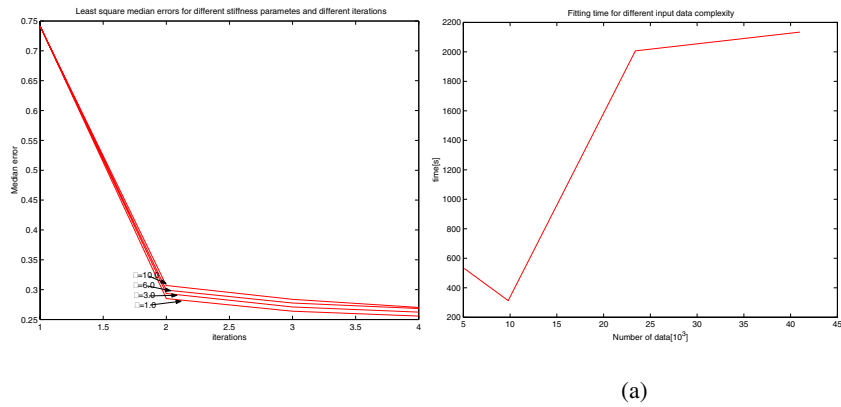
**Fig. 4.** Uncalibrated video sequence: (a) One image from the stereo pair of images with overlaid manually provided 2D observations. (b,c,d) Three images from the uncalibrated video sequence out of eight images. On the image (c) 2D observations are despicted. (d) The image with overlaid face outline. (e,f,g,h) Centers of local surface patches fitted to the raw stereo data. (i,j,k,l) Automatically recovered shaded head model projected using the (a),(b),(c) and (d) image camera models. (l) Face outlines overlaid on the shaded projection in image (d).

We also animated generated models after the automatic fitting procedure using DFFD and produced complex facial expressions Fig. 5(i,j).

Least-square adjustment is applied in several iterations for the same value of regularization parameter  $\lambda$ . However, we tested how fitting to the uncalibrated data Fig. 4(b,c,d) is influenced by different choice of regularization parameter, in our case ranging from  $\lambda = 1.0$  to  $\lambda = 10$ , and checked median error of the model to observations distance for certain number of iterations what is depicted on the Fig. 6(a). It is easy to see that final results do not depend on the choice of the regularization parameter since the median



**Fig. 5.** Textured models we created from the calibrated video sequence:(a,b,c), from stereo pair (d,e) and from uncalibrated video sequence:(f,g,h); Animated model showing complex facial expressions(i,j)

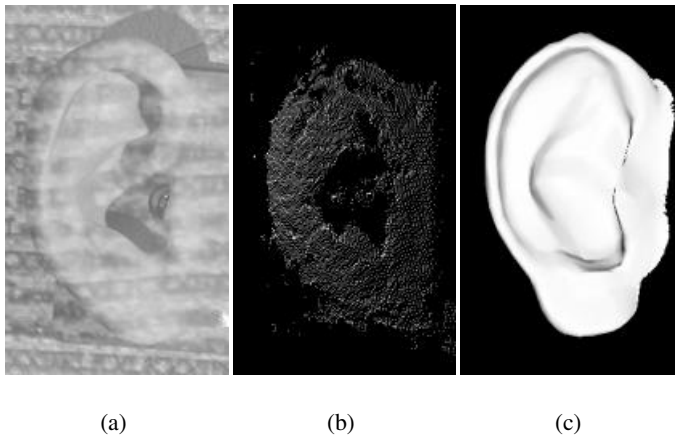


**Fig. 6.** (a) Least square median error for different regularization parameter  $\lambda$  ranging from 1.0 to 10.0 in respect to the number of iterations for video sequence whose three images are Fig. 4(b,c,d). (b) Fitting time in respect to the input data complexity

error of the model to observations distance does not greatly change with the increasing of the regularization parameter.

### 4.3 Ear Modelling

Finally, we will show that we are also able to model some other complex geometrical shapes such as human ear. The model we have is of very bad quality and we got it from the web Fig. 1(c,d). Data are produced from two images taken with the structure light Fig. 7(a). The control mesh is created manually Fig. 2(c), but this time there are no control points which exactly match some of the points on the model. Once again we demonstrated the power of DFFD based fitting to the noisy data. Results we obtained are again realistic and reliable. Deformed model on Fig. 7(c) is close to the ear shown on the Fig. 7(a). Using such complex shape to deform we can argue that the approach used is generic enough that it could be applied to any shape for which we have triangulated model.



**Fig. 7.** (a) Ear image “Courtesy of U. Maryland”, (b) data cloud, (c) automatically obtained shaded deformed model shown in the same perspective as the ear on the image

### 4.4 Performance Measures

In our framework we use generic model of the human head including ears and neck from Fig. 1(a,b) which consists of 1396 vertices and 2756 facets. This generic model is used in all tests we performed to model heads. Control mesh is the one from Fig. 2(a, b). Generic ear model from Fig. 1(c,d) is built of 1167 vertices and 1620 facets, while its corresponding control mesh is shown on Fig. 2(c) has 46 vertices and 87 facets. System is tested on ancient Silicon Graphic Octane work station with R12000 processor working on 300MHz, and with 512Mb RAM memory.

The process starts with freezing the control mesh to the surface triangulation computing necessary Sibson coordinates. This is done once at the very beginning and it is used for all input data. Freezing the control mesh of the head takes 65s. On the Fig. 6(b) is shown how the time for fitting depends on complexity of input data. Fitting time increases with the complexity of the data, but also depends on its configuration. For this test the number of fitting steps is fixed to three iterations and the stiffness parameter is set to  $\lambda = 1.0$ . Input data of the size  $10^3$ , are fitted for the shorter time than the one of the lower complexity  $5 \cdot 10^3$ , since its configuration is initially closer to the generic model, so the least-square minimization converges faster.

## 5 Conclusion

In this work, we proposed to use the powerful DFFD extension to the conventional FFD shape deformation approach to fit deformable surface models to noisy 3-D image data. DFFDs give us the ability to place control points at arbitrary locations rather than on a regular lattice, and thus much greater flexibility. We demonstrated the effectiveness and robustness of our technique in the context of complete head modeling. We also showed that, in fact, we can model any complex shape for which a deformable triangulated model exists. For the specific application we chose, DFFDs offer the added benefit that they can be used to animate the head models we create and to produce realistic human expressions. The proposed framework could thus be extended to head motion tracking.

In future work we intend to investigate the possibility of using DFFDs to deform implicit surfaces as opposed to the explicit ones we use here. In unrelated body-modelling work, we found that implicit surface formulations lend themselves extremely well to fitting to the kind of data because they allow us to define a distance function of data points to models that is both differentiable and computable without search. Combining both approaches should therefore produce an even more powerful modelling tool.

## References

1. E. Bardinet, L.D. Cohen, and N. Ayache. A Parametric Deformable Model to Fit Unstructured 3D Data. *Computer Vision and Image Understanding*, 71(1):39–54, 1998.
2. Y. Chang and A. P. Rockwood. A Generalized de Casteljau Approach to 3D Free-form Deformation. *Computer Graphics, SIGGRAPH Proceedings*, pages 257–260, 1994.
3. S. Coquillart. Extended Free-Form Deformation: A sculpturing Tool for 3D Geometric Modeling. *Computer Graphics, SIGGRAPH Proceedings*, 24(4):187–196, 1990.
4. G. Farin. Surface over Dirichlet Tessellations. *Computer Aided Design*, 7:281–292, 1990.
5. P. Fua. A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features. *Machine Vision and Applications*, 6(1):35–49, Winter 1993.
6. P. Fua. From Multiple Stereo Views to Multiple 3-D Surfaces. *International Journal of Computer Vision*, 24(1):19–35, August 1997.
7. P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2):153–171, July 2000.
8. W.M. Hsu, J.F. Hugues, and H. Kaufman. Direct manipulation of Free-Form Deformations. *SIGGRAPH*, 26(2):177–184, 1992.
9. P. Kalra, A. Mangili, N. Magnenat Thalmann, and D. Thalmann. Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. In *Eurographics*, 1992.

10. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
11. H.J. Lamousin and W.N. Waggenspack. NURBS-based Free-Form Deformations. *Computer Graphics and Applications*, 16(14):59–65, 1994.
12. W.S. Lee and N. Magnenat Thalmann. Fast Head Modelling for Animation. *Journal Image and Vision Computing*, 18(4), August 2000.
13. L. Moccozet and N. Magnenat-Thalmann. Dirichlet Free-Form Deformation and their Application to Hand Simulation. In *Computer Animation*, 1997.
14. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes, the Art of Scientific Computing*. Cambridge U. Press, Cambridge, MA, 1986.
15. T.W. Sederberg and S.R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics, SIGGRAPH Proceedings*, 20(4), 1986.
16. R. Sibson. A vector identity for the Dirichlet Tessellation. In *Math. Proc. Cambridge Philos. Soc.*, pages 151–155, 1980.
17. R. Szeliski, S. Laval, and e. Matching anatomical surface with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.
18. C. Xu and J. Prince. Snakes, Shapes, and Gradient Vector Flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998.

# Transitions of the 3D Medial Axis under a One-Parameter Family of Deformations

Peter Giblin<sup>1</sup> and Benjamin B. Kimia<sup>2</sup>

<sup>1</sup> Department of Mathematical Sciences,  
The University of Liverpool, Liverpool, England L69 3BX  
[pjgiblin@liv.ac.uk](mailto:pjgiblin@liv.ac.uk)

<sup>2</sup> Division of Engineering, Brown University,  
Providence, RI, USA  
[kimia@lems.brown.edu](mailto:kimia@lems.brown.edu)

**Abstract.** The instabilities of the medial axis of a shape under deformations have long been recognized as a major obstacle to its use in recognition and other applications. These instabilities, or *transitions*, occur when the structure of the medial axis graph changes abruptly under deformations of shape. The recent classification of these transitions in 2D for the medial axis and for the shock graph, was a key factor both in the development of an object recognition system and an approach to perceptual organization. This paper classifies generic transitions of the 3D medial axis, by examining the order of contact of spheres with the surface, leading to an enumeration of possible transitions, which are then examined on a case by case basis. Some cases are ruled out as never occurring in any family of deformations, while others are shown to be non-generic in a one-parameter family of deformations. Finally, the remaining cases are shown to be viable by developing a specific example for each. We relate these transitions to a classification by Bogaevsky of singularities of the viscosity solutions of the Hamilton-Jacobi equation. We believe that the classification of these transitions is vital to the successful regularization of the medial axis and its use in real applications.

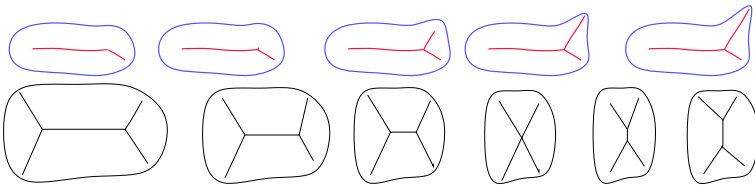
## 1 Introduction

The practical use of the medial axis in visual tasks such as object recognition, perceptual grouping, shape modeling and tracking, etc. is adversely affected by the frequent and omni-present instabilities of the medial axis under deformations of shape. Previous approaches have either embedded an implicit regularization in the detection process [11,16] or have explicitly included a post-processing “pruning” mechanism [13,15] an alternative to reducing the effect of the medial axis instability is to utilize this instability to represent shape deformations: each deformation sequence can be broken into a series of contiguous deformation segments, where the medial axis topology does not change, connected by the transition point itself. This annotation of a shape deformation by the medial axis transitions, or discretization of a continuous path into a set of discrete descriptors, has been key in the development of object recognition and perceptual



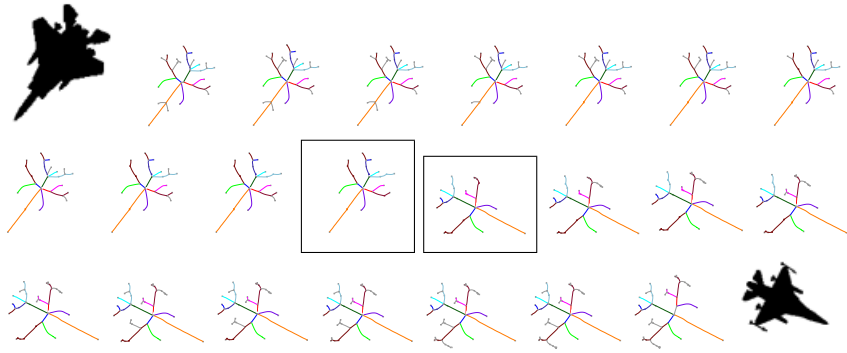
grouping approaches using the medial axis in 2D [10,14]. This paper investigates the transitions of the medial axis in 3D.

The transitions of the 2D Medial Axis under a one-parameter family of deformations were derived in [6] using results from transitions of the symmetry set [4]. These consist of two transitions,  $A_1A_3$  and  $A_1^4$ , as shown in Figure 1. The notation  $A_n^k$  implies  $k$ -fold tangency of order  $n$ , *i.e.*,  $A_1^2$  indicates the most generic situation of a circle tangent at two places each with regular tangency,  $A_1^3$  indicates a circle tangent at three places each with regular tangency, and  $A_3$  indicates a circle tangent at a curvature extremum. The three types  $A_1^2$ ,  $A_1^3$ , and  $A_3$  are the only generic forms of the medial axis in 2D [8], while generic transitions under a one-parameter family of deformations are the  $A_1A_3$  and  $A_1^4$  transitions. The  $A_1A_3$  transition occurs frequently due to boundary noise and is recognized as one of the classical instabilities of the medial axis. This transition is the result of formation of a bump on the boundary of the shape which initially bends the medial axis, but when it grows in size it will eventually “break” the axis leading to the growth of a new branch, Figure 1. The second medial axis transition, the  $A_1^4$  transition, occurs when a smooth  $A_1^2$  curve segment on the medial axis between two  $A_1^3$  points shrinks to a point so that the combination of two three-contact  $A_1^3$  points leads to a single four-contact  $A_1^4$  point (this is generic only in a family of curves), Figure 1. As the shape is compressed along the direction of the central  $A_1^2$  curve, this curve shrinks so that eventually its  $A_1^3$  end-points overlap, as in the fourth figure in the bottom row of Figure 1, the  $A_1^4$  transition. Additional deformations of the shape will form a new  $A_1^2$  axis by swapping the pairing of the four branches coming into the  $A_1^4$  points.

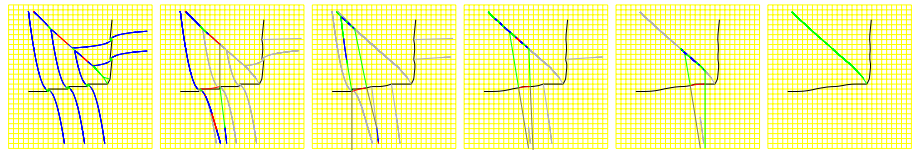


**Fig. 1.** (Top) The  $A_1A_3$  transition: the formation of a bump initially bends the related portion of the medial axis, however, as the bump grows, at some point the axis forms a discontinuity, the  $A_1A_3$  transition after which a new axis grows at this point. (Bottom) The  $A_1^4$  Transition: changes in the aspect ratio of a shape, *e.g.*, as caused by compressing it along a certain direction, can cause the central portion of the medial axis to shrink to a point ( $A_1^4$  contact). Further deformation causes the growth of another axis in the center. The transition itself can mediate the process of making the two sets of shapes on either side equivalent.

The derivation and a complete classification of these transitions, which are the *instabilities* of the medial axis, or its variant the shock graph, is significant in many ways. For example, successful recognition relies on an understanding of these points of instability so that the resulting medial axis graphs on either side of the transition, which have different structures, can be explicitly related. In



**Fig. 2.** The key role of characterizing the medial axis transitions in a 2D object recognition approach is highlighted. An arbitrary deformation sequence from one shape to another is described by the set of transitions the medial axis experiences. This “discretization” relies on the instabilities of the medial axis to reduce the search space for finding the optimal deformation path to a practical range. Results of the optimal sequence for a pair of planes are shown [14].



**Fig. 3.** The smoothing of the “L-shape” with noise and the square retain the coarse scale boundary singularity (corner of the L) while removing the noise [17].

the approach described in [14] an equivalence relation is defined for all shapes with the same shock graph topology, and another equivalence relation for all deformation paths with an identical sequence of transitions. Shape similarity is defined as finding the least action deformation path, where costs are associated with each deformation segment between transitions. Thus, the problematic medial axis instabilities are in fact used to advantage to effect stable recognition, Figure 2.

These transitions are also of critical value to perceptual grouping operations [10]. Specifically, note that the smoothing of a boundary can be achieved by deforming a shape to a neighboring transition. For example, since small perturbations cause growth after the  $A_1A_3$  transition, one can remove branches which likely arose in this manner by retaining an  $A_1A_3$  transition at that point [17], Figure 3. Other group operations are described in [10].

**The Local Form of 3D Medial Axis:** The goal of this paper is to study the transition of the medial axis of a 3D shape. The local form of the medial axis points was classified into five types [7] based on the order of contact of the corresponding sphere of tangency. The most typical example of a medial axis point is the centers of ordinary bitangent spheres,  $A_1^2$  points. These  $A_1^2$  points

organize as sheets with other neighboring  $A_1^2$  points. Less frequently, some medial axis points have spheres which are tangent at three points,  $A_1^3$  points. These organize into curves with neighboring  $A_1^3$  points, and often represent the central axis of the shape. Note that the  $A_1^3$  curves are at the intersection of three  $A_1^2$  sheets. Another possibility is for the two  $A_1^2$  points to coincide to form an  $A_3$  curve, the *rim* curve, which bounds a single  $A_1^2$  sheet and corresponds to a *ridge* on the surface. Much less frequently, some MA points may acquire four points of tangency,  $A_1^4$  points. These are generic, in contrast to five points of tangency or higher which disappear with small perturbations of the shape. The  $A_1^4$  points are isolated from other  $A_1^4$  points, and form from the intersection points of six  $A_1^2$  sheets and four  $A_1^3$  curves. Finally, some MA points are centers of spheres with regular  $A_1$  tangency at one point, and a higher-order  $A_3$  contact at another, as denoted by the  $A_1A_3$  notation. These points are isolated from other  $A_1A_3$  points, but both end at an  $A_1A_3$  point. These five types of MA points, namely  $A_1^2$  (sheets),  $A_1^3$  and  $A_3$  (curves), and,  $A_1^4$  and  $A_1A_3$  (points) are the only generic types of MA points in 3D.

**Overview :** We approach the problem of classifying the transitions of the medial axis in 3D in two distinct ways. First, in Section 2 we examine how a generic medial axis point can acquire additional non-generic contact by studying possible interactions between each pair of generic medial axis types. Some interactions can never occur, while others lead to transitions which are shown to be viable by creating an example. Second, in Section 3 we relate these transitions of the 3D medial axis to a classification of the singularity of viscosity solutions of Hamilton-Jacobi equations. Finally, in Section 4 we describe examples for each transition and illustrate its formation in a sequence of deformations.

## 2 Description of the Transitions of Medial Axes in 3D

In this section we shall describe how the possible transitions of generic 1-parameter families of medial axes in 3D can be enumerated. The list we produce is motivated by the work of Bogaevsky [2,1] and in Section 3 we give a brief indication of the connection between his work and our list<sup>1</sup>.

We can approach the listing from several perspectives; first let us consider the concept of **specialization**. If one singularity type, such as  $A_1^2$ , *specializes* to another, such as  $A_3$ , this means that in a general family we can have a sequence of  $A_1^2$  singularities which, in the limit, becomes  $A_3$ . Equivalently, we can perturb an  $A_3$  by an arbitrarily small amount so that it yields  $A_1^2$ . We write  $A_1^2 \rightarrow A_3$  or  $A_3 \leftarrow A_1^2$ . This specialization can indeed occur: as the two points of contact of a bitangent sphere tend to coincidence, so the limit of these points of contact is a ridge point of the surface (and in fact the line joining the points of contact has for its limit a principal direction at the ridge point [5]). On the other hand,  $A_1^3$

<sup>1</sup> It needs to be said here that technical difficulties, which are at present under active consideration, make it impossible for us to be completely precise about the meaning of ‘generic’ and about the equivalence relation which implicitly underlies our finite list of pictures. However, there is no doubt that all cases have been considered and there is no transition which ‘looks’ different from those we present here.

does not specialize to  $A_3$ , or, putting it the other way round, we cannot perturb an  $A_3$  by an arbitrarily small amount so as to achieve  $A_1^3$ .

Another principle which we need is that in a 1-parameter family of surfaces we can expect singularities to occur which are of one higher codimension (order of complexity) than those occurring for a single surface. Those for a single surface are  $A_1^2, A_1^3, A_1^4, A_3$  and  $A_1A_3$  and for a family of surfaces this list will be augmented by singularities only involving  $A_n^k$  with odd  $n$  and codimensions (sum of suffixes) adding to 5, (continuing to use the standard Arnold's notation for singularities, as described for example in [3]):

$$A_1^5, A_1^2A_3, A_5, D_5,$$

but  $D_5$ , whose "normal form" as a function of two variables is  $x^2y + y^4$ , is not a local minimum, so does not occur in the medial axis context. Note that we might also expect earlier singularities to produce new phenomena. The specializations relevant to the present situation are then:<sup>2</sup>

$$\begin{array}{ccccccc} A_1 & \rightarrow & A_1^2 & \rightarrow & A_1^3 & \rightarrow & A_1^4 & \rightarrow & A_1^5 \\ & & \downarrow & & \downarrow & & & & \\ & & A_3 & \rightarrow & A_1A_3 & \rightarrow & A_1^2A_3 & & \\ & & & & & & \downarrow & & \\ & & & & & & A_5 & & \end{array}$$

From this, other specializations can be deduced, for example when we perturb  $A_1^2A_3$  we can expect to find  $A_1A_3$  (since  $A_1 \rightarrow A_1^2$ ) and also (for a different perturbation)  $A_1^4$  (since  $A_1^2 \rightarrow A_3$ ).

The second approach is via the **geometry** of the situation. When one or more singularities coalesce to a more complex one (as with  $A_1^3 \rightarrow A_5$ ) we can ask how this happens geometrically. A useful result in this direction is the following.

**Proposition 1** Let  $\gamma$  be a point of the medial axis of the surface  $M$  at which the singularity is of the simplest kind,  $A_1^2$ , so that the medial axis is *smooth* at  $\gamma$ . We can therefore consider a sufficiently small neighborhood  $U$  of  $\gamma$  in 3-space, contained within the maximal ball  $D$  (solid 3-sphere) whose center is  $\gamma$ , and intersecting the medial axis in a smooth surface  $S$ . Then  $U$  does not contain any points of the medial axis of  $M$  besides those in  $S$ . This means that  $A_1^2$  singular points do not coalesce with others in making a more complex singularity: other points of the medial axis cannot approach a smooth  $A_1^2$  point in order to 'collide' with it.

**Proof.** Let  $\gamma^+$  be one of the points of contact of the bitangent sphere centered at  $\gamma$  and let  $p$  be a point on the radius of this sphere from  $\gamma$  to  $\gamma^+$  other than  $\gamma$  itself. Suppose  $p$  is on the medial axis; then it is the center of a maximal ball  $D'$ . This ball  $D'$  cannot have the point  $\gamma^+$  in its interior since  $D'$  would then

<sup>2</sup> A convenient way to visualize these specializations is to think of functions of one variable,  $y = f(x)$ , with an  $A_k$  singularity being a  $k+1$  fold root, and  $A_kA_l$  meaning both a  $k+1$  fold and an  $l+1$  fold root of the same function. Thus  $A_5$  represents a 6-fold root which can be perturbed for example into three 2-fold roots:  $A_5 \leftarrow A_1^3$ , but not into four 2-fold roots:  $A_1^4$  does not specialize to  $A_5$ .

contain points outside  $M$ . In that case  $D'$  must be small enough to be entirely inside  $D$ , and of radius strictly smaller than the radius  $r$  of  $D$ . It follows that the points where the boundary sphere of  $D'$  is tangent to  $M$  must also be strictly inside  $D$ , which is impossible since  $D$  is maximal. It now follows that all the radii outwards from *smooth* points of the medial axis near  $\gamma$  fail to contain points of the medial axis other than the points of the smooth  $A_1^2$  sheet near  $\gamma$ . But over a smooth piece of the medial axis the boundary retracts to the medial axis along the radii, so these radii fill out a neighborhood of  $\gamma$  in the surrounding 3-space, and this completes the proof. ■

We note that the same approach used to isolate an  $A_1^2$  point from other medial axis points can be applied in other situations: again, the radii out from medial axis points  $\gamma$  to points of contact  $\gamma^+$  will be free from other medial axis points. For example, two  $A_1^3$  curves cannot normally meet, but they can if they lie on the same sheet of the medial axis: they can then approach and become tangential. Similarly,  $A_3$ ,  $A_1A_3$  and  $A_1^4$  points can only be “approached” along sheets of the medial axis through them, not through the “empty space” between the sheets.

We can now make a list of the possible transitions on medial axes in 3D, with some commentary on each one. We envisage these as two singularities “coalescing” or “colliding” and use the specialization method to reduce the number of possibilities.

*Collisions involving  $A_3$ :* In this and the other tables, the third singularity is the simplest one which is a specialization of both colliding singularities  $\mathcal{A}$  and  $\mathcal{B}$ .

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C} \leftarrow \mathcal{A}$ and $\mathcal{C} \leftarrow \mathcal{B}$	See note
$A_3$	$A_3$ or $A_1A_3$	$A_5$	(1)
$A_3$	$A_1^3$	$A_1A_3$ [type II]	(2)
$A_3$	$A_1^4$	none—see above	

*Note (1).* Two  $A_3$  curves could in principle collide by moving in a common sheet of the medial axis, but since the contact points must remain on opposite sides of the collision this would imply  $A_3^2$  at the transition, and this is not one of our viable list of singularities.

*Note (2).* At the moment of collision the  $A_3$  and  $A_1^3$  curves must be tangential, something that only happens generically for a family of surfaces. The condition for this can be expressed in terms of the principal curvatures at the  $A_3$  point, the derivative of one of these and the position of the other contact point, using formulae of [5], but the result does not seem to be enlightening.

*Other collisions involving  $A_1^3$ :*

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C} \leftarrow \mathcal{A}$ and $\mathcal{C} \leftarrow \mathcal{B}$	See note
$A_1^3$	$A_1^3$	$A_1^4$	(3)
$A_1^3$	$A_1^4$	$A_1^5$	
$A_1^3$	$A_1A_3$	$A_1^2A_3$ [type II]	

*Note (3).* The two  $A_1^3$  curves must be tangent at the moment of collision. The tangent to an  $A_1^3$  axis passes through the circumcenter of the triangle through

the three contact points [5] so this implies that the four contact points for  $A_1^4$  will be *coplanar*.

*Other collisions involving  $A_1^4$ :*

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C} \leftarrow \mathcal{A}$ and $\mathcal{C} \leftarrow \mathcal{B}$	See note
$A_1^4$	$A_1^4$	$A_1^5$	
$A_1^4$	$A_1 A_3$	$A_1^2 A_3$ [type I]	

*Other collisions involving  $A_1 A_3$*  Here  $\mathcal{C} = A_5$  although as we shall see, the  $A_1 A_3$  transition itself can produce two  $A_1 A_3$  singularities.

**Table 1.** A summary of the transitions of the medial axis arising from the interaction of generic types presented in two distinct ways.

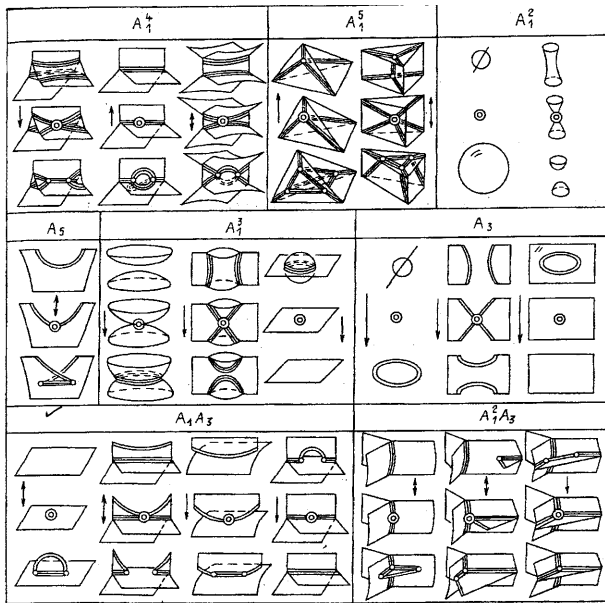
$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C} \leftarrow \mathcal{A}$ and $\mathcal{C} \leftarrow \mathcal{B}$	Transition	Collision of Types
$A_3$	$A_3$ or $A_1 A_3$	$A_5$	$A_1^4$	$A_1^3 - A_1^3$
$A_3$	$A_1^3$	$A_1 A_3$ -II	$A_1^5$	$A_1^4 - A_1^4, A_1^4 - A_1^3$
$A_1^3$	$A_1^3$	$A_1^4$	$A_5$	$A_1 A_3 - A_1 A_3, A_3 - A_3$
$A_1^3$	$A_1^4$	$A_1^5$	$A_1 A_3 - I$	$A_1 A_3 - A_1 A_3$
$A_1^3$	$A_1 A_3$	$A_1^2 A_3$ -II	$A_1 A_3 - II$	$A_1 A_3 - A_1 A_3, A_1^3 - A_3$
$A_1^4$	$A_1^4$	$A_1^5$	$A_1^2 A_3 - I$	$A_1^4 - A_1 A_3$
$A_1^4$	$A_1 A_3$	$A_1^2 A_3$ -I	$A_1^2 A_3 - II$	$A_1^3 - A_1 A_3$
$A_1 A_3$	$A_1 A_3$	$A_5, A_1 A_3$ -I, $A_1 A_3$ -II		

In summary, there are seven transitions which could possibly arise, Table 1. In Section 4 we show that the transitions do indeed arise in all cases by constructing an example for each. However, we first relate these to the singularity classification results of Bogaevsky in Section 3.

3 Relationship with the Work of Bogaevsky

In [2,1], I.A. Bogaevsky examines several related problems connected with the classification of transitions (perestroikas) of “minimum functions”. That is, we consider a local family of functions of the form  $F(t, \mathbf{x}) = \min f(t, \mathbf{x}, \mathbf{y})$  where  $t$  is a “time” parameter (so a single real parameter),  $\mathbf{x} \in \mathbf{R}^3$  and  $\mathbf{y}$ , over which the minimum is taken, is in a Euclidean space which for the medial axis application would be 2-dimensional, corresponding to the 2-dimensional surface whose medial axis is being considered. For each value of  $t$  close to some  $t_0$  (the moment of transition) we can consider the set  $X$  of points  $\mathbf{x}$  for which  $F$  is not differentiable. It is this set which is pictured as it evolves through the transition. Bogaevsky provides a complete list of these transitions, provided the function  $f$  is generic. The pictures which we reproduce are in Figure 4 from [1].

There are also two special cases considered. The first one, described as “shocks” (these are not the same as the shocks which we associate with a dynamical construction of the medial axis) is one to which an “arrow of time” can be associated, where in fact the local topology of the set  $X$  as above is “trivial”

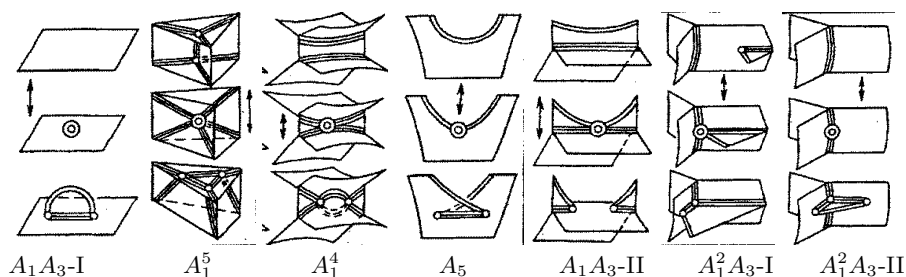


**Fig. 4.** From [1]. The 26 topologically different of perestroikas of the momentary shock waves are shown by arrows. Those relevant to the medial axis are those shown by double arrows. Single lines indicate the boundary of the depiction, double lines indicate rims,  $A^3$  curves on the medial axis which correspond to ridges on the surface, and triple lines indicate  $A_1^3$  curves. Concentric circles indicate an overlap of medial points.

(homotopic to a point) for all  $t \geq t_0$ : we might say loosely that moving through the transition from  $t < t_0$  to  $t > t_0$  “simplifies the topology.” The arrow of time is indicated on Bogaevsky’s diagrams and those without an arrow are transitions where trivial topology is guaranteed only at  $t_0$ .

The second special case is the one which concerns us most closely. Here, the topology of  $X$  is trivial for all  $t$  close to  $t_0$ , and this corresponds to the well-known fact that the medial axis of a smooth compact connected surface is contractible to a point. The arrow of time can then go either way. It is indicated on the diagrams by a double arrow. The diagrams with a double arrow therefore represent all the transitions on the medial axis, possibly including some extras which cannot occur for surfaces, provided the family of distance functions is generic<sup>3</sup>. Thus in this situation we have a family of surfaces parametrized say by  $\Gamma(t, \mathbf{y})$  where  $t$  is the 1-dimensional time parameter and  $\mathbf{y}$  is 2-dimensional. We then define  $f(t, \mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \Gamma(t, \mathbf{y})\|^2$ . The usual medial axis is simply the

<sup>3</sup> A technical difficulty that has so far not been resolved is whether this is equivalent to studying the transitions of the medial axis for a generic 1-parameter family of generic surfaces. Previous examples where this problem has arisen (*e.g.*, the study of caustics by reflection) have been answered positively, but each case needs a separate, and rather lengthy investigation.



**Fig. 5.** A summary figure of those singularities (perestroikas) of the viscosity solution of the Hamilton-Jacobi equation, represented in Figure 4, that relate to the transitions of the medial axis in 3D.

set of points for which the corresponding minimum function (minimum over  $\mathbf{y}$ ) is not differentiable. The singularities in Figure 4 involving double arrows are shown in summary form in Figure 5.

## 4 Examples and Illustrations

**1) The  $A_1A_3$ -I Transition :** Consider a medial axis sheet ( $A_1^2$  points) arising from two boundary surfaces. An analogous deformation to that in Figure 1 is where one of the boundary surfaces is protruded to form a bump. A good example of a generic bump is a two-dimensional Gaussian with asymmetric sigmas, *i.e.*,

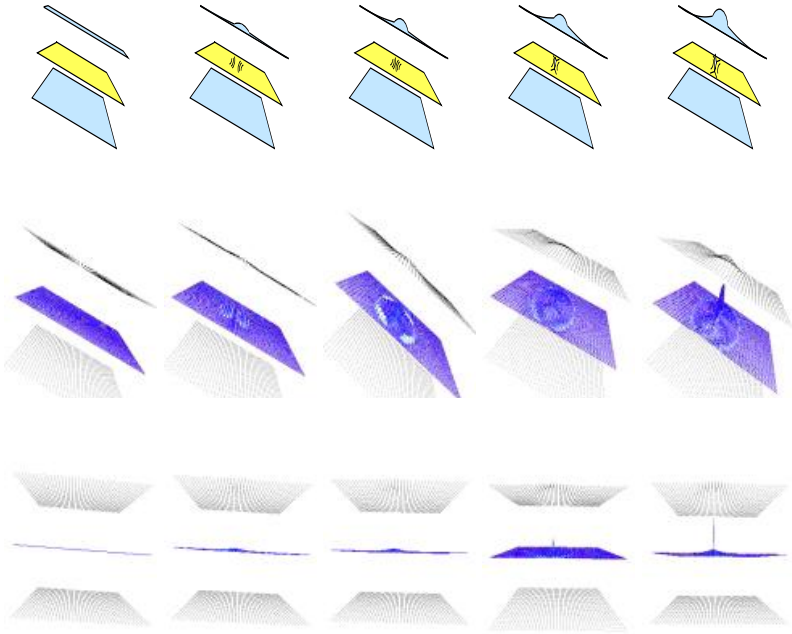
$$z = f(x, y) = Ae^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}} \text{ where } \sigma_x < \sigma_y. \quad (1)$$

Since the high or principal curvature is along the  $yz$  plane, the medial axis sheet arises from the centers of curvature of the curve  $z = f(x, 0)$ , whose curvature  $\kappa(x)$  given by

$$\kappa(x) = \frac{f''(x, 0)}{(1 + f'(x, 0)^2)^{\frac{3}{2}}} = \frac{A}{\sigma_x^4} e^{\left(\frac{-x^2}{2\sigma_x^2}\right)} (x^2 - \sigma_x^2).$$

In particular,  $\kappa(0) = \frac{-A}{\sigma_x^4}$ . This implies that in the absence of additional structure, the Gaussian bump will form an  $A_3$  point at  $\left(0, 0, f(0, 0) + \frac{1}{\kappa(0)}\right) = \left(0, 0, A - \frac{\sigma_x^2}{A}\right)$ . For a bump which is barely visible, the ratio of  $\sigma_x^2$  to  $A$  is very high, *i.e.*, the spread of the protrusion is much larger than its magnitude. Thus, the center of curvature forms very far from the peak and the likely presence of any other structure typically prevents the formation of the  $A_3$  point. As the salience of the bump increases in a one parameter family of deformations, the ratio  $\frac{\sigma_x^2}{A}$  decreases until at some point, the (thus far non-maximal) sphere of curvature of the bump also forms a contact elsewhere, forming an  $A_1A_3$  transition. This is parallel to the 2D case shown in Figure 1. From this point onward, the sphere is maximal and the corresponding medial point emerges, undisturbed by other structure. Figure 6





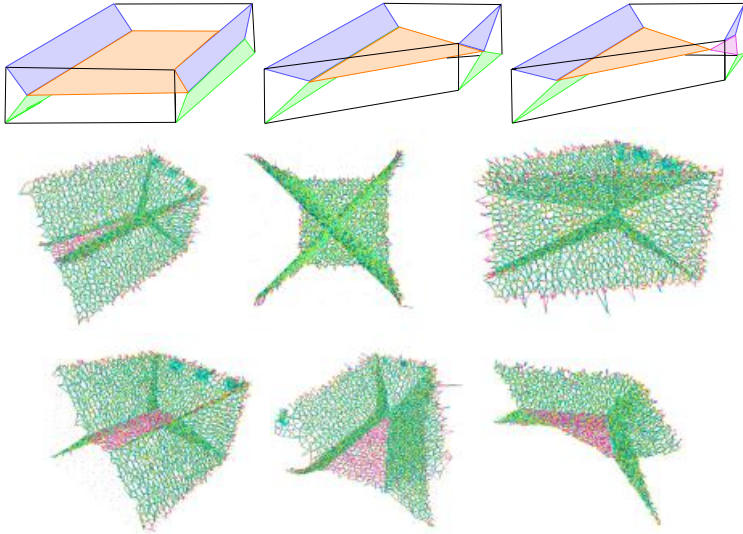
**Fig. 6.** The sketch (top row ) and two simulated views of the  $A_1A_3$ -I transition, (Middle and Bottom rows).(Simulations from [12]).

shows a sketch of this transition, and a simulation with  $\sigma_y = 2.5$ ,  $\sigma_x = 1$  and a sequence of increasing  $A$ , shown from two different views.

**2) The  $A_1^5$  Transition:** A second transition which also parallels the second 2D case shown in Figure 1 can be obtained by a shape deformation shrinking an  $A_1^3$  curve segment with two four-contact  $A_1^4$  end-points so as to obtain a five point contact  $A_1^5$  point. Since six sheets and four curves come together at an  $A_1^4$  point, the resulting  $A_1^5$  has six  $A_1^3$  curves and nine sheets: the common curve disappears from each leaving six in total; of the six sheets three are common to the two  $A_1^4$  points leaving nine in total.

An example can be easily created by simply considering the polyhedral shape formed from the tangent planes at each of five points on a sphere, representing the shphere of contact at the  $A_1^5$  transition shape. A perturbation of the five points so that they are no longer on a sphere, *e.g.*, by moving a point in or out of the sphere generates shapes on either side of the transition. Specifically, consider five points  $\{p_1, p_2, p_3, p_4, p_5\}$  on a (unit) sphere and the planes  $T_i$  tangent to the sphere at  $p_i = (x_i, y_i, z_i)$  represented by the normal vector  $n_i = (\cos \theta_i \cos \phi_i, \sin \theta_i \cos \phi_i, \sin \phi_i)$ ,  $i = 1, \dots, 5$ . Consider also  $(\theta_i, \phi_i) \in \{(0, 0), (\frac{\pi}{2}, 0), (-\frac{\pi}{2}, 0), (0, \frac{\pi}{2}), (0, -\frac{\pi}{2})\}$ . In this non-generic situation, the

skeleton has an  $A_1^5$  point, but any small variation in these angles, leads to the generic situation with two  $A_1^4$  points, Figure 7.



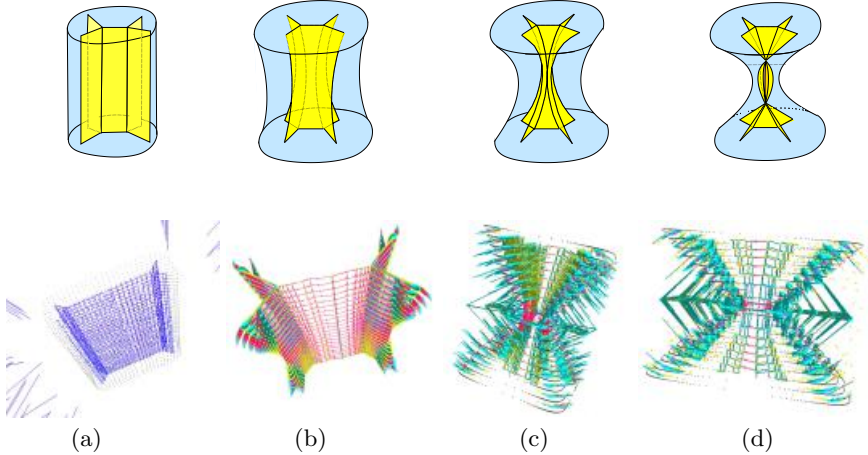
**Fig. 7.** A sketch and two views of the simulation of the  $A_1^5$  transition.(Simulations from [12]).

Note that these simulations are hard to see for two reasons : (i) From each view, the 3D structure of interest is occluded by intervening structure, so that only an interactive 3D visualization is fully effective in perceiving this; (ii) The medial axis, which is computed from point cloud data, is noisy and needs to be regularized. Successful regularization of the medial axis requires an understanding of how it is affected by small perturbances, precisely the topic of the paper! We expect to apply the transitions derived here for regularization of the medial axis in the near future.

**3)The  $A_1^4$  Transition :** In this transition two  $A_1^3$  curves approach each other with tangent planes to their respective sheets aligned at the moment of collision. Each of the three contact  $A_1^3$  points shares two points with the other so that at the transition we have a four point contact  $A_1^4$  medial axis point. Beyond this point there is a topological split in the tangential plane, due to collision of the two  $A_1^3$  curves, creating a gap in the middle of the two  $A_1^3$  curves connected via a newly generated transversal sheet bounded by two new  $A_1^3$  curves, ending at the two newly generating  $A_1^4$  points on either end. An example can be created by deforming a cylinder with a roughly rectangular base, Figure 8. This could be generated by moving a rectangular shape, *e.g.*, as generated by say a superquadric, in the  $z$  direction, leading to two  $A_1^3$  curves parallel to  $z$ -direction-axis. Now, to move the two  $A_1^3$  curves towards each other, we can pinch a “waist” in the cylinder and decrease the size of the waist until transition point where the two  $A_1^3$  curves become tangent at an  $A_1^4$  point. With additional pinching two  $A_1^4$  points move in opposite directions. Specifically, consider the 3D superquadric

$$\left(\frac{x}{a_1 f(z)}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{a_2 f(z)}\right)^{\frac{2}{\epsilon}} = 1, \text{ where } 0 < \epsilon < 2, 0 < a_1 < a_2, f(z) = \frac{z^2 - 1 + a_3}{a_3}, \quad (2)$$

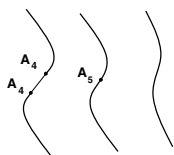
so that at  $z = \pm 1$  we have  $f(z) = 1$ , *i.e.*, the cross section is independent of the parameter  $a_3$ . We then vary  $a_3$  to change the extent by which the tube is “squeezed”, Figure 8.



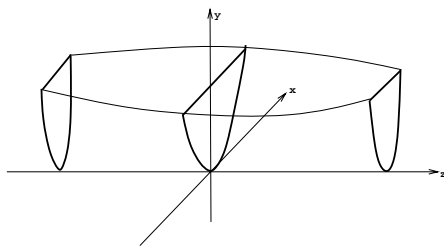
**Fig. 8.** Sketch and Example of Perestroika with an  $A_1^4$  point. (Simulations from [12]).

**4) The  $A_5$  Transition :** In this transition, in the course of a deformation sequence, two surface ridges approach each other and combine into one. Each ridge is manifested as an  $A_3$  curve on the medial axis, terminating at an  $A_1A_3$ . A common  $A_1^3$  axis connects the two ridges close to the  $A_5$  transition. To create an example, one can initially use the  $A_1A_3$  transition to generate a ridge and a corresponding  $A_3$  medial curve hanging onto a newly generated  $A_1^3$  curve. This can then be followed by a second  $A_1A_3$ -I transition to generate another ridge, and move the newly generated  $A_3$  curve to split the existing  $A_1^3$  curve via an  $A_1A_3$ -II transition, to arrive at the picture at the bottom of the  $A_5$  transition, Figure 5.

It is interesting to approach this transition on the medial axis from the perspective of transition on ridges. These transitions have been studied extensively in [9]. The only transition of ridges which is manifested in the medial axis is the  $A_5$  transition where two  $A_4$  turning points approach to form the  $A_5$ , Figure 9. The segment of the ridge between the two  $A_4$  turning points is not visible on the medial axis, in the sense that it does not give rise to maximal spheres. The portions of the ridge, which can be visible lead to two  $A_3$  curves, which *must* each end in an  $A_1A_3$  before the  $A_4$  turning points are reached.



**Fig. 9.** The only transition of ridges which is manifested in medial axis: two  $A_4$  turning points, bounding an “invisible” portion of the ridge approach each other and form an  $A_5$  point at the transition. The entire ridge is visible after this point.



**Fig. 10.** Generating an  $A_5$  transition shape which can then be perturbed to give shapes on either sides of the transition.

We can generate a shape with  $A_5$  contact and perturb it to see the effect of changes at the  $A_5$  transition. Consider the following graph of  $f(x)$  and its curvature:

$$f(x) = x^2 + bx^3 + cx^4 + \cdots, \quad \kappa(x) = \frac{2 + 6bx + 12cx^2}{[1 + (2x + 3bx^2 + 4cx^3 + \cdots)]^{\frac{3}{2}}}.$$

Thus, the distance of  $(x, f(x))$  from the center of curvature  $(0, \frac{1}{2})$  is

$$d^2(x) = x^2 + (x^2 + bx^3 + cx^4 + \cdots - \frac{1}{2})^2 = -bx^3 + (1-c)x^4 + 2bx^5 + (b^2 + 2c)x^6 + \cdots$$

To get an  $A_5$  contact we need to have  $b = 0$ ,  $c = 1$ , but a non-zero coefficient of  $x^6$ , which can be ensured by simply ignoring all higher order terms in  $f(x)$  after  $x^4$ , since  $b^2 + 2c = 2 \neq 0$ .

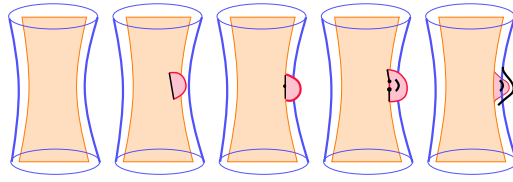
We can now generate a 3D shape by translating this graph as a profile and slightly perturbing it to general shapes on either sides of the transition, Figure 10,

$$y^2 = x^2 + \left( \frac{z + a_1}{a_2} \right) x^3 + x^4.$$

The parameter  $a_2$  when high enough ensures that near  $z = 0$  the function behaves as  $x^2 + x^4$ . The parameter  $a_1 = 0$  gives the transition shape, but perturbing  $a_1$  to non-zero values should interfere with the formation of  $A_5$  contact at  $z = 0$ .

We now construct an example where a ridge is added to an object which already has a “prominent” ridge using the transition shown in Figure 6. As the newly formed ridge approaches the existing ridge it “dissolves” into it, taking a new identity, as the left to right sequence in the  $A_5$  transition sketched in Figure 11 illustrates. Specifically, the example is generated from a cylinder with an elliptical base which is “pinched” in the middle  $\frac{x^2}{a_1^2} + \frac{y^2}{a_2^2} = \frac{z^2 - 1 + a_3}{a_3}$ , where  $a_1 > a_2$  controls the aspect ratio of the ellipse (eccentricity) and  $a_3 \geq 1$  controls the size of the “waist”, Figure 11(a). This formulation ensures that the  $z^2 = 1$  level sets are the same as  $a_3$  is altered.

Now, consider placing a “bump” near the ridge in the central portion, through an  $A_1A_3$ -I transition, Figure 11(b), as explained earlier. As the “bump” travels

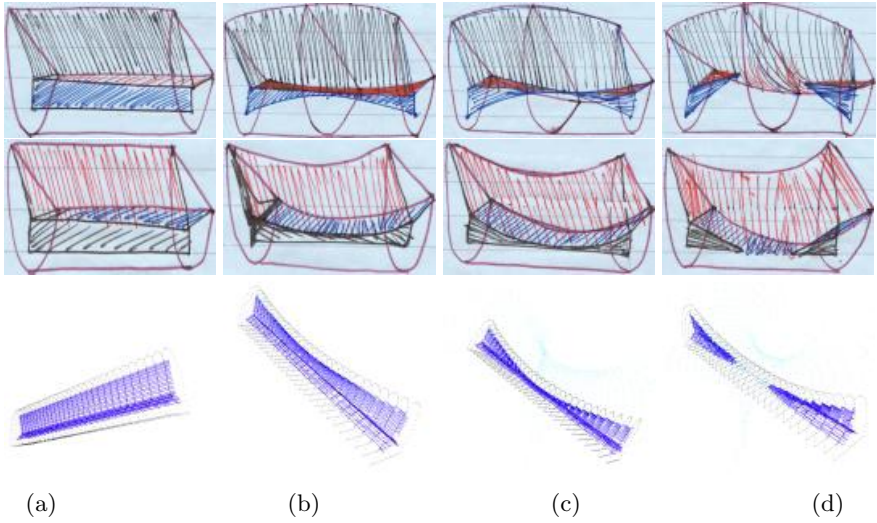


**Fig. 11.** The  $A_5$  transition is a merging of two ridges into one. In this example we begin with an existing ridge (ridge of elliptical cylinder), create a new one (red tab), move it towards the original ridge until they merge.

towards the ridge, the rib corresponding to the ridge and the  $A_1^3$  axis corresponding to the bump experience an  $A_1A_3$ -II transition, Figure 11(c). After this point, we have two situations, the top main ridge connected with the new ridge and similarly for the bottom main ridge, which are candidates for the  $A^5$  transition. Note that the original ridge has somewhat flattened due to the nearby placement of the new bump. Thus, its effect is manifested in the  $A_3$  axis on each side. Note that a requirement of the shrinkage of the  $A^3$  axis in this transition is that the  $A_3$  curve becomes tangent to it.

**5) The  $A_1A_3$ -II Transition :** In this transition two curves, one  $A_3$  and one  $A_1^3$ , approach each other, become tangent, and then split into two portions each, grouped at the new  $A_1A_3$  points. It is rather straightforward to generate examples for this transition: by “flattening ” a ridge, its  $A_3$  (rim) curve moves into the  $A_1^3$  sheet it is on. If this sheet is bounded by an  $A_1^3$  curve elsewhere, the  $A_3$  and  $A_1^3$  curves can eventually collide, Figure 12, top row. Alternatively, the ridge can be fixed and the “width” of the shape can be reduced, thus moving the  $A_1^3$  curve closer to the  $A_3$  curve, with the same effect. In the first case, we sweep a parabola along the  $z$  axis, widening it at  $z = 0$ , and closing it up with a “top” at  $y = 1$ . Note that the parabola  $y = ax^2$ , has curvature  $2a$  at its vertex, so that when swept it generates a ridge and a corresponding  $A_3$  curve (rim), Figure 12. By a modification, we can make this parabola bulge in the middle while keeping  $z = \pm 1$  sections constant.  $y = [a_1(1 - z^2) + a_2z^2]x^2$ , so that at  $z = 0$ , the curvature of the cross section is  $2a_1$ , while at  $z = \pm 1$  the curvature of the section is  $2a_2$ . Alternatively, one can close up the graph  $y = a_1x^2$  with a “top”  $y = a_2z^2$ , as illustrated in Figure 12, middle row.

**6) The  $A_1^2A_3$ -II Transition:** This transition describes the interaction of a ridge curve endpoint  $A_1A_3$  with a distinct central axis curve  $A_1^3$ . The ridge initially “hangs off” a “central” portion of some medial axis sheet, as generated by the  $A_1A_3$ -I example (the only way to generate a new ridge!). The latter sheet typically intersects two other sheets at an  $A_1^3$  curve. Now, as the ridge is moved so that its corresponding endpoint  $A_1A_3$  on the medial axis moves towards the latter  $A_1^3$  axis, it is possible for the two to interact at some point. Since the end points of a rim are  $A_1A_3$  points, we expect these to approach the  $A_1^3$  curves transversally. At the point of collision the  $A_1A_3$  contact sphere of the rim-end encounters the third boundary surface (*i.e.*, the third contact in the  $A_1^3$  curve not represented by the  $A_1^2$  sheet). Thus, it becomes an  $A_1^2A_3$  contact point joining



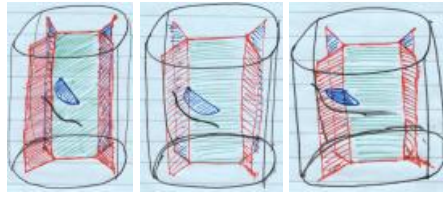
**Fig. 12.** (Top row) : The flattening of the central portion of this parabolic shape moves the ridge-related  $A_3$  curve (rim) up towards the central axis  $A_1^3$  curve until they collide at an  $A_1A_3$ -II transition. After this point the  $A_1^3$  and  $A_3$  curves both split into two pieces joined at two newly formed  $A_1A_3$  points. Note that the medial axis between these points is a smooth  $A_1^2$  sheet. (Middle Row): An alternate form. (Bottom Row) : The simulation of the medial axis evolution showing the Perestroika at an  $A_1A_3$ -II point. (Simulations from [12]).

the  $A_3$  and  $A_1^3$  curves corresponding to the ridge and its base, respectively, with the two  $A_1^3$  half portion of the original  $A_1^3$  curves. After this point, as the ridge is pushed further along, the  $A_3$  contact of the ridge pairs off with the newly acquired  $A_1$  contact and form an  $A_1A_3$ . The  $A_1^2A_3$  contact becomes tangent at the original three points and the sides of the ridge for an  $A_1^4$  contact. Four  $A_1^3$  curves come together at this point, Figure 13.

A shape which gives rise to this transition can be constructed as follows: consider the shape used in Figure 8, generated by Equation 2, or the parabolic “gutter” in Figure 12. Create a bump using the  $A_1A_3$ -I transition by adding a Gaussian, Equation 1, in the central portion of the shape. We give this bump a random orientation  $\theta$ ,

$$y(x, z) = Ae^{\frac{-(x \cos(\theta) + z \sin(\theta))^2}{2\sigma_1^2}} e^{\frac{-(-x \sin(\theta) + z \cos(\theta))^2}{2\sigma_2^2}}. \quad (3)$$

The bump can then be placed on the surface and positioned roughly on the central portion, Figure 13, and then moved until its base on the medial axis (the  $A_1^3$  curve generated by the bump) collides with the  $A_1^3$  curve of the main shape, Figure 13(b). Thereafter, the base is split into two segments one on each of the corresponding sheets of the  $A_1^3$  curve. The simulation using a parabolic gutter is shown in Figure 14.



**Fig. 13.** The  $A_1^2 A_3$ -I transition. A “bump” (blue) is placed so that it modifies the central sheet of the medial axis (green). As the bump is moved left towards the  $A_1^3$  curve on the left side of the object, the  $A_1 A_3$  point of the bump eventually hits the  $A_1^3$  axis creating an  $A_1^2 A_3$  transition. Beyond this point the “base” of the bump is on two distinct sheets.



**Fig. 14.** Example of Perestroika with an  $A_1^2 A_3$ -I point. (Simulations from [12]).

**7) The  $A_1^2 A_3$ -II Transition :** This transition describes the process of creating a ridge whose base is on an  $A_1^3$  medial axis curve itself, as opposed to the central portion of an  $A_1^2$  sheet, *i.e.*, in creating the ridge in the  $A_1 A_3$  transition, the  $A_1 A_3$  medial point arises not on an  $A_1^2$  sheet but on the  $A_1^3$  curve.

This transition is related to the previous  $A_1^2 A_3$ -I transition, but instead of using an  $A_1 A_3$  transition to generate a ridge positioned on a medial sheet and then moved to collide with the  $A_1^3$  axis, it is generated and then moved right on the  $A_1^3$  axis. A shape deformation corresponding to this transition then applies an  $A_1 A_3$  transition of the form of Equation 3 but positioned on one of the true constant curves of the  $A_1^3$  medial curve on the surface, with the amplitude increasing. Initially when  $A$  is small, the effect is to perturb the  $A_1^3$  axis slightly. But with increasing  $A$  the transition is reached, and finally the ridge becomes apparent as the rightmost figures in Figures 13 and 14.

**Acknowledgments.** We gratefully acknowledge the support of NSF grants BCS-9980091 and ECS-0070887. We also appreciate the assistance of Mireille Boutin, Frederic Leymarie and Raghavan Dhandapani in creating some example cases and in the preparation of this document.

## References

1. I. A. Bogaevski. Singularities of viscosity solutions of Hamilton-Jacobi equations. 2001.

2. I. A. Bogavski. Metamorphoses of singularities of minimum functions and bifurcations of shock waves of the Burgers Eq. with vanishing viscosity. *Math. J.*, 1(4):807–823, 1990.
3. J. Bruce and P. Giblin. *Curves and Singularities*. Cambridge University Press, 1984.
4. J. Bruce, P. Giblin, and C. Gibson. Symmetry sets. *Proceedings of the Royal Society of Edinburgh*, 101A:163–186, 1985.
5. P. J. Giblin and B. B. Kimia. On the local form of symmetry sets, and medial axes, and shocks in 3D. Technical Report LEMS-171, LEMS, Brown University, May 1998.
6. P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, and medial axes, and shocks in 2D. In *ICCV*, pages 385–391, KerKyra, Greece, Sept. 1999.
7. P. J. Giblin and B. B. Kimia. On the local form of symmetry sets, and medial axes, and shocks in 3D. In *Proceedings of CVPR*, pages 566–573, Hilton Head Island, South Carolina, USA, June 13-15 2000. IEEE Computer Society Press.
8. P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *IJCV*, Submitted, March, 2001.
9. P. L. Halliman, G. G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Three-Dimensional Patterns of the Face*. A. K. Peters, 1999.
10. M. S. Johannes, T. B. Sebastian, H. Tek, and B. B. Kimia. Perceptual organization as object recognition divided by two. In *Workshop on POCV*, pages 41–46, 2001.
11. L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Trans. on PAMI*, 14(9):869–885, September 1992.
12. F. F. Leymarie. *3D Shape Representation via Shock Flows*. PhD thesis, Brown University, 2002.
13. R. L. Ogniewicz and O. Kubler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.
14. T. B. Sebastian, P. N. Klein, and 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.
15. D. Shaked and A. M. Bruckstein. Pruning medial axes. *Computer Vision and Image Understanding*, 69:156–169, 1998. 2.
16. S. Tari and J. Shah. Extraction of shape skeletons from grayscale images. *Computer Vision Image Understanding*, 66(2):133–146, 1997.
17. H. Tek and B. B. Kimia. Boundary smoothing via symmetry transforms. *Journal of Mathematical Imaging and Vision*, 14(3):211–223, May 2001.



# Learning Shape from Defocus<sup>\*</sup>

Paolo Favaro<sup>1</sup> and Stefano Soatto<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering  
Washington University, St. Louis - MO 63130, USA  
fava@ee.wustl.edu

<sup>2</sup> Department of Computer Science  
University of California, Los Angeles - CA 90095, USA  
soatto@ucla.edu

**Abstract.** We present a novel method for inferring three-dimensional shape from a collection of defocused images. It is based on the observation that defocused images are the null-space of certain linear operators that depend on the three-dimensional shape of the scene as well as on the optics of the camera. Unlike most current work based on inverting the imaging model to recover the “deblurred” image and the shape of the scene, we approach the problem from a new angle by collecting a number of deblurred images, and estimating the operator that spans their left null space directly. This is done using a singular value decomposition. Since the operator depends on the depth of the scene, we repeat the procedure for a number of different depths. Once this is done, depth can be recovered in real time: the new image is projected onto each null-space, and the depth that results in the smallest residual is chosen. The most salient feature of this algorithm is its robustness: not only can one learn the operators with one camera and then use them to successfully retrieve depth from images taken with another camera, but one can even learn the operators from simulated images, and use them to retrieve depth from real images. Thus we train the system with synthetic patterns, and then use it on real data without knowledge of the optics of the camera. Another attractive feature is that the algorithm does not rely on a discretization or an approximation of the radiance of the scene (the “deblurred” image). In fact, the operator we recover is finite-dimensional, but it arises as the orthogonal projector of a semi-infinite operator that maps square-integrable radiance distributions onto images. Thus, the radiance is never approximated or represented via a finite set of filters. Instead, the rank of the operator learned from real data provides an estimate of the intrinsic dimensionality of the radiance distribution of real images. The algorithm is optimal in the sense of  $\mathcal{L}^2$  and can be implemented in real time.

## 1 Introduction

A real imaging system involves a map from a 3-D environment to a 2-D surface which is function of the optical settings (e.g. focal length, lens diameter etc.) and the shape of the scene. The intensity at each pixel in an image also depends upon the radiance distribution<sup>1</sup> “glued” to the observed surface.

<sup>\*</sup> This research is sponsored by ARO grant DAAD19-99-1-0139 and Intel grant 8029.

<sup>1</sup> We do not distinguish between the radiance and the reflectance of a scene since both camera and lights are not moving.

Estimating shape from defocus consists in retrieving the depth information of a scene exploiting the blurring variation of a number of images captured at different focus settings. Together with depth, one can also recover the radiance distribution. Thus, the whole system is not only useful as a modality to retrieve depth with a single camera, but can also be used, for example, to enhance pictorial recognition systems by providing shape cues.

Defocused images are typically modeled as the result of a linear operator with a kernel that depends upon the optics of the camera as well as the three-dimensional shape of the scene, acting on an “ideal” deblurred image. The process can be expressed as:

$$I_p(x, y) = \int h_p^s(x, y) dR \quad (1)$$

where  $I_p$  denotes the image generated with optics parameters  $p$ ,  $(x, y)$  are coordinates lying on a compact discrete lattice (defined on the image plane),  $h_p^s$  is a family of kernels that depends on the scene shape  $s$  and the parameters  $p$ , and  $R$  is the radiance distribution defined on the continuous 3-D space.

If we collect a number  $K$  of images with different parameters  $p$ , and organize them as  $I = [I_{p_1} \ I_{p_2} \ \dots \ I_{p_K}]$  and we do the same for the corresponding kernels  $h^s = [h_{p_1}^s \ h_{p_2}^s \ \dots \ h_{p_K}^s]$ , then the equations can be rewritten more compactly as:

$$I(x, y) = \int h^s(x, y) dR \quad (2)$$

and we can pose the estimation of shape and radiance from blurred images as the following optimization problem:

$$\hat{s}, \hat{R} \doteq \arg \min_{s, R} \left\| I - \int h^s dR \right\|^2 \quad (3)$$

where  $\| \cdot \|$  is the Euclidean norm. Typically, the problem is addressed by inverting the imaging model to retrieve both the radiance distribution (the “deblurred” image) and the shape of the scene (the kernel parameters). This is a severely ill-posed inverse problem.

## 1.1 Relation to Previous Work

This work falls within the general category of passive methods for depth from defocus. That is, no active illumination is used, and no active control on the focus setting is performed (shape from focus [7,9]). A common assumption in passive shape from defocus is that shape can be approximated locally with planes parallel to the image plane. This implies that the family of kernels describing the imaging process is shift-invariant and common techniques can be applied such as Fourier transforms, moment filters, estimating relative blurring between images, or approximating images in the spatial domain through polynomials or simple discretizations [4,5,10,11,12,13,14]. Some work has been done also when kernels are shift-varying [8] and Markov random fields have been proposed as models for both depth and focused image [1,3]. Our approach differs from most current approaches in that we bypass both the modeling of the optics and the choice of a finite-dimensional representations for the radiance. Instead, working in a learning framework, we automatically learn the defocusing process through the

properties of linear operators. Our approach could potentially be applied to solve blind deconvolution problems in different fields.

When modeling the optics, one is faced with the problem of choosing the kernel family that best approximates the real imaging process. Kernel families (or *point spread functions*, PSF) adopted by most of the algorithms in the literature include *pillbox* functions and Gaussians. Also, during the shape reconstruction process, many algorithms require modeling the radiance using an ad-hoc choice of a basis of filters, or a discretization. Our approach does not entail any modeling of the PSF, nor a choice of a finite representation for the radiance distribution. We construct a set of linear operators (matrices) which are learned from blurred images. In order to do this, we generate a *training set* of images of a certain shape<sup>2</sup>, by changing the radiance distribution defined on its surface. Then, we capture the surface parameters common to the training set using the singular value decomposition (SVD), and build an orthogonal projector operator such that novel images generated by the same shape, but with any radiance, belong to its null space. We repeat this procedure for a number of shapes and store the resulting set of operators. Then, we compute the residual for each stored operator. Depth is estimated at each pixel by searching for the operator leading to the minimum residual. These operations involve only a finite set of matrix-vector computations (of the size of the set of the pre-computed orthogonal operators) which can be performed independently at each pixel on the images, allowing for high parallelism. Hence, the whole system can be implemented in real-time. Notice that, since the same set of operators is used for any novel set of images, shape reconstruction is independent of the radiance distribution, and no approximations or finite representations is required.

## 2 Learning Defocus

In this section we will explain in detail how to *learn* shape from defocused images through orthogonal operators, and how to use them to retrieve shape from novel images.

### 2.1 Formalization of the Problem

Under mild assumptions on the reflectance of the scene, the integral in Eq. (2) can be interpreted in the Riemannian sense (see [2]). If we choose a suitable parameterization, we can rewrite Eq. (2) as  $\int_{\mathbb{R}^2} h^s(x, y, \hat{x}, \hat{y}) r(\hat{x}, \hat{y}) d\hat{x} d\hat{y}$ , where  $r$  is the radiant density.  $r$  belongs to the Hilbert space  $\mathcal{L}^2(\mathbb{R}^2)$  with the inner product  $\langle \cdot, \cdot \rangle : \mathcal{L}^2(\mathbb{R}^2) \times \mathcal{L}^2(\mathbb{R}^2) \rightarrow \mathbb{R}$  defined by:

$$(f, g) \mapsto \langle f, g \rangle \doteq \int f(\hat{x}, \hat{y}) g(\hat{x}, \hat{y}) d\hat{x} d\hat{y}. \quad (4)$$

Since images are defined on a compact lattice  $C \doteq \mathbb{R}^{M \times N}$  (the CCD array grid), where images have size  $M \times N$  pixels, it is also useful to recall the finite dimensional Hilbert space  $\mathbb{R}^L$  of vectors, where  $L = M \times N \times K$ , with inner product  $\ll \cdot, \cdot \gg : \mathbb{R}^L \times \mathbb{R}^L \rightarrow \mathbb{R}$  defined as:

$$(V, W) \mapsto \ll V, W \gg \doteq \sum_{i=1}^L V_i W_i. \quad (5)$$

<sup>2</sup> In our current implementation we choose planes parallel to the image plane, but the method can be extended to other shapes as well.

We define the linear operator  $H_s : \mathcal{L}^2(\mathbb{R}^2) \rightarrow \mathbb{R}^L$  such that  $r \mapsto h_s^s(x, y, r)$ . Using this notation, we can rewrite our imaging model as

$$I(x, y) = (H_s r)(x, y). \quad (6)$$

The problem can then be stated as

$$s, r = \arg \min_{s \in D, r \in \mathcal{L}^2(\mathbb{R}^2)} \|I - H_s r\|^2 \quad (7)$$

where  $D$  is a suitable compact set (the space of shapes), and the norm  $\|\cdot\|$  is naturally induced by the inner product relative to the same Hilbert space, i.e.  $\|V\|^2 = \langle\langle V, V \rangle\rangle$ .

## 2.2 Learning Null Spaces

In order to compute explicitly our interest operators, we need some definitions. Since  $H_s$  is a linear bounded operator, it admits a unique adjoint  $H_s^* : \mathbb{R}^L \rightarrow \mathcal{H}(\mathbb{R}^2)$ , mapping  $I \mapsto \hat{\mathcal{H}}^s(\hat{x}, \hat{y}, I) \gg$ , where  $\hat{h}^s(\hat{x}, \hat{y}, x, y) = h^s(x, y, \hat{x}, \hat{y})$ , and such that

$$\langle\langle H_s r, I \rangle\rangle = \langle r, H_s^* I \rangle \quad (8)$$

for any  $r \in \mathcal{L}^2(\mathbb{R}^2)$  and  $I \in \mathbb{R}^L$ . Now we are ready to define the orthogonal projector, which is a matrix  $H_s^\perp : \mathbb{R}^L \rightarrow \mathbb{R}^L$  with  $I \mapsto H_s^\perp I = I - (H_s H_s^\dagger) I$ , where  $H_s^\dagger$  is the (Moore-Penrose) pseudo-inverse. The pseudo-inverse, when it exists, is defined as the operator  $H_s^\dagger : \mathbb{R}^L \rightarrow \mathcal{H}(\mathbb{R}^2)$  such that  $r = H_s^\dagger I$  satisfies  $H_s^*(H_s r) = H_s^* I$ .

The orthogonal operator  $H_s^\perp$  is useful in that the original (infinite dimensional) minimization in Eq. (7) can be transformed into a finite-dimensional one. This is guaranteed by the fact that the functionals

$$\Phi(s, r) = \|I - H_s r\|^2 \quad (9)$$

$$\Psi(s) = \|H_s^\perp I\|^2 \quad \text{with } r = H_s^\dagger I \quad (10)$$

have the same set of local extrema, assuming that there exists a non-null operator  $H_s^\perp$  and an operator  $H_s^\dagger$ , as discussed in [6,11].

Rather than obtaining a closed form solution for  $H_s^\perp$ , we take a different point of view, and choose to compute it numerically exploiting its properties.  $H_s^\perp$  is a symmetric matrix (i.e. such that  $H_s^\perp = (H_s^\perp)^T$ ) which is also idempotent (i.e. such that  $H_s^\perp = (H_s^\perp)^2$ ). According to the first property, we can write  $H_s^\perp$  as the product of a matrix  $A$  of dimensions  $m \times n$ ,  $m \geq n$ , with its transpose; for the second property we have that the columns of  $A$  must be orthonormal, and thus  $H_s^\perp$  can uniquely be written as:

$$H_s^\perp = A A^T \quad (11)$$

where  $A \in V_{n,m}$  and  $V_{n,m}$  is a Stiefel manifold<sup>3</sup>. Furthermore, from Eq. (10), the null space of  $H_s^\perp$  is all of the range of  $H_s$ . That is, given any set of  $T$  images  $\{I_i\}_{i=1 \dots T}$  generated by the same shape  $s_1$ , for fixed focal settings, but changing radiance densities, we have  $\|H_{s_1}^\perp I_i\|^2 = 0 \quad \forall i = 1 \dots T$ , where  $H_{s_1}^\perp$  has been trained with shape  $s_1$ .

Now we are ready to list the 4 steps to *learn*  $H_s^\perp$  from data, once a specific class of shapes has been chosen:

<sup>3</sup> A Stiefel manifold  $V_{n,m}$  is a space where each point is a set of  $n$  orthonormal vectors  $v \in \mathbb{R}^m$  and  $n \leq m$ :  $V_{n,m} = \{X(m \times n) : X^T X = I_d(n)\}$ , where  $I_d(n)$  is the  $n \times n$  identity matrix.

1. Generate a number  $T$  of training images  $I_1 \dots I_T$  of the shape  $s$  changing the optical settings; for example, see Figure 3. On the left two images have been generated. One horizontal stripe from each of the images corresponds to the same equifocal plane<sup>4</sup> in the scene. Hence, we can split each of the two stripes in  $T$  square windows. Then, we couple corresponding windows from the two stripes and collect them into  $I_i$ . Every image  $I_i$  is generated by a different radiance density  $r_i$ . Then, rearrange  $I_i$  in a column vector of length  $L$  for each  $i = 1 \dots T$ ;
2. Collect all images into a matrix  $P = [I_1 \ I_2 \ \dots \ I_T]$  of dimensions  $L \times T$ . Apply the singular value decomposition (SVD) to  $P$  such that  $P = USV^T$ . Figure 1 shows the resulting  $P$  for a single depth level and two optical settings;
3. Determine the rank  $q$  of  $P$  (for example by imposing a threshold on the singular values);
4. Decompose  $U$  as  $U = [U_1 \ U_2]$ , where  $U_1$  contains the first  $q$  columns of  $U$  and  $U_2$  the remaining columns; then build  $H_s^\perp$  as:

$$H_s^\perp = U_2 U_2^T. \quad (12)$$

The number  $T$  of training images has to be at least bigger or equal than  $L$  (in our experiments we set  $T = 2L$ ). Otherwise, the rank of  $P$  will not be determined by the defocusing process, but by the lack of data.

*Remark 1.* It is interesting to note that the rank of  $H_s^\perp$  relates inversely with the intrinsic dimensionality of the observed radiance. For example, if in the training phase we employ “simple” radiance densities, for example piecewise constant intensities, the rank of  $P$  will be low, and, as a consequence, the rank of the resulting operator  $H_s^\perp$  will be high. Intuitively, as the rank of  $H_s^\perp$  approaches  $L$  (the maximum value allowed), the bigger the set of possible orthogonal operators will be, and the bigger the space of shapes that can be distinguished.

This gives insight into how to estimate shape from defocus using linear operators, which is the case of most of the algorithms in the literature. There is a fundamental tradeoff between the precision with which we can determine shape and the precision with which we can model the radiance dimensionality: *the finer the shape estimation, and the more we require “simple” radiance densities. The more we model high dimensionality radiance densities, and the rougher the shape estimation.*

### 2.3 Estimating Shape

Certainly, it would not be feasible to apply the previous procedure to “recognize” any kind of shape. The set of orthogonal operators we can build is always with finite dimension, and hence we are only able to reconstruct shapes (which are defined in the continuum) up to a class of equivalence. Even restricting the system to recognize a small set of shapes would not be the best answer to the problem, since that would reduce the range

<sup>4</sup> An equifocal plane is a plane parallel to both the image plane and the lens plane. By the thin lens law, such a plane contains points with the same focal.

of applications. Therefore, we choose to simplify the problem by assuming that locally, at each point on the image plane, the corresponding surface can be approximated with an equifocal plane. Thus, restricting our attention to a small window around each pixel, we estimate the corresponding surface through one single parameter, its depth in the camera frame. Notice that the image model (2) does not hold exactly when restricted to a small window of the input image. In fact, it does not take into account additional terms coming from outside the window. However, these terms are automatically modeled during the training phase (see Section (2.2)), since we obtain the operators  $H_s^\perp$  using images with such artifacts. The overall effect is a “weighting” of the operators, where components relative to the window boundary will have smaller values than the others (see Figure 2).

Also, to achieve real-time performance, we propose to pre-compute a finite set of operators for chosen depths. Then, the estimation process consists in computing the norm (10) for each operator and searching for the one leading to the smallest residual. The procedure can be summarized as follows:

1. Choose a finite set of depths  $\mathcal{S}$  (i.e. our depth resolution) and a window size  $W \times W$  such that the equifocal assumption holds (in our experiments we choose windows of  $7 \times 7$  pixels). Then pre-compute the corresponding set of orthogonal operators using the algorithm described in Section (2.2)
2. At each pixel  $(i, j)$  of the novel  $L$  input images, collect the surrounding window of size  $W \times W$  and rearrange the resulting set of windows into a column vector  $I$  (of length  $LW^2$ )
3. Compute the cost function  $\|H_d^\perp I\|^2$  for each depth value  $d \in \mathcal{S}$  and search for the minimum. The depth value  $\hat{d}$  leading to the minimum is set to be the reconstructed depth of the surface at  $(i, j)$ .

This algorithm enjoys a number of properties that make it suitable for real-time implementation. First, the only operations involved are matrix-vector multiplications, which can be easily implemented in hardware. Second, the process is carried out at each pixel independently, enabling for high parallelism computations. It would be possible, for example, to have CCD arrays where each pixel is a computing unit returning the depth relative to it.

Also, as to overcome the limitations of choosing a finite set of depth levels, one can further refine the estimates by interpolating the computed cost function. The search process can be accelerated, using well-known descent methods (i.e. gradient descent, Newton-Raphson, tangents etc.), or using a dicotomic search, or exploiting the previous estimates as a starting point (this relies on smoothness assumptions on the shape). All these improvements are possible since the computed cost functions are smooth, as it can be seen in Figure 5.

### 3 Experiments

We test our algorithm on both synthetic and real images. In generating a synthetic set of images, we use a *pillbox* kernel, for simplicity. We use such synthetic images to train the

system, and then test the reconstruction on real images captured using a camera kindly lent to us by S. K. Nayar. It has two CCDs mounted on carts with one degree of freedom along the optical axis. These carts can be shifted by means of micrometers, which allows for a fine tuning of the focal settings.

### 3.1 Experiments with Synthetic Data

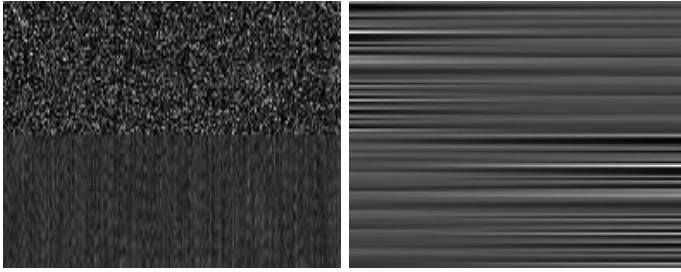
In Figure 1 we show part of the training set used to compute the orthogonal operators. We choose a set of 51 depth levels and optical settings such that the maximum blurring radius is of 1.7 pixels. The first and last depth levels correspond to the first and second focal planes. We generate random radiance densities and for each of them compute the images as if they were generated by equifocal planes placed at all the chosen depth levels. Based on these images we compute the 51 orthogonal operators  $H_s^\perp$ , 6 of which are shown in Figure 2. Notice that the operators are symmetric and that as we go from the smallest value of the parameter  $s$  (relative to the first depth level) to the biggest one (relative to the last depth level), the distribution of “weights” moves from one side of the matrix to the other, corresponding to the image more in focus.

To estimate the performance of the algorithm we generate a set of two novel images of  $51 \times 2601$  pixels that are segmented in horizontal stripes of  $51 \times 51$  pixels (see Figure 3). Every stripe has been generated by the the same random radiance but with equifocal planes at decreasing depths as we move from the top to the bottom of the image. We generate a plane for each of the 51 depths, starting from 0.52m and ending at 0.85m. In Figure 4 we evaluate numerically the shape estimation performance. Both mean and standard deviation (solid lines) of the estimated depth are plotted over the ideal characteristic curve (dotted). Notice that even when no post-filtering is applied, the mean error (RMS) is remarkably small (3.778mm). In the same figure we also show the characteristic curve when a  $3 \times 3$  median filter is applied to the resulting shape. While there is a visible improvement in the standard deviation of the shape estimation, the mean error (RMS) presents almost no changes (3.774mm).

In Figure 5 we show a typical minimization curve. As it can be seen, it is smooth and allows for using descent methods (to improve the speed of the minimum search) and interpolation (to improve precision in determining the minimum).

### 3.2 Experiments with Real Data

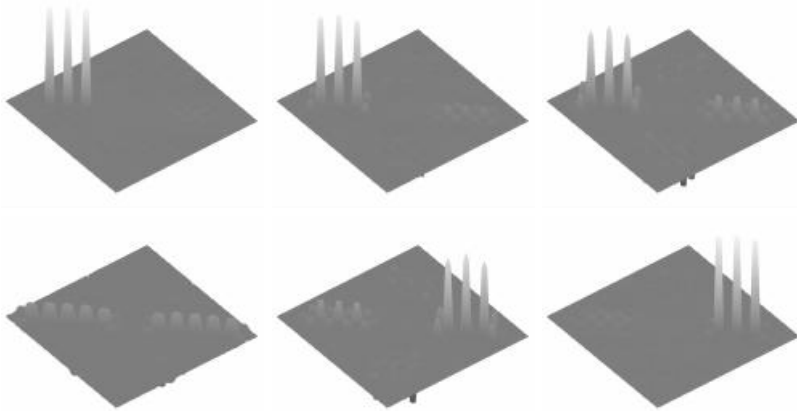
In order to test the robustness of the proposed algorithm, we train it on synthetic images as described above, and then use it on real images obtained with two different cameras. In the first data set we capture images using an 8 bits camera made of two independently moving CCDs, with focal length of 35mm and lens aperture of  $F/8$ . Figure 6 shows the two input images and the relative depth reconstruction. The far focused plane is at 0.77m while the near focused is at 0.70m. Windows of  $7 \times 7$  pixels have been chosen. Figure 7 shows an experiment with the second data set (provided to us by S. K. Nayar) of a scene captured with a different camera. The focal length is of 25mm and the lens aperture of  $F/8.3$ . Even though no ground truth is available to compare to the reconstructed surface, the shape has been qualitatively captured in both cases.



**Fig. 1.** *Left: the training set for the first depth level (corresponding to blurring radii 0 and 1.7 pixels for the first and second image respectively). The resulting matrix has size  $98 \times 196$  pixels. Right: the training set produced with a single radiance for all the 51 depth levels (the matrix has size  $98 \times 51$  pixels).*

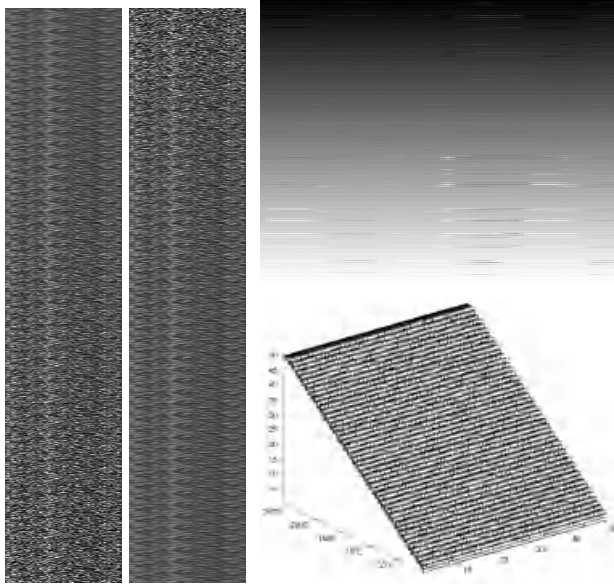
## 4 Conclusions

We have presented a novel technique to infer 3-D shape from defocus which is optimal in the sense of  $\mathcal{L}^2$ . We construct a set of linear operators  $H_s^\perp$  (matrices) *learning* the left null space of blurred images through singular value decomposition. This set of orthogonal operators is used to estimate shape in a second stage, computing the cost function  $\|H_s^\perp I\|$  for each  $H_s^\perp$  and novel input images  $I$ . Then, depth is determined performing a simple minimization of the residual value. The algorithm is robust, since

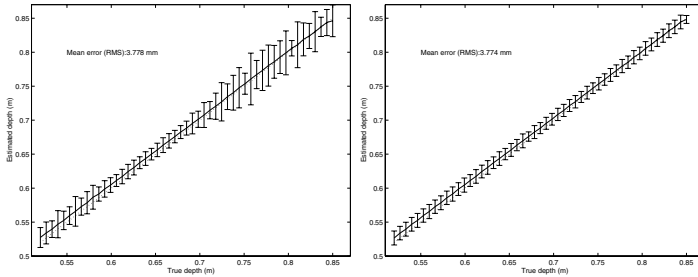


**Fig. 2.** *The operator  $H_s^\perp$  as it appears for few depth levels. Top: from left to right, depth levels are 1, 10 and 15, corresponding to blurring radii of (0, 1.7) pixels, (0.45, 1.25) pixels and (0.66, 1.04) pixels respectively, where with  $(b_1, b_2)$  we mean the blurring radius  $b_1$  is for the near focused image and the blurring radius  $b_2$  for the far focused one. Bottom: from left to right, depth levels are 20, 35 and 40, corresponding to blurring radii of (0.85, 0.84) pixels, (1.3, 0.38) pixels and (1.45, 0.25) pixels. Notice that as we go from left to right and from top to bottom, the weights of the operators move from one side to the other, corresponding to which of the images is more focused.*



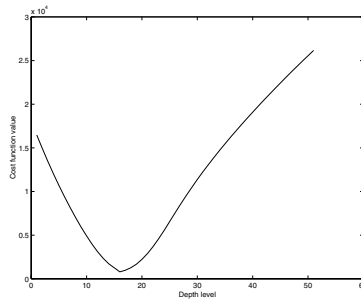


**Fig. 3.** Test with simulated data. Left and Middle: two artificially generated images are shown. The surface in the scene is a piecewise constant function (a stair) such that each horizontal stripe of  $51 \times 51$  pixels corresponds to an equifocal plane. Depth levels decrease moving from the top to the bottom of the images. Top-Right: gray-coded plot of the estimated depths. Darker intensities mean large depths values, while lighter intensities mean small depth values. Bottom-Right: mesh of the reconstructed depths after a  $3 \times 3$  pixels median filtering (re-scaling has been used for ease of visualization).



**Fig. 4.** Performance test. Left: evaluation of mean and standard deviation of the depth reconstruction. Right: evaluation of mean and standard deviation of the depth reconstruction after a  $3 \times 3$  pixels median filtering. Notice that the mean error (RMS) over all depths both with or without pre-filtering is substantially low (3.778mm and 3.774mm respectively).

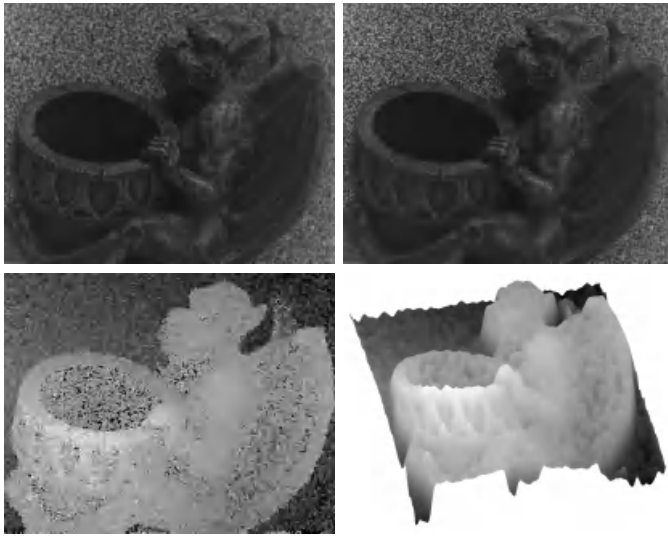
operators learned via simulated data can be used effectively on real data, and it does not entail any choice of basis or discretization of the radiance distribution. Furthermore, the rank of the orthogonal operators provides an estimate of the intrinsic dimensionality of the radiance distribution in the scene.



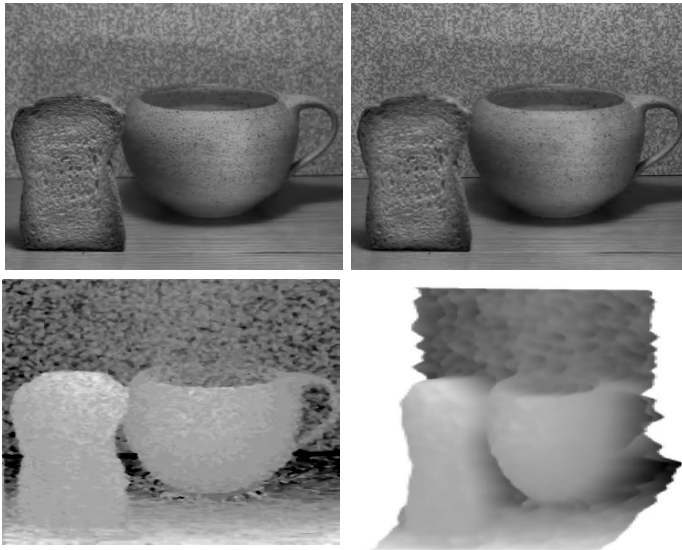
**Fig. 5.** One minimization curve. The curve is smooth and allows for fast searching methods. Also, interpolation can be employed to achieve higher precision in depth estimation.

## References

1. A. N. Rajagopalan and S. Chaudhuri. A variational approach to recovering depth from defocused images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, (no.10):1158–64, October 1997.
2. W. Boothby. *Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.
3. S. Chaudhuri and A. N. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.
4. J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:97–108, 1993.
5. P. Favaro and S. Soatto. Shape and reflectance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, pages 755–768, June 2000.
6. G. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least-squares problems whose variables separate. *SIAM J. Numer. Anal.*, 10(2):413–532, 1973.
7. R. A. Jarvis. A perspective on range finding techniques for computer vision. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 2, page 122:139, March 1983.
8. H. Jin and P. Favaro. A variational approach to shape from defocus. In *Proc. of the European Conference on Computer Vision*, volume in presss, May 2002.
9. S. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.
10. A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:523–531, 1987.
11. S. Soatto and P. Favaro. A geometric approach to blind deconvolution with application to shape from defocus. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 10–17, June 2000.
12. M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *Intl. J. of Computer Vision*, 13:271–294, 1994.
13. M. Watanabe and S. K. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.
14. Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, pages 68–73, 1993.



**Fig. 6.** First set of real images. Top: the two input images (of size  $640 \times 480$  pixels). The left image is near focused (at 0.70m), while the right one is far focused (at 0.77m). Bottom-Left: gray-coded plot of the estimated depths. Dark intensities map to large depths, while light intensities map to small depths. Notice that where the radiance is too poor (for example inside the vase), depth recovery is not reliable. Bottom-Right: smoothed 3-D mesh of the reconstructed surface.



**Fig. 7.** Second set of real images (provided to us by S. K. Nayar). Top: the two input images (of size  $320 \times 480$  pixels). The left image is near focused (at 0.53m), while the right one is far focused (at 0.87m). Bottom-Left: gray-coded plot of the estimated depths. Notice that where the radiance is too poor and the surface is far from being an equifocal plane (bottom of the table), depth recovery is not reliable. Bottom-Right: smoothed 3-D mesh of the reconstructed surface.

# A Rectilinearity Measurement for Polygons

Joviša Žunić\* and Paul L. Rosin

Computer Science, Cardiff University,  
Queen's Buildings, Newport Road, PO Box 916,  
Cardiff CF24 3XF, Wales, U.K.  
{J.Zunic, Paul.Rosin}@cs.cf.ac.uk

**Abstract.** In this paper we define a function  $\mathcal{R}(P)$  which is defined for any polygon  $P$  and which maps a given polygon  $P$  into a number from the interval  $(0, 1]$ . The number  $\mathcal{R}(P)$  can be used as an estimate of the *rectilinearity* of  $P$ . The mapping  $\mathcal{R}(P)$  has the following desirable properties:

- any polygon  $P$  has the estimated rectilinearity  $\mathcal{R}(P)$  which is a number from  $(0, 1]$ ;
- $\mathcal{R}(P)=1$  if and only if  $P$  is a rectilinear polygon, i.e., all interior angles of  $P$  belong to the set  $\{\pi/2, 3\pi/2\}$ ;
- $\inf_{P \in \Pi} \mathcal{R}(P) = 0$ , where  $\Pi$  denotes the set of all polygons;
- a polygon's rectilinearity measure is invariant under similarity transformations.

A simple procedure for computing  $\mathcal{R}(P)$  for a given polygon  $P$  is described as well.

**Keywords:** Shape, polygons, rectilinearity, measurement.

## 1 Introduction

Shape plays an important part in the processing of visual information, and is actively being investigated in a wide spectrum of areas, from art [13] through to science [3]. Within computer vision there have been many applications of shape to aid in the analysis of images, and standard shape descriptors include compactness, eccentricity [12], circularity [4], ellipticity [9], and rectangularity [11].

This paper describes a shape measure that has received little attention: *rectilinearity*. While there exists a variety of approaches to computing the related measure of *rectangularity* [11], rectilinearity covers a wider space of shapes since the number of sides of the model shape is variable. It is only required that the angles of a rectilinear polygon belong to the set  $\{\pi/2, 3\pi/2\}$ . This means that it is not convenient to fit the model<sup>1</sup> to the data and measure the discrepancies between the two, which is the approach that is often applied to compute compactness and rectangularity.

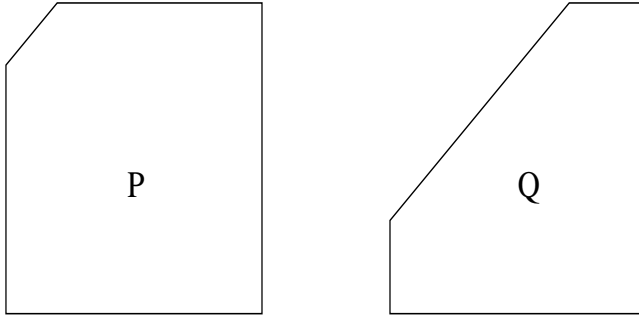
---

\* J. Žunić is also with the Mathematical Institute of Serbian Academy of Sciences, Belgrade.

<sup>1</sup> Fitting a rectilinear shape is possible, as demonstrated by Brunn *et al.* [1], but is complex, and potentially unreliable and inaccurate. Our proposed approach avoids fitting, and is therefore simpler and faster.

Our motivation in developing a rectilinearity shape measure is to provide a useful tool for the analysis of buildings in aerial photographs. Over the last 10-20 years there has been considerable research in this area with the goal of providing automatic photo interpretation which would be particularly useful for cartographers [2,5,6,8]. Such systems are typically based on perceptual grouping of edges and incorporate additional information such as shadows, knowledge-based rules, as well as shape (typically rectangular). Since many buildings appear rectilinear from an overhead view then such a shape measure could be used in a hypothesis and test paradigm to filter out unlikely candidates which have inappropriate shapes.

Let us conclude this introduction by noticing that a variety of rectilinearity measures for polygons which are based only on a measure of the angles of the considered polygon can be derived very easily. But such a defined rectilinearity measure would imply that the polygons with the same angles have the same estimated rectilinearity which is not always acceptable (see Fig. 1).



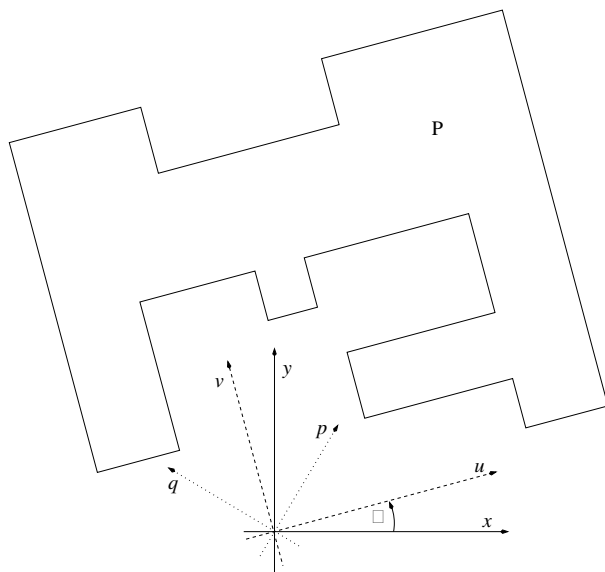
**Fig. 1.** Two given 5-gons have identical angles, but it is natural to expect that  $P$  should have a higher estimated rectilinearity than  $Q$ .

## 2 Definitions and Basic Statements

A polygon  $P$  is rectilinear if its interior angles belong to the set  $\{\pi/2, 3\pi/2\}$  (see Fig. 2 for an example). In this paper we define a rectilinearity measure for arbitrary polygons. Precisely, we define a mapping  $\mathcal{R}(P)$  which maps a given polygon  $P$  into a number from the interval  $(0, 1]$ .

We will use the following denotations (see Fig. 3 and Fig. 4 for some illustrations).

The set of all polygons will be denoted by  $\Pi$ . For a given  $n$ -gon  $P$  having vertices denoted by  $A_0, A_1, \dots, A_{n-1}, A_n = A_0$ , its edges will be denoted  $e_i = [A_{i-1}, A_i]$  for  $i = 1, 2, \dots, n$ . The Euclidean length of the straight line segment  $e = [(x_1, y_1), (x_2, y_2)]$  is  $l_2(e) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ , while the length of  $e$  according to the  $l_1$  metric is  $l_1(e) = |x_1 - x_2| + |y_1 - y_2|$ .



**Fig. 2.** For the given rectilinear 20-gon  $P$ , its  $l_1$  perimeter  $\mathcal{P}_1(P)$  has the minimum value if the coordinate axes are chosen to be parallel with  $u$  and  $v$ . This minimum value corresponds to  $\mathcal{P}_1(P, -\alpha)$  if  $x$  and  $y$  are taken to be the coordinate axes.

$\mathcal{P}_2(P)$  will denote the Euclidean perimeter of  $P$ , while  $\mathcal{P}_1(P)$  will denote the perimeter of  $P$  in the sense of  $l_1$  metrics. So,

$$\mathcal{P}_2(P) = \sum_{e_i \text{ is an edge of } P} l_2(e_i) \quad \text{and} \quad \mathcal{P}_1(P) = \sum_{e_i \text{ is an edge of } P} l_1(e_i).$$

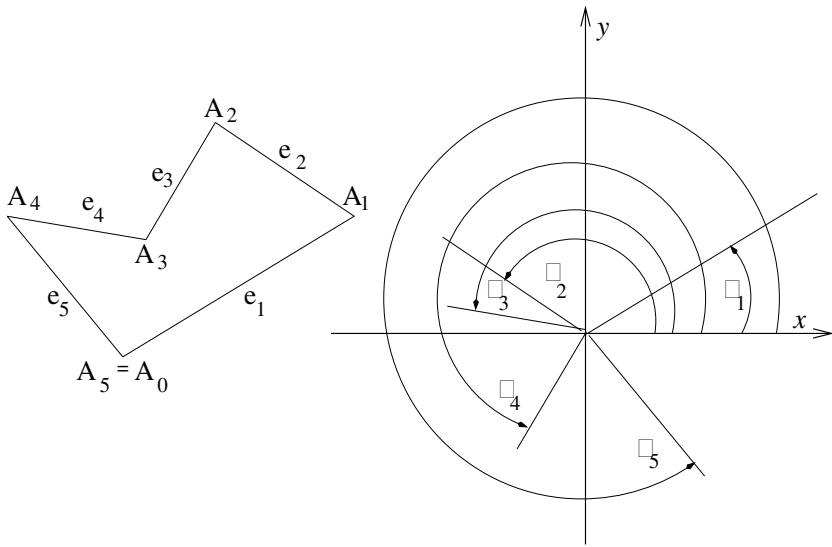
Since isometric polygons do not necessarily have the same perimeter under the  $l_1$  metric, we shall use  $\mathcal{P}_1(P, \alpha)$  for the  $l_1$  perimeter of the polygon which is obtained by the rotating  $P$  by the angle  $\alpha$  with the origin as the centre of rotation. If the same rotation is applied to the edge  $e$ , the  $l_1$  perimeter of the obtained edge will be denoted as  $l_1(e, \alpha)$ .

If the oriented angle between the positively oriented  $x$ -axis and the vector  $\overrightarrow{A_{i-1}A_i}$  is denoted by  $\phi_i$  ( $i = 1, 2, \dots, n$ ), then obviously  $l_1(e_i) = l_2(e_i) \cdot (|\cos \phi_i| + |\sin \phi_i|)$ , while  $l_1(e_i, \alpha) = l_2(e_i) \cdot (|\cos(\phi_i + \alpha)| + |\sin(\phi_i + \alpha)|)$ . Thus, by using  $1 \leq |\cos \beta| + |\sin \beta| \leq \sqrt{2}$  (for any  $\beta$ ), we have

$$\mathcal{P}_2(P) \leq \mathcal{P}_1(P, \alpha) = \sum_{i=1}^n l_2(e_i) \cdot (|\cos(\phi_i + \alpha)| + |\sin(\phi_i + \alpha)|) \leq \sqrt{2} \cdot \mathcal{P}_2(P). \quad (1)$$

We will exploit the following property of rectilinear polygons which is formulated as a theorem.

**Theorem 1.** *A given polygon  $P$  is rectilinear if and only if there exists a choice of the coordinate system such that the Euclidean perimeter of  $P$  and the  $l_1$  perimeter of  $P$  coincide, i.e.,*



**Fig. 3.** The used denotations are illustrated on an example of a given 5-gon.

$$\mathcal{P}_2(P) = \mathcal{P}_1(P, \alpha) \quad \text{for some } \alpha.$$

*Proof.* If the given polygon  $P$  is rectilinear, then a rotation of  $P$ , such that the edges of  $P$  become parallel to the coordinate axes, preserves the equality  $\mathcal{P}_2(P) = \mathcal{P}_1(P, \alpha)$  where  $\alpha$  is the rotation angle.

On the other hand, if  $\mathcal{P}_2(P) = \mathcal{P}_1(P, \alpha)$  then (by using (1) and  $|\cos \beta| + |\sin \beta| \geq 1$ ) it must be  $|\cos(\phi_i + \alpha)| + |\sin(\phi_i + \alpha)| = 1$  for any edge  $e_i$ ,  $1 \leq i \leq n$ , of the given  $n$ -gon  $P$ . That means (for any  $i$ ,  $1 \leq i \leq n$ ) it is either  $\cos(\phi_i + \alpha) = 0$  or  $\sin(\phi_i + \alpha) = 0$  – but it implies that all edges of  $P$  are either parallel or orthogonal to the same line. This completes the proof.  $\square$

### 3 The Basic Idea and Necessary Mathematics

Theorem 1 gives a useful characterisation of rectilinear polygons and gives the basic idea for the polygon rectilinearity measurement described in this paper. In the first stage, Theorem 1 together with  $\mathcal{P}_2(P) \leq \mathcal{P}_1(P)$  (see (1)) suggests that the ratio  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)}$  can be used as a rectilinearity measure for the polygon  $P$ .

Precisely, the ratio  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)}$  has the following “good” properties:

- a1) it is a strictly positive number;
- a2) it is defined for any polygon  $P$ ;
- a3) it can be calculated easily;
- a4) for any non rectilinear polygon it is strictly less than 1 and for any rectilinear polygon it is exactly 1 if the coordinate axes are suitably chosen.

But, on the other side  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)}$  has the following “bad” properties:

- b1) it is not invariant under similarity (even isometric) transformations;  
 b2) the infimum for the set of values of  $\mathcal{Q}(P, \alpha) = \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  is not zero; for an example, it can be seen easily (from (1)) that there no exists a polygon  $P$  such that  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)} \notin \left(0, \frac{\sqrt{2}}{2}\right)$ .

In this section we develop necessary mathematical tools in order to define a function  $\mathcal{R}(P)$  which satisfies a1)–a5) but not b1) and b2).

The problem described by b1) can be avoided by considering  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  instead of  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)}$ , but it opens the question of how to compute this maximum.

Further, the inequalities from (1) give

$$\frac{\sqrt{2}}{2} \leq \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)} \leq 1, \quad \text{and consequently,} \quad \frac{\sqrt{2}}{2} \leq \max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} \leq 1$$

for any polygon  $P$ . But, while the inequality  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P)} \leq 1$  is sharp, and moreover,

the equality  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} = 1$  is satisfied if and only if  $P$  is a rectilinear polygon (due to Theorem 1), it can be seen easily that there exists no polygon  $P$  such that  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} = \frac{\sqrt{2}}{2}$ . Namely, if an  $n$ -gon  $P$  satisfies the last

equality, then for some  $\alpha_0$  we have  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha_0)} = \frac{\sqrt{2}}{2}$  which (by (1)) would imply  $\sqrt{2} \cdot l_2(e_i) = l_1(e_i, \alpha_0)$  or, equivalently,  $\phi_i + \alpha_0 \in \{\pi/4, 3\pi/4, 7\pi/2, 9\pi/4\}$  for any edge  $e_i$  where  $1 \leq i \leq n$ . Thus,  $P$  must be rectilinear, and due to Theorem 1, the considered maximum must be equal to 1, which is a contradiction.

So, for our purpose it is necessary to determine the maximal possible  $\mu$  such that  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  belongs to the interval  $[\mu, 1]$  for any polygon  $P$ . The next two lemmas together show  $\mu = \frac{\pi}{4}$  and moreover, there is no polygon  $P$  such that  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} = \frac{\pi}{4}$ .

**Lemma 1.** *The inequality*

$$\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} > \frac{\pi}{4}$$

*holds for any polygon  $P$ .*



*Proof.* We prove the statement by a contradiction. Let us assume the contrary, i.e., there exists an  $n$ -gon  $P$  such that

$$\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} \leq \frac{\pi}{4}$$

for any choice of  $\alpha$ , or equivalently,

$$\frac{\mathcal{P}_1(P, \alpha)}{\mathcal{P}_2(P)} \geq \frac{4}{\pi} \quad \text{for any } \alpha \in [0, 2\pi].$$

Since  $\frac{\mathcal{P}_1(P, \alpha)}{\mathcal{P}_2(P)}$  is a continuous non constant function (for more details see Section 5) depending on  $\alpha$  (if  $P$  is fixed), and since it is assumed to be greater or equal to  $\frac{4}{\pi}$  it must be

$$\int_0^{2\pi} \frac{\mathcal{P}_1(P, \alpha)}{\mathcal{P}_2(P)} \cdot d\alpha > \int_0^{2\pi} \frac{4}{\pi} \cdot d\alpha = 8. \quad (2)$$

By using (2) we have:

$$\begin{aligned} 8 &< \int_0^{2\pi} \frac{\mathcal{P}_1(P, \alpha)}{\mathcal{P}_2(P)} \cdot d\alpha \\ &= \frac{1}{\mathcal{P}_2(P)} \cdot \int_0^{2\pi} \left( \sum_{i=1}^n l_1(e_i, \alpha) \right) \cdot d\alpha \\ &= \frac{1}{\mathcal{P}_2(P)} \cdot \sum_{i=1}^n \left( \int_0^{2\pi} l_1(e_i, \alpha) \cdot d\alpha \right) \\ &= \frac{1}{\mathcal{P}_2(P)} \cdot \sum_{i=1}^n \left( \int_0^{2\pi} l_2(e_i) \cdot (|\sin(\phi_i + \alpha)| + |\cos(\phi_i + \alpha)|) \cdot d\alpha \right) \\ &= \frac{1}{\mathcal{P}_2(P)} \cdot \left( \sum_{i=1}^n 8 \cdot l_2(e_i) \right) = \frac{1}{\mathcal{P}_2(P)} \cdot \mathcal{P}_2(P) \cdot 8 = 8. \end{aligned}$$

The contradiction  $8 < 8$  finishes the proof.  $\blacksquare$

So, in accordance with the above discussion, Lemma 1 shows that the required number  $\mu$  is not smaller than  $\frac{\pi}{4}$ . The next lemma shows that  $\mu$  is not bigger than  $\frac{\pi}{4}$ . Thus, both lemmas together give  $\mu = \frac{\pi}{4}$ .

**Lemma 2.** *The infimum for the set of values  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  for all possible choices of polygon  $P$  is  $\frac{\pi}{4}$ , i.e.,*

$$\inf_{P \in \Pi} \left\{ \max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} \right\} = \frac{\pi}{4}$$

*Proof.* To prove the statement it is enough to find a sequence of polygons  $P_3, P_4, P_5, \dots$  such that

$$\lim_{n \rightarrow \infty} \left( \max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P_n)}{\mathcal{P}_1(P_n, \alpha)} \right) = \frac{\pi}{4}.$$

We will prove that the sequence of regular  $n$ -gons  $P_n$  inscribed into the unit circle satisfies the previous equality. Namely, it can be seen easily that the sequence of the Euclidean perimeters of  $P_n$  tends to the perimeter of the unit circle, i.e.,

$$\lim_{n \rightarrow \infty} \mathcal{P}_2(P_n) = 2 \cdot \pi \quad (3)$$

but also

$$\lim_{n \rightarrow \infty} \mathcal{P}_1(P_n, \alpha) = 8 \quad (4)$$

holds independently on the choice of  $\alpha$ . Precisely, if it is considered that (for any  $\alpha \in [0, 2\pi]$ ) the  $l_1$  perimeter  $\mathcal{P}_1(P_n, \alpha)$  equals the sum of the projections of all edges of  $P_n$  onto  $x$  and  $y$  axes, than it follows that this sum tends to 8 as  $n \rightarrow \infty$  (see Fig. 4 for an illustration). Since the limits in (3) and (4) are independent on  $\alpha$  we have

$$\lim_{n \rightarrow \infty} \frac{\mathcal{P}_2(P_n)}{\mathcal{P}_1(P_n, \alpha)} = \lim_{n \rightarrow \infty} \left( \max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P_n)}{\mathcal{P}_1(P_n, \alpha)} \right) = \frac{\pi}{4},$$

what finishes the proof.  $\square$

## 4 A Rectilinearity Measure

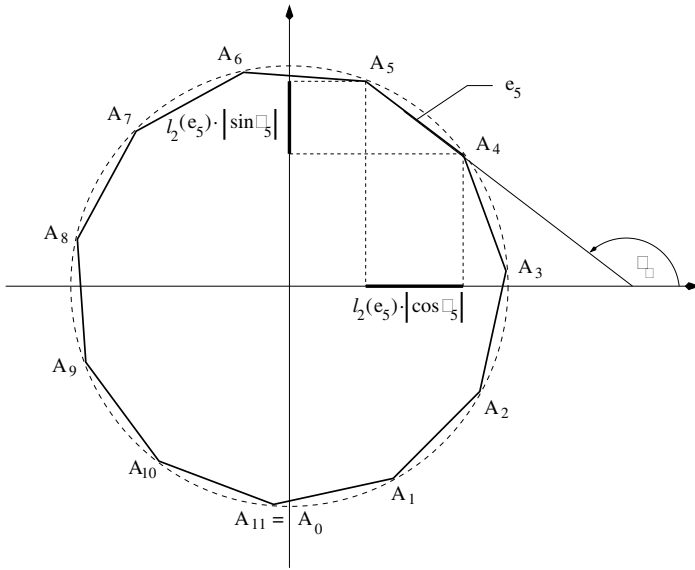
Motivated by the proved properties of the function  $\max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  we give the following definition for the new rectilinearity measurement of polygons.

**Definition 1.** *For an arbitrary polygon  $P$  we define its rectilinearity  $\mathcal{R}(P)$  as*

$$\mathcal{R}(P) = \frac{4}{4 - \pi} \cdot \left( \max_{\alpha \in [0, 2\pi]} \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)} - \frac{\pi}{4} \right).$$

The following theorem summarises the properties of the polygon rectilinearity measure proposed here.

**Theorem 2.** *For any polygon  $P$ , we have:*



**Fig. 4.** A regular 11-gon inscribed into the unit circle with the centre placed at the origin.

- i)  $\mathcal{R}(P)$  is well defined and  $\mathcal{R}(P) \in (0, 1]$ ;
- ii)  $\mathcal{R}(P) = 1$  if and only if  $P$  is rectilinear;
- iii)  $\inf_{P \in \Pi} (\mathcal{R}(P)) = 0$ ;
- iv)  $\mathcal{R}(P)$  is invariant under similarity transformations.

*Proof.* For a fixed  $P$ ,  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  is a continuous function depending on  $\alpha$  (for more details see the next section). Consequently, it must reach its minimum on the closed interval  $[0, 2\pi]$ , i.e.,  $\mathcal{R}(P)$  is well defined.  $\mathcal{R}(P) \in (0, 1]$  follows from (1), Definition 1, and Lemma 1.

The item *ii*) is a direct consequence of Theorem 1.

The item *iii*) is the statement of Lemma 2.

To prove *iv*) let us notice that  $\mathcal{R}(P)$  is invariant under all isometric transformations – which follows from the definition. Also,  $\frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  and consequently  $\mathcal{R}(P)$  are invariant under any transformation of the form  $(x, y) \rightarrow (\lambda \cdot x, \lambda \cdot y)$  for any choice of  $\lambda \neq 0, P$ , and  $\alpha$ . That completes the proof.  $\blacksquare$

Some examples of polygons with their computed rectilinearity are given on Fig. 5.

## 5 Computation of $\mathcal{R}(P)$

The question which remains open is how to compute  $\mathcal{R}(P)$  effectively if a polygon  $P$  is given. Since  $\mathcal{P}_2(P)$  can be easily calculated from the vertices of  $P$  it remains to describe the computation of the minimum value of  $\mathcal{P}_1(P, \alpha)$  when  $\alpha$  varies from 0 to  $2\pi$ . In this section we describe a procedure for such a computation.

Let us consider an edge  $e_i$  ( $1 \leq i \leq n$ ) of a given  $n$ -gon  $P$ . Trivially,

$$l_1(e_i, \alpha) = \begin{cases} +l_2(e_i) \cdot \cos(\phi_i + \alpha) + l_2(e_i) \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [-\phi_i, \frac{\pi}{2} - \phi_i] \\ -l_2(e_i) \cdot \cos(\phi_i + \alpha) + l_2(e_i) \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\frac{\pi}{2} - \phi_i, \pi - \phi_i] \\ -l_2(e_i) \cdot \cos(\phi_i + \alpha) - l_2(e_i) \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\pi - \phi_i, \frac{3\pi}{2} - \phi_i] \\ +l_2(e_i) \cdot \cos(\phi_i + \alpha) - l_2(e_i) \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\frac{3\pi}{2} - \phi_i, 2\pi - \phi_i]. \end{cases}$$

Consequently, there is an integer  $k \leq 4 \cdot n$  and an increasing sequence of angles  $0 \leq \alpha_1 < \alpha_2 < \dots < \alpha_k \leq 2\pi$  such that

$$\mathcal{P}_1(P, \alpha) = \begin{cases} \sum_{i=1}^n a_{1,i} \cdot \cos(\phi_i + \alpha) + b_{1,i} \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\alpha_1, \alpha_2] \\ \sum_{i=1}^n a_{2,i} \cdot \cos(\phi_i + \alpha) + b_{2,i} \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\alpha_2, \alpha_3] \\ \dots\dots\dots \\ \sum_{i=1}^n a_{k-1,i} \cdot \cos(\phi_i + \alpha) + b_{k-1,i} \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\alpha_{k-1}, \alpha_k] \\ \sum_{i=1}^n a_{k,i} \cdot \cos(\phi_i + \alpha) + b_{k,i} \cdot \sin(\phi_i + \alpha) & \text{if } \alpha \in [\alpha_k, 2\pi + \alpha_1], \end{cases}$$

where

$$\{a_{i,j}, b_{i,j} \mid 1 \leq i \leq k, 1 \leq j \leq k\} \subset \{+l_2(e_i), -l_2(e_i)\},$$

or precisely, for any  $i \in \{1, 2, \dots, n\}$  and any  $j \in \{1, 2, \dots, k\}$

$$a_{j,i} = +l_2(e_i) \quad \text{if} \quad \cos(\phi_i + \alpha) > 0 \quad \text{for } \alpha \in (\alpha_j, \alpha_{j+1}) \quad (5)$$

$$a_{j,i} = -l_2(e_i) \quad \text{if} \quad \cos(\phi_i + \alpha) < 0 \quad \text{for } \alpha \in (\alpha_j, \alpha_{j+1}) \quad (6)$$

and analogously,

$$b_{j,i} = +l_2(e_i) \quad \text{if} \quad \sin(\phi_i + \alpha) > 0 \quad \text{for } \alpha \in (\alpha_j, \alpha_{j+1}) \quad (7)$$

$$b_{j,i} = -l_2(e_i) \quad \text{if} \quad \sin(\phi_i + \alpha) < 0 \quad \text{for } \alpha \in (\alpha_j, \alpha_{j+1}) \quad (8)$$

What is important for us is

$$0 < \mathcal{P}_1(P, \alpha) = -\mathcal{P}_1''(P, \alpha) \quad \text{if} \quad \alpha \notin \{\alpha_1, \alpha_2, \dots, \alpha_k\}$$

which implies that  $\mathcal{P}_1(P, \alpha)$  does not have any local minimum inside the open intervals

$$(\alpha_1, \alpha_2), (\alpha_2, \alpha_3), \dots, (\alpha_{k-1}, \alpha_k), (\alpha_k, 2\pi + \alpha_1)).$$

In other words we have

$$\min_{\alpha \in [0, 2\pi]} \mathcal{P}_1(P, \alpha) = \min \{\mathcal{P}_1(P, \alpha_1), \mathcal{P}_1(P, \alpha_2), \dots, \mathcal{P}_1(P, \alpha_k)\}. \quad (9)$$

Now, we are able to describe the following simple procedure for computing  $\mathcal{R}(P)$ .

**PROCEDURE**  $\mathcal{R}(P)$  *Computation*

**Input:** The vertices  $A_1, A_2, \dots, A_n$  of a given  $n$ -gon  $P$ .

**1. Step.**

For any  $i$ ,  $i = 1, 2, \dots, n$ , compute the angle  $\phi_i$ , and compute the absolute values of the following angle differences:

$$|\phi_i - \frac{\pi}{2}|, |\phi_i - \pi|, |\phi_i - \frac{3\pi}{2}|, |\phi_i - 2\pi|.$$

Sort the above angle-values in the increasing order and denote the sorted values as:  $0 \leq \alpha_1 \leq \alpha_2 < \dots < \alpha_k \leq 2\pi$ .

**2. Step.**

Assign either  $+l_2(e_i)$  or  $-l_2(e_i)$  to  $a_{i,j}$  and  $b_{i,j}$  for any  $i$  from  $\{1, 2, \dots, n\}$  and any  $j$  from  $\{1, 2, \dots, k\}$   
( $\star$  in accordance with (5)-(8)  $\star$ ).

**3. Step.**

Compute  $\min_{\alpha \in [0, 2\pi]} \mathcal{P}_1(P, \alpha)$  as the minimum of the finite number point set

$$\{\mathcal{P}_1(P, \alpha_1), \mathcal{P}_1(P, \alpha_2), \dots, \mathcal{P}_1(P, \alpha_{k-1}), \mathcal{P}_1(P, \alpha_k)\}$$

( $\star$  in accordance with (9)  $\star$ )

then compute  $\mathcal{R}(P)$  as

$$\mathcal{R}(P) := \frac{4}{4 - \pi} \cdot \left( \frac{\mathcal{P}_2(P)}{\min_{\alpha \in [0, 2\pi]} \mathcal{P}_1(P, \alpha)} - \frac{\pi}{4} \right)$$

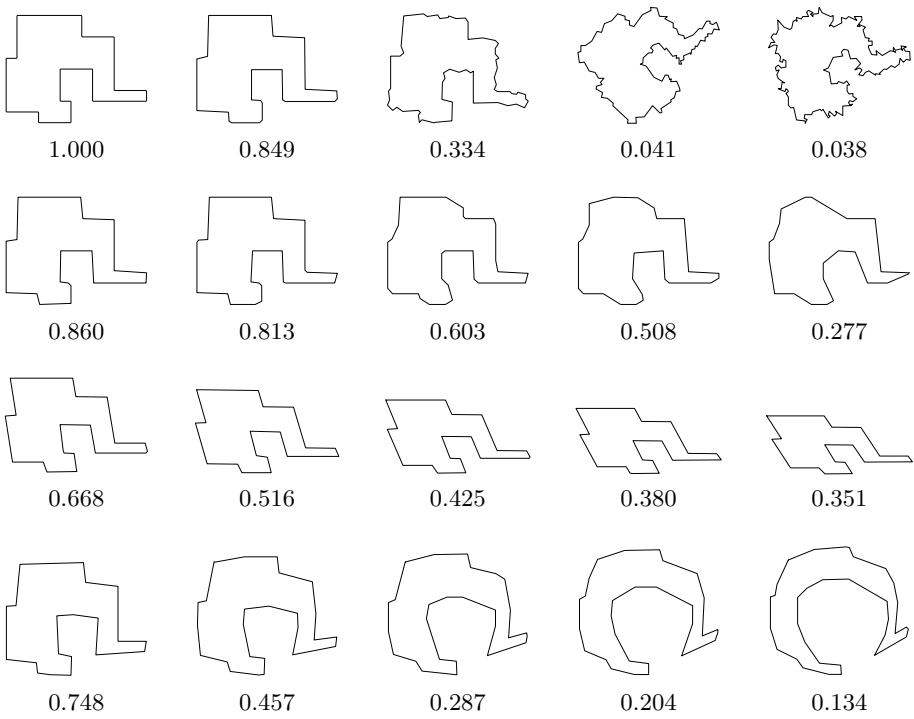
( $\star$  in accordance with Definition 1  $\star$ ).

**Output:**  $\mathcal{R}(P)$ .

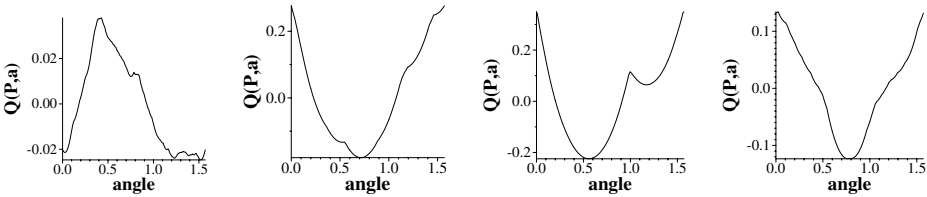
## 6 Some Examples and Concluding Remarks

The rectilinearity measure is applied to a (perfect) rectilinear polygon in the top left hand polygon in Fig. 5 which is then degraded in various ways. The first row demonstrates the effect of increasing levels of local noise applied to the polygon's vertices. In the second row the polygon is edited, eliminating vertices, which effectively rounds corners and increases its convexity. A shearing transformation is applied in the third row. Finally, the polygon is warped, and the axis aligned edges are increasingly rounded. All examples show that the rectilinearity measure is well behaved; increasing distortion consistently decreases the computed value.

Note also that the orientations that maximised  $\mathcal{Q}(P, \alpha) = \frac{\mathcal{P}_2(P)}{\mathcal{P}_1(P, \alpha)}$  match our



**Fig. 5.** Examples of polygons with their rectilinearity measured as proposed in this paper. Polygons are rotated to the orientations that maximised  $Q(P, \alpha)$ .



**Fig. 6.** Plots of normalised  $Q(P, \alpha)$  for each of the rightmost examples in Fig. 5.

expectations except at high noise levels when the rectilinearity measure has dropped close to zero. For each of the maximally degraded polygons (i.e. the rightmost examples in each row) Fig. 6 plots  $Q(P, \alpha)$ . It can be seen that it is well behaved and, despite the effects of noise and other distortions which introduce local maxima, the main peak remains distinct.

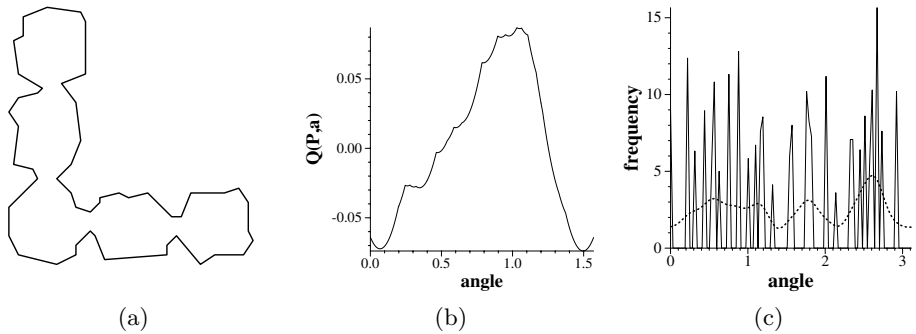
A second example is shown, working this time with real data from a Digital Elevation Model (DEM), i.e. a depth map (see Fig 7 (a)). Some simple noise filtering and segmentation techniques were applied to produce a set of polygons. These are further processed using Ramer’s line simplification algorithm [10] to reduce the effects of quantisation since at a pixel level a dense 4-connected curve



**Fig. 7.** (a) A DEM of Hannover; (b) segmented regions with filled intensities proportional to rectilinearity

will be perfectly rectilinear! Fig 7 (b) plots the regions filled with intensities proportional to their rectilinearity; thus rectilinear shapes should appear bright. A casual glance seems to indicate that in some cases the computed rectilinearity does not correspond well with human perception. As an example, the polygon making up the central building in Fig 7 (a) only receives a value of 0.087 although it initially looks approximately rectilinear. Nevertheless, when plotted again in Fig. 8 (a) at the orientation maximising  $Q(P, \alpha)$  the coordinate frame can be seen to be correctly determined. Moreover, the plot of  $Q(P, \alpha)$  (Fig. 8 (b)) contains only the single main peak. In fact, the problem arises from errors in the boundary due to the various steps in data processing. The deviations of local edge orientation are substantial as demonstrated by a histogram of their orientation weighted by edge length. Fig. 8 (c) shows the histogram at two bin sizes, and it can be seen that the distribution is fairly uniform, therefore producing a low rectilinearity value.

As a possible means of reconciling this inconsistency between our perceptual and measured evaluation of rectilinearity we note that relatively small adjustments to the polygon could greatly increase the values of  $Q(P)$ . Thus we intend to apply such a process, moving vertices such that they maximise  $Q(P)$  while minimising the shift in vertex location. A similar scheme of “shape correction”



**Fig. 8.** (a) An example building rotated to maximise  $Q(P, \alpha)$ ; (b) the corresponding plot of  $Q(P, \alpha)$ ; (c) histogram of length weighted edge orientation of (a).

has already been applied by Brunn *et al.* [1] who applied an iterative MDL vertex shifting and removal scheme, and by Mayer [7] who used constrained snakes.

## References

1. A. Brunn, U. Weidner, and W. Förstner. Model-based 2d-shape recovery. In G. Sagerer *et al.*, editor, *Mustererkennung*, pages 260–268. Springer-Verlag, 1995.
2. R.T. Collins, C.O. Jaynes, Y.Q. Cheng, X.G. Wang, F.R. Stolle, E.M. Riseman, and A.R. Hanson. The ascender system: Automated site modeling from multiple aerial images. *Computer Vision and Image Understanding*, 72(2):143–162, 1998.
3. S. Hyde *et al.* *The Language of Shape*. Elsevier, 1997.
4. R.M. Haralick. A measure for circularity of digital figures. *IEEE Transactions on Systems, Man and Cybernetics*, 4:394–396, 1974.
5. R.B. Irvin and D.M. McKeown. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1564–1575, 1989.
6. Y.T. Liow and T. Pavlidis. Use of shadows for extracting buildings in aerial images. *Computer Vision, Graphics and Image Processing*, 49(2):242–277, 1990.
7. S. Mayer. Constrained optimization of building contours from high-resolution ortho-images. In *Proc. Int. Conf. Image Processing*, 2001.
8. S. Noronha and R. Nevatia. Detection and modeling of buildings from multiple aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):501–518, 2001.
9. D. Proffitt. The measurement of circularity and ellipticity on a digital grid. *Pattern Recognition*, 15(5):383–387, 1982.
10. U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1:244–256, 1972.
11. P.L. Rosin. Measuring rectangularity. *Machine Vision and Applications*, 11:191–196, 1999.
12. M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Chapman and Hall, 1993.
13. J. Willats. *Art and Representation: New Principles in the Analysis of Pictures*. Princeton University Press, 1997.



# Local Analysis for 3D Reconstruction of Specular Surfaces – Part II

Silvio Savarese and Pietro Perona

California Institute of Technology, Pasadena CA 91200, USA

**Abstract.** We analyze the problem of recovering the shape of a mirror surface. We generalize the results of [1], where the special case of planar and spherical mirror surfaces was considered, extending that analysis to any smooth surface. A calibrated scene composed of lines passing through a point is assumed. The lines are reflected by the mirror surface onto the image plane of a calibrated camera, where the intersection and orientation of such reflections are measured. The relationship between the local geometry of the surface around the point of reflection and the measurements is analyzed. We give necessary and sufficient conditions, as well as a practical algorithm, for recovering first order local information (positions and normals) when three intersecting lines are visible. A small number of ‘ghost solutions’ may arise. Second order surface geometry may also be obtained up to one unknown parameter. Experimental results with real mirror surfaces are presented.

**Keywords:** Shape recovery, geometry, mirror surfaces.

## 1 Introduction and Motivation

We are interested in the possibility of recovering information on the shape of a surface from the specular component of its reflectance function. Since we wish to ignore the contributions of shading and texture, we will study surfaces that are perfect mirrors. A curved mirror surface produces ‘distorted’ images of the surrounding world. For example, the image of a straight line reflected by a curved mirror is, in general, a curve (see Fig. 1). It is clear that such distortions are systematically related to the shape of the surface. Is it possible to invert this map, and recover the shape of the mirror from the images it reflects? The general ‘inverse mirror’ problem is clearly underconstrained: by opportune manipulations of the surrounding world we may produce almost *any* image from *any* curved mirror surface as illustrated by the anamorphic images that were popular during the Renaissance. The purpose of this paper is to continue the investigation started in our previous work [1] where we presented a novel study on the basic geometrical principles linking the shape of a mirror surface to the distortions it produces on a scene. We assumed a calibrated world composed of the simplest primary structures: one point and one or more lines through it. We studied the relationship between the local geometry of the mirror surface around the point of reflection, and the position, orientation and curvature of the reflected images



**Fig. 1.** M.C. Escher (1935): Still Life with Spherical Mirror

of such point and lines. Additionally, we derived an explicit solution for planar and spherical surfaces. In this paper we extend this analysis to generic smooth surfaces. We show that it is possible to recover first order local information (positions and normals) when three intersecting lines are reflected by the surface, although a small number of “ghost” solutions in the reconstruction may arise. Such solutions might be removed by considering either more than 3 no-coplanar lines or a rough a priori estimate of the surface point location or a second order local differential analysis as derived in [1]. Second order surface geometry may also be obtained up to one unknown parameter, which we prove cannot be recovered from first order local measurements (position and tangents).

Applications of our work include recovering the global shape of highly glossy surfaces. Two possible situations are: a) placing a suitable calibrated pattern of intersecting lines near the specular surface and applying our analysis at the locus of the observed reflections of the pattern intersections; b) placing a calibrated reference plane near the specular surface, projecting a suitable pattern with a calibrated LCD projector over the specular surface and applying our analysis at the locus of the intersections reflected by the surface over the reference plane and observed by the camera [14]; such setup is appealing since it requires the same hardware used by common structured lighting techniques. Finally our work may provide useful mathematical tools for the analysis and the calibration of omniview cameras with curved surfaces mirrors.

A summary of the notation and results obtained in [1] is presented in Sec. 2. Main geometrical properties and the reconstruction method for general mirror surfaces are described in Sec. 3. Experimental results with real mirror surface are shown in Sec. 4. The paper is concluded with a discussion on our findings and a number of issues for further research.

### 1.1 Previous Work

Previous authors have used highlights as a cue to infer information about the geometry of a specular surface. Koenderink and van Doorn [10] qualitatively de-

scribed how pattern of specularities change under viewer motion. This analysis was extended by Blake *et al.* and incorporated in a stereoscopic vision framework [4] [3]. Additionally, Zisserman *et al.* [13] investigated what geometrical information can be obtained from tracked motion of specularities. Other approaches were based on mathematical models of specular reflections (e.g. reflectance maps) [8] or extension of photometric stereo models [9]. Oren and Nayar developed in [11] an analysis on classification of real and virtual features and an algorithm recovering the 3D surface profiles traveled by virtual features. Zheng and Murata developed a system [12] where extended lights illuminate a rotating specular object whose surface is reconstructed by analyzing the motion of the highlight stripes. In [7], Halsead *et al.* proposed a reconstruction algorithm where a surface global model is fitted to a set of normals by imaging a pattern of light reflected by specular surface. Their results were applied in the interactive visualization of the cornea. Finally, Perard [14] used a structured lighting technique for the iterative reconstruction of surface normal vectors and topography.

Contrary to previous techniques, in our method, surrounding world and viewer are assumed to be static. Monocular images rather than stereo pairs are needed for the reconstruction. The analysis is local and differential rather than global and algebraic.

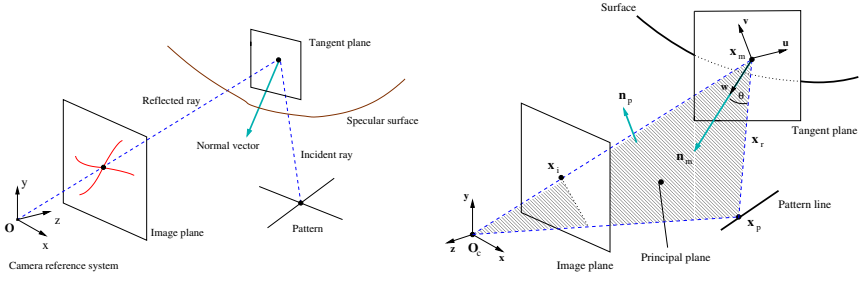
## 2 The Geometry of the Specular Reflections

Our goal is to obtain local geometrical information about an unknown smooth mirror surface. The basic geometric setup is depicted in Fig. 2 (left panel). A calibrated pattern is reflected by a curved mirror surface and the reflection is observed by a calibrated camera. The pattern may be formed by either one point or one point and one line, or 2 (or more) intersecting lines. We start our analysis studying which local information about the surface can be obtained by considering a single pattern point and its corresponding image reflection. We begin with a summary of notation and results in [1].

### 2.1 Definitions and Basic Specular Reflection Constraints

A point (or a vector) in the 3D space is expressed by a column 3-vector and is denoted by a bold letter (e.g.  $\mathbf{x} = (x y z)^T$ ). A vector whose norm is 1 is denoted by a bold letter with hat (e.g.  $\hat{\mathbf{n}}$ ). A coordinate reference system  $[XYZ]$  is chosen with origin  $\mathbf{O}_c$  in the center of projection of the camera. See Fig. 2 (right panel). Let  $\mathbf{x}_p$  be the pattern point.  $\mathbf{x}_i$  denotes the image of  $\mathbf{x}_p$  reflected by the surface and  $\mathbf{x}_m$  denotes the corresponding reflection point on the mirror surface. Since the camera and pattern are calibrated,  $\mathbf{x}_p$  and  $\mathbf{x}_i$  are known, whereas  $\mathbf{x}_m$  is unknown. The normal to the surface in  $\mathbf{x}_m$  is indicated by  $\hat{\mathbf{n}}_m$  and is unknown as well. Let us call *principal plane* the plane defined by  $\mathbf{x}_i$ ,  $\mathbf{x}_p$  and  $\mathbf{O}_c$  (dashed area in Fig. 2 – right panel) and let  $\hat{\mathbf{n}}_p$  be its normal vector. Hence  $\hat{\mathbf{n}}_p$  is a known quantity. The geometry of our setup satisfies 3 basic constraints: **1)** The perspective projection constraint: the point  $\mathbf{x}_m$  must belong to the line defined by  $\mathbf{O}_c$  and  $\mathbf{x}_i$ , namely,

$$\mathbf{x}_m = s\hat{\mathbf{x}}_i \quad (1)$$



**Fig. 2.** Left panel: the basic setup. Right panel: the geometry

where  $s$  is the distance between the center of the camera  $\mathbf{O}_c$  and  $\mathbf{x}_m$ . As a result,  $\mathbf{x}_m$  is known up to a scalar factor. **2)** The incident vector  $\mathbf{x}_m - \mathbf{x}_p$  and the reflected vector  $\mathbf{x}_m - \mathbf{O}_c$  must belong to the same plane, that is, the principal plane. **3)** The angle between incident vector and normal vector must be equal to the angle between reflected vector and normal vector. By combining such constraints it is straightforward to conclude that  $\mathbf{n}_m$  and reflection angle  $\theta$  are parametrized by  $s$  as follows:

$$\mathbf{n}_m = [\hat{\mathbf{x}}_i - \frac{(s\hat{\mathbf{x}}_i - \mathbf{x}_p)}{\|s\hat{\mathbf{x}}_i - \mathbf{x}_p\|}] \times \hat{\mathbf{n}}_p \quad (2)$$

$$\cos \theta = \frac{\sqrt{2}}{2} \sqrt{\frac{s - \hat{\mathbf{x}}_i^T \mathbf{x}_p}{\|s\hat{\mathbf{x}}_i - \mathbf{x}_p\|} + 1} \quad (3)$$

See [1] for a derivation of these equations.

## 2.2 The Pattern Line Constraint

Since our goal is to obtain local geometrical information about the mirror surface at the reflection point  $\mathbf{x}_m$ , as first attempt, we would like to compute the unknown parameter  $s$ . We notice that, if  $s$  were known, by means of Eq. 1 and Eq. 2, the surface point  $\mathbf{x}_m$  and the surface normal vector  $\hat{\mathbf{n}}_m$  would be known as well. Thus, as to the first order surface description, the local geometry would be fully recovered. It is clear that a further constraint is needed. To this end, we consider one pattern line through  $\mathbf{x}_p$ . The pattern line reflected by the mirror surface can be captured by the camera and the tangent direction of such observed curve at  $\mathbf{x}_i$  can be measured. Before investigating how to exploit such measurement we first introduce further geometrical objects.

A more suitable coordinate reference system  $[UVW]$ , which we call *principal reference system* (see Fig. 2 – right panel) was first introduced by Blake in [3]. The principal reference system is centered in  $\mathbf{x}_m$ ; the  $\hat{\mathbf{w}}$  axis is coincident with  $\hat{\mathbf{n}}_m(s)$ ; the  $\hat{\mathbf{v}}$  axis is coincident with  $\hat{\mathbf{n}}_p$ ; the  $\hat{\mathbf{u}}$  axis is given by  $\hat{\mathbf{u}} = \hat{\mathbf{v}} \times \hat{\mathbf{w}}$ . Thus, a point  $\mathbf{x}$  in the  $[XYZ]$  and the corresponding point  $\mathbf{x}'$  in  $[UVW]$  are related by transformation  $\mathbf{x}' = \mathbf{R}^T(\mathbf{x} - \mathbf{T})$ , where  $\mathbf{R}(s) = [\hat{\mathbf{n}}_p \times \hat{\mathbf{n}}_m(s) \quad \hat{\mathbf{n}}_p \quad \hat{\mathbf{n}}_m(s)]$  and  $\mathbf{T}(s) = s\hat{\mathbf{x}}_i$ . For instance, the center of the camera becomes  $-\mathbf{R}(s)^T \mathbf{T}(s)$ . Notice that the transformation is function of  $s$ . From now on, we shall always

omit  $s$  from the notation (unless we need to show explicitly such dependency) and assume that we work in the principal reference system.

The pattern is formed by one point and one line passing through it. Let  $\mathbf{x}_{po}$  be such a point and  $\Delta\mathbf{p} = [\Delta p_u \ \Delta p_v \ \Delta p_w]^T$  the orientation vector of the line in space. We can describe the generic pattern line in parametric form as follows:

$$\mathbf{x}_p(t) = \mathbf{x}_{po} + t\Delta\mathbf{p} \quad (4)$$

where  $t$  is a parameter. Since the pattern is calibrated,  $\mathbf{x}_{po}$  and  $\Delta\mathbf{p}$  are known quantities in the  $[XYZ]$  reference system, whereas they become function of  $s$  in the  $[UVW]$  reference system.

In general, the mirror surface can be implicitly described by an equation  $g(x, y, z) = 0$ . Since we are interested in analyzing the surface locally, we can consider the corresponding *Monge representation* of the surface; that is, the surface can be described by the graph  $z = G(x, y)$ . In the principal reference system, the normal of the surface at the origin is  $\hat{\mathbf{w}}$  and the tangent plane to the surface at the origin is the plane defined by  $\hat{\mathbf{u}}$  and  $\hat{\mathbf{v}}$ . Therefore the equation of the surface around  $\mathbf{x}_m$  can be written in the *special Monge form* [6] as follows:

$$w = \frac{1}{2!}(au^2 + 2cuv + bv^2) + \frac{1}{3!}(eu^3 + 3fu^2v + 3guv^2 + hv^3) + \dots \quad (5)$$

Notice that the parameters  $a, b, c, \dots$  of the Monge form are unknown, since we do not have any information about the mirror surface.

Let us define a mapping function  $\mathbf{f}$  which maps a point  $\mathbf{x}_p$  (within the pattern line) into the corresponding reflection point  $\mathbf{x}_m$  in the mirror surface, given a fixed observer  $\mathbf{O}_c$ . Since  $\mathbf{x}_p$  is constrained to belong to the parametrized pattern line, the mapping can be expressed as follows:

$$\mathbf{f} : t \in \mathcal{R} \rightarrow \mathbf{x}_m \in \mathcal{R}^3 \quad (6)$$

In other words, Eq. 6 defines a parametrized space curve  $\mathbf{f}(t)$  lying within the mirror surface which describes the position of the reflection point  $\mathbf{x}_m$ , as  $t$  varies. When  $t = t_o = 0$ ,  $\mathbf{x}_m = \mathbf{x}_{m_o} = \mathbf{f}(t_o)$ , namely, the origin of the principal reference system. The pattern line, reflected by the mirror surface, is imaged as a curve line in the image plane. We call such curve  $\mathbf{f}_i(t)$ .  $\mathbf{f}_i(t)$  is essentially the perspective projection of  $\mathbf{f}(t)$  onto the image plane. Let  $\mathbf{x}_{io}$  be the perspective projection of  $\mathbf{x}_{m_o}$  onto the image plane. Let  $\mathbf{t}_o = [\dot{u}_o \ \dot{v}_o \ \dot{w}_o]^T$  and  $\hat{\mathbf{t}}_{io}$  be the tangent vectors of the curves  $\mathbf{f}(t)$  and  $\mathbf{f}_i(t)$  at  $t_o$  respectively.

It is not difficult to show (see [1]) that  $\hat{\mathbf{t}}_{io}$  and  $\hat{\mathbf{t}}_o$  are linked by the following relationship:

$$\hat{\mathbf{t}}_o = \frac{\hat{\mathbf{n}}_m \times (\hat{\mathbf{O}}_c \times \hat{\mathbf{t}}_{io})}{\|\hat{\mathbf{n}}_m \times (\hat{\mathbf{O}}_c \times \hat{\mathbf{t}}_{io})\|} \quad (7)$$

Thus, since  $\hat{\mathbf{t}}_{io}$  can be measured,  $\hat{\mathbf{t}}_o$  turns out to be known, up to  $s$ .

We present now the fundamental relationship between  $\hat{\mathbf{t}}_o$ , the geometry of the pattern line, the center of the camera  $\mathbf{O}_c$ , the reflection point  $\mathbf{x}_{m_o}$  and the parameters of the Taylor expansion of the surface Monge form  $G$ . Introducing

the problem as Chen and Arvo did in [5] and following the analysis described in [1], we obtain:

$$\tan \varphi = \frac{(J_u - 2a \cos \theta)B_v + 2cB_u \cos \theta}{(J_v - 2b \cos \theta)B_u + 2cB_v \cos \theta} \quad (8)$$

where,

$$\begin{aligned} B_v &= -\frac{\Delta p_v}{\|\mathbf{x}_{po}\|} & B_u &= \frac{1}{\|\mathbf{x}_{po}\|} (\Delta p_w \cos \theta \sin \theta - \Delta p_u \cos^2 \theta) \\ J_u &= \cos^2 \theta \frac{s + \|\mathbf{x}_{po}\|}{s \|\mathbf{x}_{po}\|} & J_v &= \frac{s + \|\mathbf{x}_{po}\|}{s \|\mathbf{x}_{po}\|} & \tan \varphi &= \frac{\dot{v}_o}{\dot{u}_o} \end{aligned} \quad (9)$$

Notice that  $\theta$ ,  $\|\mathbf{x}_{po}\|$ ,  $\Delta \mathbf{p} = [\Delta p_u \Delta p_v \Delta p_w]^T$  depend upon  $s$ ; the angle  $\varphi$  (namely, the orientation of  $\hat{\mathbf{t}}_o$  in the surface tangent plane at  $\mathbf{x}_{mo}$ ) can be expressed as function of  $s$  and  $\hat{\mathbf{t}}_{io}$  by means of Eq. 7;  $a$ ,  $b$  and  $c$  are second order parameters of the Taylor expansion of the surface Monge form (Eq. 5). Also, notice that no third and higher parameters of the Taylor expansion do appear in Eq. 8. Finally, we highlight that no assumption on the type of surface have been made, namely, Eq. 7 is valid for both concave or convex surfaces.

As a conclusion, Eq. 8 represents the constraint introduced by one pattern line passing through  $\mathbf{x}_{po}$  and the tangent vector measurement  $\hat{\mathbf{t}}_{io}$ . However, since in Eq. 8 there appear four unknowns ( $s$ ,  $a$ ,  $b$ ,  $c$ ) rather than just  $s$ , the reconstruction problem must be solved by jointly estimating both first and second order parameters and by using more than one pattern line.

### 3 Recovery of the Surface

As shown in [1], in the case of spherical mirror surface, we carried out an explicit solution for the distance  $s$  and the sphere curvature by means of Eq. 8 and by imposing that  $a = b$  and  $c = 0$ . In the following sections we investigate the more general case when  $s$ ,  $a$ ,  $b$ ,  $c$  are fully unknown. In Sec. 3.1 and 3.2, we assume that  $s$  is known and we analyze geometrical properties of the second order surface parameters. In Sec. 3.3 we explicitly describe how to estimate  $s$ .

#### 3.1 Analysis of Second Order Surface Parameters

Let us assume that the distance  $s$  is known. As shown in Sec. 2.1,  $\mathbf{x}_{mo}$ , the surface tangent plane and surface normal at  $\mathbf{x}_{mo}$  become known as well. As a result, the first order local description of the mirror surface is completely known if  $s$ ,  $\hat{\mathbf{x}}_{io}$ ,  $\mathbf{x}_{po}$  and  $\mathbf{O}_c$  are known. In such a case we say that the *first order description* of the mirror surface is given by the quadruplet  $(\mathbf{x}_{po}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s)$ . Thus, we want to address the following question: given a surface whose first order description is given by a quadruplet  $(\mathbf{x}_{po}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s)$ , what can we tell about the second order surface parameters  $a$ ,  $b$  and  $c$ ?

Let us consider  $n$  pattern lines  $\lambda_1, \lambda_2, \dots, \lambda_n$  intersecting in  $\mathbf{x}_{po}$ . Each pattern line produces a reflected curve on the mirror surface and a corresponding tangent

vector at  $\mathbf{x}_{mo}$ . We can impose the constraint expressed by Eq. 8 for each pattern line, obtaining the following system:

$$\begin{cases} \tan \varphi_1 = \frac{(J_u - 2a \cos \theta)B_{v_1} + 2cB_{u_1} \cos \theta}{(J_v - 2b \cos \theta)B_{u_1} + 2cB_{v_1} \cos \theta} \\ \vdots \\ \tan \varphi_n = \frac{(J_u - 2a \cos \theta)B_{v_n} + 2cB_{u_n} \cos \theta}{(J_v - 2b \cos \theta)B_{u_n} + 2cB_{v_n} \cos \theta} \end{cases} \quad (10)$$

where the subscripts 1,  $\dots$   $n$  indicate the quantities attached to  $\lambda_1, \dots \lambda_n$  respectively. After simple manipulations, we have:

$$\begin{cases} (J_u - 2a \cos \theta)B_{v_1} - (J_v - 2b \cos \theta)B_{u_1} \tan \varphi_1 + 2c \cos \theta (B_{u_1} - B_{v_1} \tan \varphi_1) = 0 \\ \vdots \\ (J_u - 2a \cos \theta)B_{v_n} - (J_v - 2b \cos \theta)B_{u_n} \tan \varphi_n + 2c \cos \theta (B_{u_n} - B_{v_n} \tan \varphi_n) = 0 \end{cases} \quad (11)$$

which is a linear system of  $n$  equations in 3 unknowns ( $a$ ,  $b$  and  $c$ ). The system of Eq. 11 can be expressed in the following matrix form:

$$\mathbf{H} \mathbf{g} = \begin{bmatrix} B_{v_1} & -B_{u_1} \tan \varphi_1 & B_{u_1} - B_{v_1} \tan \varphi_1 \\ B_{v_2} & -B_{u_2} \tan \varphi_2 & B_{u_2} - B_{v_2} \tan \varphi_2 \\ \vdots & \vdots & \vdots \\ B_{v_n} & -B_{u_n} \tan \varphi_n & B_{u_n} - B_{v_n} \tan \varphi_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = 0 \quad (12)$$

where  $\alpha = J_u - 2a \cos \theta$ ,  $\beta = J_v - 2b \cos \theta$ ,  $\gamma = 2c \cos \theta$ ,  $\mathbf{H}$  and  $\mathbf{g}$  are a  $n \times 3$  matrix and a vector respectively capturing the quantities at right side of the equality. Eq. 12 is an homogeneous linear system in the unknowns  $\alpha$ ,  $\beta$  and  $\gamma$ . We want to study the properties of such a system.

**Definition 1.** A surface, whose first order description is given by the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s)$ , is called singular at  $\mathbf{x}_{m_o}$  if its second order parameters  $a, b$  and  $c$  are:

$$\begin{cases} a = \frac{J_u}{2 \cos \theta} \\ b = \frac{J_v}{2 \cos \theta} \\ c = (\frac{J_u}{2 \cos \theta} - a)(\frac{J_v}{2 \cos \theta} - b) = 0 \end{cases} \quad (13)$$

As shown in details in [1], for a surface singular at  $\mathbf{x}_{m_o}$ , it turns out that the Jacobian attached to mapping  $t_o \in \mathbb{R} \rightarrow \mathbf{x}_{m_o} \in \mathbb{R}^3$  is singular and the resulting Eq. 8 is no longer valid.

**Proposition 1.** Let us assume to have a mirror surface whose first order description is given by the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s)$  and which is non singular at  $\mathbf{x}_{m_o}$ . Let us consider  $n \geq 2$  pattern lines passing through  $\mathbf{x}_{p_o}$  but not lying in the principal plane. Then the rank of matrix  $\mathbf{H}$  is 2.

*Proof.* That  $\mathbf{H}$  must have rank  $\leq 3$  is trivial. We want to prove, by contradiction, that the rank cannot be neither 3, nor 1, nor 0. For reason of space we omit the proof of the last 2 cases. Interested readers may find more details in a forthcoming technical report. Let us suppose that  $\mathbf{H}$  has rank 3. The homogeneous system of Eq. 12 has a unique solution, which must be  $\mathbf{g} = 0$ . Thus,  $J_u - 2a \cos \theta = 0$ ,  $J_v - 2b \cos \theta = 0$  and  $2c \cos \theta = 0$ . Since  $J_u$ ,  $J_v$  and  $\cos \theta$  are positive quantities, the surface must be singular at  $\mathbf{x}_{m_o}$ . As a conclusion  $\mathbf{H}$  cannot be a full rank matrix.  $\square$

Proposition 1 tells us that, no matter how many tangent vector measurement  $\hat{\mathbf{t}}_{io}$ 's are used, the second order surface parameters  $a$ ,  $b$  and  $c$  can be estimated only up to an unknown parameter. As final remark, we notice that both hypotheses of proposition 1 are necessary for observations (measured tangent vectors) to be meaningful and, therefore, for the reconstruction to be feasible. Thus, in all practical cases, both hypotheses are always satisfied and therefore the proposition verified.

Let us consider a mirror surface whose first order description is given by a quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s)$ . Since  $\text{rank}(\mathbf{H}) = 2$ , the space spanned by the rows of  $\mathbf{H}$  is a plane. The vector  $\mathbf{g}$  must be orthogonal to such a plane. Let  $\mathbf{h}_i$  and  $\mathbf{h}_j$  be any two row vectors of  $\mathbf{H}$ . If we define the vector  $\mathbf{v} = [v_1 \ v_2 \ v_3]^T$  as follows:

$$\mathbf{v} = \mathbf{h}_i \times \mathbf{h}_j = \begin{bmatrix} -B_{u_i} \tan \varphi_i (B_{u_j} - B_{v_j} \tan \varphi_j) + (B_{u_j} - B_{v_j} \tan \varphi_i) B_{u_i} \tan \varphi_j \\ (B_{u_j} - B_{v_j} \tan \varphi_i) B_{v_j} - (B_{u_i} - B_{v_j} \tan \varphi_j) B_{v_i} \\ -B_{v_i} B_{u_j} \tan \varphi_j + B_{u_i} B_{v_j} \tan \varphi_i \end{bmatrix} \quad (14)$$

we have:

$$k \mathbf{v} = \mathbf{g} = \begin{bmatrix} J_u - 2a \cos \theta \\ J_v - 2b \cos \theta \\ 2c \cos \theta \end{bmatrix} \quad (15)$$

where  $k$  is a scalar. Combining Eq. 14 with Eq. 15:

$$\begin{cases} a = \frac{J_u}{2 \cos \theta} - k \frac{v_1}{2 \cos \theta} \\ b = \frac{J_v}{2 \cos \theta} - k \frac{v_2}{2 \cos \theta} \\ c = k \frac{v_3}{2 \cos \theta} \end{cases} \quad (16)$$

As a result, any two tangent vector measurements suffice to constrain the second order description of the mirror surface around  $\mathbf{x}_{m_o}$  up to the unknown parameter  $k$ . Proposition 1 guarantees that we cannot do better than so, even using more than two pattern lines. Eqs. 16 give a quantitative relationship between the second order surface parameters  $a$ ,  $b$  and  $c$ , any two pattern line orientations (embedded in the  $B_u$ 's and  $B_v$ 's), the corresponding tangent vector measurements (embedded in the  $\varphi$ 's) and the unknown parameter  $k$ .



### 3.2 The Space of Paraboloids

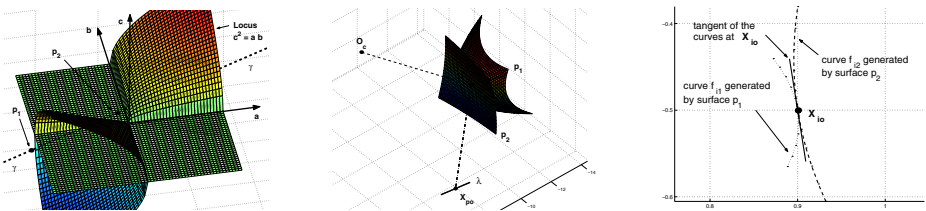
In this section we introduce a  $\mathbb{R}^3$ -space, called *space of paraboloids*, in which the geometry describing our results can be represented in a more clear fashion. In the space of paraboloids, the coordinates of a point  $[a \ b \ c]^T$  univocally describe a paraboloid given by  $w = au^2 + bv^2 + 2cuv$ . See Fig. 3 (left panel).

Let us consider a mirror surface  $M^*$  whose first order description is given by the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{i_o}, s^*)$ . Since the second order terms of the Taylor expansion around  $\mathbf{x}_{m_o}$  of the surface Monge form attached to  $M^*$  define a paraboloid with parameters  $a$ ,  $b$  and  $c$ , we say that the *second order description* of  $M^*$  is given by a point  $\mathbf{p}$  in the space of paraboloids  $\wp$ . Thus,  $\wp$  represents the space of all possible second order descriptions of a surface having vertex in  $\mathbf{x}_{m_o}$  and normal  $\mathbf{n}_{m_o}$  at  $\mathbf{x}_{m_o}$ . Let  $\mathbf{p}^*$  be the unknown paraboloid defining the second order description of  $M^*$ . If we take  $n$  pattern lines and the corresponding tangent vector measurements in the image plane, Proposition 1 tells us that we cannot fully estimate  $\mathbf{p}^*$ . However, by means of Eqs. 16, with any 2 pattern lines and the corresponding measurements we can estimate a *family* of paraboloids parametrized by  $k$ . In  $\wp$ , such a family is a line  $\gamma$  described by the following parametric form:

$$\mathbf{p}(k) = \mathbf{p}_o + k \mathbf{v}' = \begin{bmatrix} \frac{J_u}{2 \cos \theta} \\ \frac{J_v}{2 \cos \theta} \\ 0 \end{bmatrix} + k \frac{\mathbf{v}}{2 \cos \theta} \quad (17)$$

**Proposition 2.** *Consider a mirror surface  $M^*$  whose first order description is given by the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{i_o}, s^*)$  and whose second order description is given by  $\mathbf{p}^*$ . Assume that  $M^*$  is not singular at  $\mathbf{x}_{m_o}$ . Then any pair of pattern lines and corresponding measurements produce the same family of paraboloids, namely, the same line  $\gamma$ .*

*Proof.*  $\gamma$  is constrained to pass through  $\mathbf{p}_o$ , which only depends on the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{i_o}, s^*)$ . Additionally,  $\gamma$  is constrained to pass through  $\mathbf{p}^*$ . On



**Fig. 3.** **Left panel:** the space of paraboloids.  $\mathbf{p}_1$  and  $\mathbf{p}_2$  are two paraboloids belonging to the same family  $\gamma$ . The locus  $c^2 = ab$  (namely, the set of all parabolic paraboloids) separates the space into two regions. All of the points such that  $c^2 < ab$  correspond to elliptic paraboloids whereas all of the points such that  $c^2 > ab$  corresponds to hyperbolic paraboloids. Thus,  $\mathbf{p}_1$  is elliptic whereas  $\mathbf{p}_2$  is hyperbolic. **Middle panel:**  $\mathbf{p}_1$  and  $\mathbf{p}_2$  in the  $[XYZ]$  reference system.  $\lambda$  is a possible pattern line. **Right panel:** Images of the reflected pattern line  $\lambda$ .  $\mathbf{f}_{i1}$  and  $\mathbf{f}_{i2}$  are generated by  $\mathbf{p}_1$  and  $\mathbf{p}_2$  respectively. Notice that the tangents of the curves at  $\mathbf{x}_{i_o}$  are coincident (ambiguity of type I).

the other hand,  $\mathbf{p}_o$  is the second order description of a surface singular at  $\mathbf{x}_{mo}$ . Thus,  $\mathbf{p}_o \neq \mathbf{p}^*$ . As a result,  $\gamma$  must be invariant, no matter which pair of pattern lines and corresponding measurements are considered.  $\square$

Consider now a family of mirror surfaces whose first order description is given by the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s^*)$  and whose second order description is given by any paraboloid belonging to a line  $\gamma$ . Then, given an arbitrary pattern line, any surface belonging to the family must produce the same measurement, i.e. the same tangent vector  $\hat{\mathbf{t}}_{io}$ . This conclusion highlights a fundamental ambiguity as far as the second order description of a surface is concerned:

**Proposition 3. *Specular reflection ambiguity of type I.*** *Given a camera and a pattern line passing through a point  $\mathbf{x}_{p_o}$ , there exists a whole family of mirror surfaces producing a family of reflected image curves whose tangent vector at  $\mathbf{x}_{i_o}$  is invariant —  $\mathbf{x}_{i_o}$  being the image of the reflection of  $\mathbf{x}_{p_o}$ .*

In order to validate our theoretical results, we have implemented a program in MatLab to simulate specular reflections. Given a pattern line, a known surface (defined as a graph) and the observer, the routine computes the corresponding reflected curve imaged by the observer. In Fig. 3 an example of ambiguity of type I is provided.

### 3.3 Estimation of the Distance Parameter $s$

A crucial assumption made in the previous sections was that a quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s^*)$  giving the first order description of an unknown mirror surface  $M^*$  was available. However, while  $\mathbf{x}_{p_o}$ ,  $\mathbf{O}_c$ ,  $\hat{\mathbf{x}}_{io}$  are known since we assume that both camera and pattern are calibrated, the parameter  $s^*$  still needs to be estimated,  $s^*$  being the distance between  $\mathbf{O}_c$  and  $\mathbf{x}_{m_o}$ .

Let us take  $n$  pattern lines  $\lambda_1, \lambda_2, \dots, \lambda_n$  intersecting in  $\mathbf{x}_{p_o}$  and the corresponding tangent vector measurements and consider the matrix  $\mathbf{H}$  of Eq. 12. Since each entry of  $\mathbf{H}$  is parametrized by  $s$ ,  $\det(\mathbf{H}^T \mathbf{H})$  is a function of  $s$ . Let us call it  $\Psi(s)$ . When  $s = s^*$ , Proposition 1 is verified. Thus  $\Psi(s^*) = 0$ . On the other hand, when  $s \neq s^*$ , we cannot say anything about  $\det(\mathbf{H}^T \mathbf{H})$  but we would expect it to be different from zero since our measurements would not be consistent with tangent vectors produced by the geometry attached to  $s \neq s^*$ . In Fig. 4 an instance of  $\Psi(s)$  is shown. Such a plot was obtained by means of our specular reflection simulator. A triplet of pattern lines with the 3 corresponding measurements was considered. Thus,  $\Psi$  is just the determinant of the corresponding  $3 \times 3$  matrix  $\mathbf{H}$ . As we can see from the plot,  $\Psi(s)$  vanishes in  $s^*$ . However,  $\Psi(s)$  vanishes in other point,  $s'$ , as well. Such value of  $s$  corresponds to a wrong (or *ghost*) solution. Namely, the quadruplet  $(\mathbf{x}_{p_o}, \mathbf{O}_c, \hat{\mathbf{x}}_{io}, s')$  gives a first order description of a mirror surface  $M' \neq M^*$  and, as far as the 3 tangent vector measurements are concerned, there is no way to discriminate such surface from the correct one. In other words, the 3 tangent vectors produced in the image plane by  $M^*$  and  $M'$  are exactly the same. This is what we call the **specular reflection ambiguity of type II**. In our experience, only a few (usually 1 or 2, even none) ambiguities arise for each  $s^*$  (see Fig. 8).

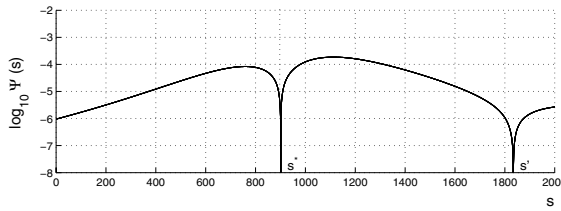


Fig. 4. An instance of  $\det(\mathbf{H}) = \Psi(s)$ .

We can think about three possible ways to get rid of such ambiguities. i) Our simulations show that the specular reflection ambiguity of type II is actually related to a particular triplet of pattern lines. Namely, by considering  $m > 3$  no-coplanar pattern lines, the actual  $s^*$  can be found without ambiguities, since  $\det(\mathbf{H}^T \mathbf{H})$  — the matrix  $\mathbf{H}$  being  $m \times 3$  — vanishes only in  $s^*$ . Further work is needed to theoretically validate such conclusions. ii) If a rough estimate of the distance is available, then usually only one solution is consistent with this estimate. iii) A second order approach, by means of curvature estimates of the image curves at  $\mathbf{x}_{i_o}$ , can be used. The basic equations have been derived in [1] although further theoretical and experimental investigation is needed.

### 3.4 Special Cases: Sphere and Cylinder

Let us assume that we have some a priori information about the surface. Such information may be translated into a relationship  $R(a, b, c) = 0$  between the second order surface parameters.  $R(a, b, c) = 0$  can be seen as a surface (or a volume or a curve, depending on the type of relationship) in the space of paraboloids. Thus, by intersecting the family (line)  $\gamma$  and such  $R(a, b, c) = 0$ , more information about the mirror surface become available. In some particular cases, the full second order surface description can be achieved. Let us examine two interesting cases.

**Sphere.** If the mirror surface is a sphere with unknown radius  $r$ , the relationship  $R(a, b, c) = 0$  becomes:  $a = b$  and  $c = 0$ , which is simply a line  $\rho$  lying in the plane defined by  $c = 0$ . At  $s = s^*$ , the intersection between  $\rho$  and  $\gamma$  allows to compute  $r$ . Namely, imposing that  $a = b$  and having in mind Eq. 17, we obtain:

$$k = \frac{J_u - J_v}{v_1 - v_2}; \quad r = \frac{2(v_1 - v_2) \cos \theta}{J_v v_1 - J_u v_2} \quad (18)$$

Eq. 18 completely solves the ambiguity of type I.

Additionally, by imposing that  $c = 0$ , we have  $k v_3 / \cos \theta = 0$ . Since  $\cos \theta \neq 0$  and  $k \neq 0$  (otherwise the surface would be singular),  $v_3$  must be zero, namely:

$$\phi(s^*) = -B_{v_i} B_{u_j} \tan \varphi_j + B_{u_i} B_{v_j} \tan \varphi_i = 0 \quad (19)$$

which is exactly the result achieved in [1]: the parameter  $s^*$  was found by imposing  $\phi(s)$  to vanish. The condition  $\det(\mathbf{H}^T \mathbf{H}) = 0$  given in this paper is a generalization. Notice that  $\phi(s)$  is the determinant of the  $2 \times 2$  matrix  $\mathbf{H}_s$  obtained by taking the  $i^{th}$  and  $j^{th}$  rows and the first 2 columns of  $\mathbf{H}$ . Since Eq. 19

holds  $\forall i, j$  with  $i \neq j$ ,  $\det(\mathbf{H}_s) = 0 \Rightarrow \det(\mathbf{H}^T \mathbf{H}) = 0$ . Such a result can be used in order to easily remove ambiguities of type II. In fact, if there is an  $\bar{s}$  such that  $\psi(\bar{s}) = 0$  but  $\phi(\bar{s}) \neq 0$ ,  $\bar{s}$  must be a ghost solution.

**Cylinder.** Let us focus our attention on the side surface  $M_c$  of the cylinder. The second order term of the Taylor expansion around any point  $\in M_c$  of the surface Monge form attached to  $M_c$  is described by a parabolic paraboloid (see, for instance, [6]). Thus,  $R(a, b, c) = 0$  is  $c^2 = ab$ , which is the locus depicted in figure 3. At  $s = s^*$ , the intersection between such locus and  $\gamma$  gives:

$$k = \frac{(J_v v_1 + J_u v_2) \pm \sqrt{(J_v v_1 + J_u v_2)^2 - 4(v_1 v_2 - v_3^2) J_u J_v}}{2(v_1 v_2 - v_3^2)} \quad (20)$$

The corresponding  $a$ ,  $b$  and  $c$  can be computed by means of Eq. 16 or Eq. 17.

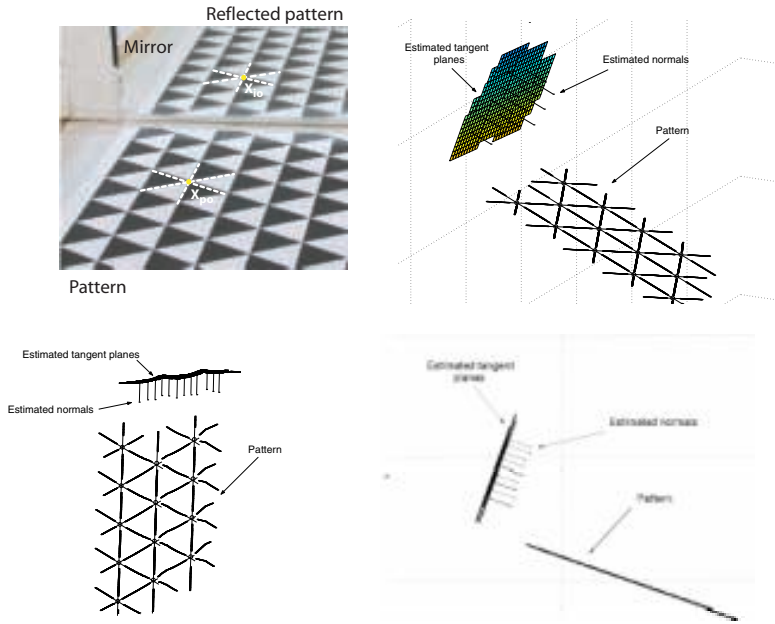
### 3.5 The Reconstruction Procedure for a Generic Smooth Surface

According to the results discussed in previous sections, we outline the following reconstruction procedure. A calibrated camera facing an unknown mirror surface and a calibrated pattern (e.g. 3 lines intersecting in  $\mathbf{x}_{p_o}$ ) are considered. The image point  $\mathbf{x}_{i_o}$  and the tangent vectors of the image reflected curves at  $\mathbf{x}_{i_o}$  are measured. Thus, the entries of  $\mathbf{H}$  are function of  $s$  only (Eq. 15). Since  $\det(\mathbf{H})(s)$  vanishes in  $s^*$ , where  $s^*$  is the correct distance between  $\mathbf{O}_c$  and the reflection point  $\mathbf{x}_{m_o}$ , we solve  $\det(\mathbf{H})(s) = 0$  numerically. If  $s^*$  is the unique solution,  $\mathbf{x}_{m_o}$  and the normal of the surface at  $\mathbf{x}_{m_o}$  are calculated in  $s^*$  by means of Eq. 1 and Eq. 2. Thus, the first order description of the surface is completely known. As for the second order description, the parameters  $a$ ,  $b$  and  $c$ , up to the unknown parameter  $k$ , are calculated by means of Eq. 16. If  $\det(\mathbf{H})(s) = 0$  yields multiple solutions, we may want to consider the discussion in 3.3.

## 4 Experimental Results

Our setup is sketched in Fig. 2. A camera faces the mirror surface and the pattern. In our experiments, a Canon G1 digital camera, with image resolution of  $2048 \times 1536$  pixels, was used. The surface was typically placed at a distance of  $30 \div 50$  cm from the camera. The pattern — a set of planar triplets of intersecting lines — is formed by a tessellation of black and white equilateral triangles. For instance, 3 white dashed edges as in Fig. 5 form a triplet of lines. The camera and the ground plane (i.e. the plane where the pattern lies) were calibrated by means of standard calibration techniques.

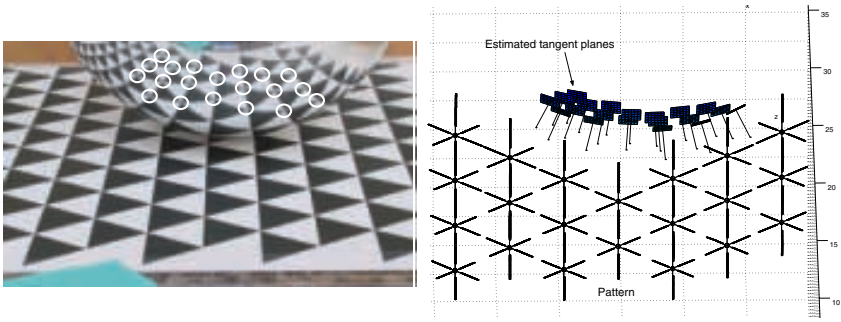
The reconstruction routine proceeds as follows. We manually selected a pair of triplet of lines (i.e. a triplet of pattern lines and a corresponding triplet of reflected pattern lines) from the image plane. See, for instance, the 2 dashed line triplets in Fig. 5. The selected triangle edges were estimated with sub-pixel accuracy and a B-spline was used to fit the edge points. The points  $\mathbf{x}_{p_o}$  and  $\mathbf{x}_{i_o}$  were computed by intersecting the corresponding splines. The tangents at  $\mathbf{x}_{p_o}$  and  $\mathbf{x}_{i_o}$  are obtained by numerically differentiating the splines. According to Sec. 3,  $\mathbf{x}_{p_o}$ ,  $\mathbf{x}_{i_o}$  and the corresponding tangents are used to estimate the distance  $s$  from the reflection point  $\mathbf{x}_{m_o}$  on mirror surface to the center of the camera.



**Fig. 5. Reconstruction of a planar mirror.** Upper left panel: a planar mirror placed orthogonal with respect to the ground plane. A triplet of pattern lines and the corresponding reflected triplet are highlighted with dashed lines. We calculated the ground truth on the position and orientation of the mirror by attaching a calibrated pattern to its surface. We then reconstructed 15 surface points and normals with our method. The resulting mean position error (computed as average distance from the reconstructed points to the ground truth plane) is  $-0.048$  cm with a standard deviation of  $0.115$  cm. The mean normal error (computed as the angle between ground truth plane normal and estimated normal) is  $1.5 \times 10^{-4}$  rad with a standard deviation of  $6.5 \times 10^{-4}$  rad. The reconstructed region is located at about 50 cm to the camera. Upper right: 3/4 view of the reconstruction. For each reconstructed point, the normal and the tangent plane are also plotted. Lower left: top view. Lower right: side view.

The normal of the surface and the tangent plane at  $\mathbf{x}_{m_o}$  are estimated by means of Eq. 2. We validated the method with four mirror surfaces: a plane (Fig. 5), a sphere (Fig. 6), a cylinder (Fig. 7) and a sauce pan's lid (Fig. 9). Where we had a ground truth to compare with, we qualitatively tested the reconstruction results. As for the plane, depth and normal reconstruction errors are about 0.2% and less than 0.1% respectively. As for the sphere, the curvature reconstruction error is about 2%.

As explained in more details in [1], the reconstruction is not feasible when a pattern line is either orthogonal or belonging to the principal plane. In such cases, the constraint expressed by the tangent vector does no longer carry meaningful information. See Fig. 7 for an example. As a further remark, according to Sec. 3.3, we remind that each reconstructed point might be associated to one or more



**Fig. 6. Reconstruction of the sphere.** **Left panel:** a spherical mirror with radius  $r = 6.5$  cm, placed on the ground plane. We reconstructed the surface at the points highlighted with white circles. For each surface point we estimated the radius by means of Eq. 18. The mean reconstructed radius is 6.83 cm and the standard deviation is 0.7 cm. The reconstructed region is located at a distance about 30 cm to the camera. **Right panel:** top view of the reconstruction.

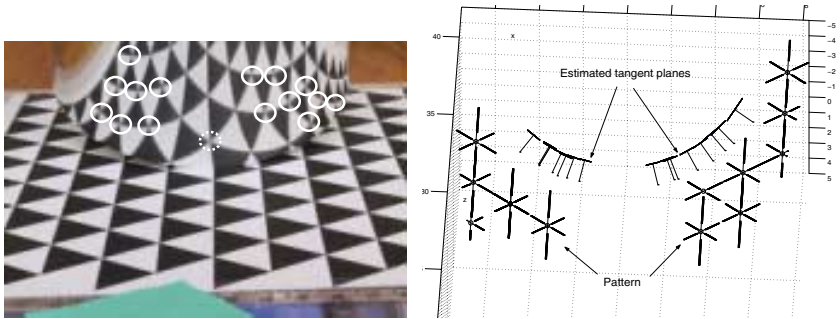
ghost solutions. Such solutions were easily rejected since always located farther than the correct ones from the center of camera. See Fig. 7 for examples.

From a practical point of view, detection and labeling of reflected and pattern lines are not to be considered a negligible issue. We notice here that in presence of unpolished mirror surfaces, the lines and the corresponding tangents might be estimated with a certain amount of noise. Further work is needed to study how much such a noise may affect the reconstruction results. It might also be interesting to relate the tangent estimate accuracy to the surface local curvature. Additionally, notice that we have always considered smooth surfaces. Since the analysis is local, the smoothness is not a necessary hypothesis for the reconstruction to be feasible as long as the reflecting points do not lie in any surface discontinuities. However, in practice, reflected pattern lines may be very hard to detect in a neighborhood of the discontinuities. This is also the case when concave surfaces are considered: although (as discussed in Sec. 2) the method deals with generic surfaces, multiple reflections and inter-reflections can make the line detection a quite difficult problem.

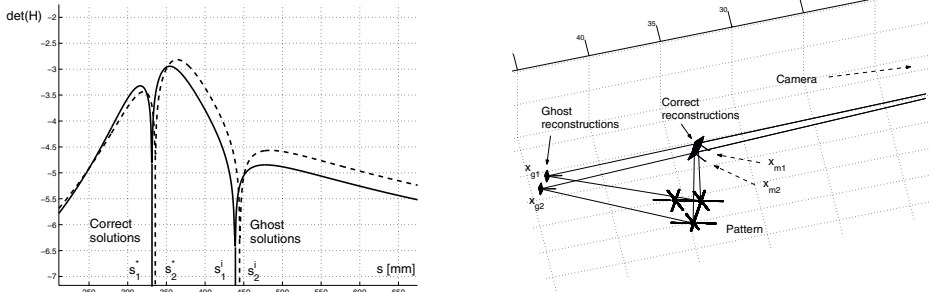
## 5 Conclusion and Future Work

We have presented an explicit solution for reconstructing the first order local geometry (position and normals) of generic smooth mirror surfaces. Such analysis relates the position of a point in the image, and the orientation of 3 lines through that point, to the local structure of the mirror around the point of reflection. A few discrete ambiguities in the reconstruction may arise. They may be removed by considering either more than 3 non-coplanar lines or a rough a priori estimate of the surface point location. Additionally, we have explicitly expressed the second order local surface parameters as a function of an unknown scalar factor. Finally, we have validated our analysis with experimental results.

Future work is needed to study how sensitive the estimated parameters are with respect to noise added to the tangents, to investigate whether the second order ambiguity can be solved by measuring the reflected line curvatures or



**Fig. 7. Reconstruction of the cylinder.** **Left panel:** a cylinder placed with the main axis almost orthogonal to the ground plane. We reconstructed the surface at the points highlighted with white circles. The dashed circle indicate an instance of point for which the reconstruction is not feasible or highly inaccurate (see Sec. 4). **Right:** top view of the reconstruction.

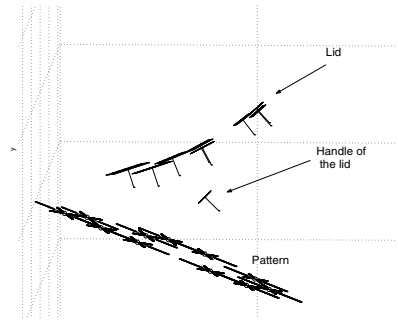
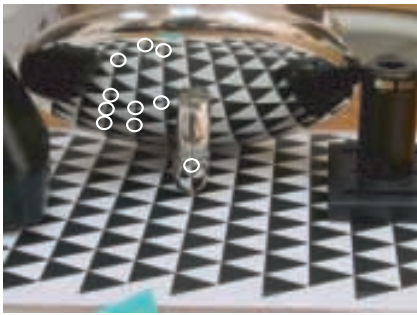


**Fig. 8. Example of ambiguities of type II.** **Left panel:** Two reconstructed points from the cylinder are considered. The plots of the corresponding  $\det(\mathbf{H})$  functions are in dashed and solid lines. The corresponding correct solutions are  $s_1^*$  and  $s_2^*$ . The ghost solutions are  $s_1'$  and  $s_2'$  respectively. **Right:** The correct reconstructed points (attached to  $s_1^*$  and  $s_2^*$ ) are  $\mathbf{x}_{m1}$  and  $\mathbf{x}_{m2}$ . The ghost solutions are  $\mathbf{x}_{g1}$  and  $\mathbf{x}_{g2}$ . Such solutions can be easily rejected since they appear at about 10 cm farther than the correct ones.

whether such ambiguities would disappear if the pattern is differentially moved in a known direction, and, finally, to extend the analysis to the stereoscopic vision case.

We view our results as a promising start in the quest of computing the global shape of specular surfaces under fairly general conditions. The more interesting case of an uncalibrated world appears much more challenging and will require most certainly the integration of additional cues and some form of prior knowledge on the likely statistics of the scene geometry.

**Acknowledgments.** This work is supported by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (EEC-9402726). We wish to thank Min Chen, Marzia Polito and Fei Fei Li for helpful feedback and comments.



**Fig. 9. Reconstruction of the sauce pan's lid.** **Left panel:** a sauce pan's lid placed with the handle touching the ground plane. We reconstructed the surface at the points highlighted with white circles. Notice that one point belongs to the handle of the lid. **Right panel:** side view of the reconstruction. Notice how the reconstructed point on the handle sticks out from the body of the lid.

## References

1. S. Savarese and P. Perona: Local Analysis for 3D Reconstruction of Specular Surfaces. *IEEE Conf. on Computer Vision and Pattern Recognition*, II 738–745 (2001)
2. T. Binford: Inferring surfaces from images. *Artificial Intelligence*, **17** (1981) 205–244.
3. A. Blake: Specular stereo. *IJCAI* (1985) 973–976.
4. A. Blake and G. Brelstaff: Geometry from specularities. *ICCV Proc. of Int Conf. of Computer Vision* (1988) 394–403.
5. M. Chen and J. Arvo: Theory and Application of Specular Path Perturbation. *ACM Transactions on Graphics*. **19** (2000) 246–278.
6. R. Cipolla and P. Giblin: Visual motion of curves and surfaces. *Cambridge University Press* 2000.
7. M. Halsead, A. Barsky, S. Klein, and R. Mandell: Reconstructing curved surfaces from reflection patterns using spline surface fitting normals. *SIGGRAPH* (1996).
8. G. Healey and T. Binford: Local shape from specularity. *Computer Vision, Graphics, and Image Processing* **42** (1988) 62–86.
9. K. Ikeuchi: Determining surface orientation of specular surfaces by using the photometric stereo method. *IEEE PAMI* **3** (1981) 661–669.
10. J. Koenderink and A. van Doorn: Photometric invariants related to solid shape. *Optica Acta* **27** (1980) 981–996.
11. M. Oren and S. K. Nayar: A theory of specular surface geometry. *Trans. Int. Journal of Computer Vision* (1997) 105–124.
12. J. Zheng and A. Murata: Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *IEEE PAMI* **8** (2000).
13. A. Zisserman, P. Giblin, and A. Blake: The information available to a moving observer from specularities. *Image and Video Computing* **7** (1989) 38–42.
14. D. Perard: Automated visual inspection of specular surfaces with structured-lighting reflection techniques. *PhD Thesis – VDI Verlag* **Nr. 869** (2001)



# Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration

Nikos Paragios<sup>1</sup>, Mikael Rousson<sup>2</sup>, and Visvanathan Ramesh<sup>1</sup>

<sup>1</sup> Siemens Corporate Research,  
Imaging and Visualization Department,  
755 College Road East, Princeton, NJ 08540, USA  
{nikos,rameshv}@scr.siemens.com

<sup>2</sup> I.N.R.I.A.  
B.P. 93, 2004 Route des Lucioles,  
06902 Sophia Antipolis Cedex, France  
Mikael.Rousson@sophia.inria.fr

**Abstract.** This paper deals with the matching of geometric shapes. Our primary contribution is the use of a simple, robust, rich and efficient way to represent shapes, the level set representations according to signed distance transforms. Based on these representations we propose a variational framework for global as well as local shape registration that can be extended to deal with structures of higher dimension. The optimization criterion is invariant to rotation, translation and scale and combines efficiently a global motion model with local pixel-wise deformations. Promising results are obtained on examples showing small and large global deformations as well as arbitrary topological changes.

## 1 Introduction

The problem of registering geometric shapes [23] is a complex issue in computer vision, computer graphics and recently medical imaging. It has been studied in various forms during the last decade due to its wide application not limited to registration, recognition, retrieval, tracking, etc.

A general registration formulation can be stated as follows: given two shapes, an input  $\mathcal{D}$  and a target  $\mathcal{S}$ , and a dissimilarity measure, find the best transformation that associates to any point of  $\mathcal{D}$  a corresponding point at  $\mathcal{S}$  and minimizes the dissimilarity measure between the transformed shape  $\hat{\mathcal{D}}$  and the target  $\mathcal{S}$ . This dissimilarity can be defined either along the contour (*shape-based*) or in the entire region (*area-based*) determined by the contour.

At this point, we have to distinguish the methods that aim to provide shape recognition. In that case correspondences between the source and the target shape are considered known or can be easily recovered and the objective is to find from a given set of examples the shape that provides the lower dissimilarity measurement with the target [1, 19, 22, 26]. Towards this direction, methods that do not require correspondence and are based on the matching of some global shape characteristics have been also investigated. It is important to note that

our paper does not focus on the matching and the recognition of shapes but addresses registration with unknown correspondence.

Shape registration has been approached into a number of ways, that can be mainly categorized [14] according to three factors: (i) Nature of Transformation, (ii) Domain of Transformation, (iii) Optimization Procedure. An effort to briefly cover the existing literature on these areas will be made in the next paragraphs.

A critical component of the registration procedure is the underlying motion model (nature of transformation) that is used to map the current shape to the target. The selection of the motion model can affect drastically the performance of the registration procedure. *Rigid* transformations refer to translation and rotation and are a compromise between low complexity and fairly acceptable matching between the different structures. A step forward, the *affine* transformation, is a more complicated model that is invariant to a large variety of motions. The use of *projective* geometry has been also considered to match shapes. Finally, *curved* or *elastic* registration methods can better account for local deformations and therefore have been also considered.

As far as the domain of transformation is concerned, the distinction is easier. *Global* transformations are valid for the entire shape [1] and for any point of  $\mathcal{D}$  we can obtain the corresponding point at  $\mathcal{S}$  by applying the same transformation, while *local* transformations are applied at the pixel level [5].

Finally, the optimization procedure refers to the selection of a mathematical framework by means of finding an optimum of some functional defined on the parameter space. These functionals attempt to quantify the similarity between the two shapes and can be based either in variational [4, 9], or stochastic principles [24]. As far the optimization procedures are concerned, gradient-descent, geometric hashing, iterative closest point, etc. can be found in the literature. Finally, methods for shape registration can be categorized according to the feature space on which the matching is performed (Euclidean space, affine space, curvature space, medial axes/skeletons [19, 26], graph and shocks [21], etc.).

In this paper we propose a variational framework for shape alignment that can be extended to any arbitrary dimension. We adopt an implicit representation inspired by the level set methods [16] and refers to signed distance transforms. Furthermore, we introduce an optimization criterion that can account for global (*rigid*, *affine*) and local pixel-wise deformations. This criterion is defined in the space of signed distance transforms, and is minimized using a gradient descent method. Global and local registration parameters are recovered using different update factors that allow us to recover first the global transformation and then the local deformations. Encouraging experimental results using 2D shapes were obtained.

Distance transforms [18] have been considered for image registrations in the past [6, 8, 12, 13]. In [6] a local-registration is recovered by seeking pixel-wise the lowest cost geodesic path between the source and the target. This method is very local, does not account for any neighborhood coherency and can fail to provide a meaningful registration result unless properly initialized.

In [8,12] the Euclidean distance was considered to be the feature space. Then 2D/3D objects are to be registered using linear transformations. In [12] registration is done towards a prior shape model that refers to a collection of points obtained through uniform sampling. In [8] distance maps from edges are considered. In both cases registration is obtained through the minimization of a metric defined on the space of distance transform. Both methods focus on global transformations and cannot deal with local deformations. Furthermore, they cannot deal with important scale variations and do make use of a feature space with less information compared with the one consider in this paper.

The reminder of this paper is organized as follows. In Section 2 we briefly introduce the selected shape representation. The registration method of is presented in Section 3 while summary and discussion appear in Section 4.

## 2 Shape Representation

A crucial component in the procedure of registering arbitrary shapes is the underlying shape representation [11] since it can significantly affect the performance of the selected registration algorithm.

The use of point-based snake models [10], deformable models/templates [2], active shapes [7], Fourier descriptors, medial axis, level set representations [3, 15] are some alternatives. Although, these representations are powerful enough to capture a certain number of local deformations, they require a large number of parameters to deal with important shape deformations, and they cannot deal with changes of topology. Also, their extension to describe structures of higher dimension than curves and surfaces is in most of the cases not trivial. Based, on these considerations and given the assumption that we would like to obtain a global to local registration, we will consider the use of the level set representations that can deal with local deformations and is invariant to translation and rotation.

Level Set Representation [16] is a powerful mathematical tool that can to deal with applications that share a common concern, evolving interfaces. Within these representations, contours/shapes are modeled as the zero-level set ( $\phi = 0$ ) of a function defined in a higher dimension. The most common selection for the embedding function is signed (Euclidean) distance transform.

In order to facilitate the introduction of the method, we consider the 2D case and let  $\Phi : \Omega \rightarrow \mathcal{R}^+$  be a Lipchitz function that refers to a level set representation for a given shape  $\mathcal{S}$ . This shape defines a region  $\mathcal{R}$  in the image plane  $\Omega$ . Given these definitions the following shape representation is considered:

$$\Phi_{\mathcal{S}}(x, y) = \begin{cases} 0, & (x, y) \in \mathcal{S} \\ +D((x, y), \mathcal{S}) > 0, & (x, y) \in \mathcal{R}_{\mathcal{S}} \\ -D((x, y), \mathcal{S}) < 0, & (x, y) \in [\Omega - \mathcal{R}_{\mathcal{S}}] \end{cases}$$

where  $D((x, y), \mathcal{S})$  refers to the min Euclidean distance between the grid location  $(x, y)$  and the shape  $\mathcal{S}$ . The fast marching algorithm can be used for the construction of these representations [20].

It can be easily shown that this representation  $[\Phi_S]$  is invariant to translation and rotation. Let us consider a shape  $\mathcal{D}$  that is obtained after rotating  $\mathcal{S}$  by an  $\theta$  angle and translating by a vector  $(T_x, T_y)$ . Then, if we consider a pixel  $(x, y)$  with distance  $d$  from the interface  $\mathcal{D}$ , the following relation holds:

$$d = D((x, y), \mathcal{D}) = \min_{(u, v) \in \mathcal{D}} \left\{ \sqrt{(u - x)^2 + (v - y)^2} \right\}$$

The use of the inverse transformation between  $\mathcal{D}$  and  $\mathcal{S}$  for  $(x, y)$  leads to the following equation:

$$\begin{aligned} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} &= \begin{pmatrix} x \cos(-\theta) + y \sin(-\theta) - T_x \\ -x \sin(-\theta) + y \cos(-\theta) - T_y \end{pmatrix} \\ \mathcal{S} &= \left\{ \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} u \cos(-\theta) + v \sin(-\theta) - T_x \\ -u \sin(-\theta) + v \cos(-\theta) - T_y \end{pmatrix} : (u, v) \in \mathcal{D} \right\} \end{aligned}$$

Thus, the distance between  $\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$  and the interface  $\mathcal{S}$  is given by

$$\begin{aligned} D((\hat{x}, \hat{y}), \mathcal{S}) &= \min_{(\hat{u}, \hat{v}) \in \mathcal{S}} \left\{ \sqrt{(\hat{u} - \hat{x})^2 + (\hat{v} - \hat{y})^2} \right\} \\ &= \min_{(u, v) \in \mathcal{D}} \left\{ \sqrt{\begin{aligned} &((u - x) \cos(-\theta) - (v - y) \sin(-\theta))^2 \\ &+ ((u - x) \sin(-\theta) + (v - y) \cos(-\theta))^2 \end{aligned}} \right\} \\ &= \min_{(u, v) \in \mathcal{D}} \left\{ \sqrt{(u - x)^2 + (v - y)^2} \right\} = D((x, y), \mathcal{D}) \end{aligned}$$

We have shown that the selected representation is invariant to rotation and translation. On the other hand when the transformation has as a scale component  $s$  then level set shape representations are not invariant and the following relation holds between the representation of  $\mathcal{S}$  and  $\mathcal{D}$ :

$$D((x, y), \mathcal{D}) = s D((\hat{x}, \hat{y}), \mathcal{S})$$

### 3 Shape Alignment and Registration

In the registration problem the objective is to find a point-wise transformation between the current shape  $\mathcal{D}$  and the target shape  $\mathcal{S}$  that minimizes a given dissimilarity measure between the two shapes.

The use of level set representations changes the dynamics of the problem since these representations refer to a higher dimensional space than the original one (2D shapes). Hence, we will seek for a transformation  $A$  that creates pixel-wise intensity correspondences (level set values) between the current shape representation  $s \Phi_{\mathcal{D}}$  and the target shape  $\Phi_{\mathcal{S}}$ . Due to the fact that the selected representations  $[\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}]$  are invariant to translation and rotation we can write:

$$\begin{aligned} &\left\{ \begin{aligned} &A(x, y) = \begin{pmatrix} A_x \\ A_y \end{pmatrix} = s \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \\ &\underline{\forall (\mathbf{x}, \mathbf{y}) \in \Omega : s \Phi_{\mathcal{D}}(\mathbf{x}, \mathbf{y}) = \Phi_{\mathcal{S}}(A(\mathbf{x}, \mathbf{y}))} \end{aligned} \right. \quad (s, \theta, T) \end{aligned}$$

Thus, registration can be viewed now as a global optimization task that involves all pixels in the image plane. Based on this assumption, several optimization criteria have been proposed to recover the unknown transformation parameters; sum of squared differences, optimization of the correlation ratio, mutual information, etc. In order to introduce and demonstrate the performance of our method, at the very beginning we will consider the simplest possible criterion, the sum of square differences.

### 3.1 Sum of Squared Differences

Initially, we assume a global rigid deformation between  $\mathcal{S}$  and  $\mathcal{D}$  that involves three parameters  $[A = (s, \theta, T)]$ ; a rotation angle  $\theta$ , and translation vector  $T = (T_x, T_y)$  and a scale factor  $s$ . Then, the optimization criterion is given by:

$$E(s, \theta, T) = \iint_{\Omega} (s\Phi_{\mathcal{D}}(x, y) - \Phi_{\mathcal{S}}(A^T(x, y)))^2 dx dy$$

However, the level set representations for the considered shapes  $\mathcal{S}$  and  $\mathcal{D}$  are not equally defined in the image plane (they depend on the positions of the initial shapes). As a consequence, we can constrain the information space and decrease the complexity of the problem by considering the area defined by two equal distance contours (inwards, outwards) from the input shapes.

$$E(s, \theta, T) = \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}(x, y), \Phi_{\mathcal{S}}(A^T(x, y))) (s\Phi_{\mathcal{D}}(x, y) - \Phi_{\mathcal{S}}(A^T(x, y)))^2 dx dy$$

where  $N_{\delta}$  is a binary function given by

$$N_{\delta}(\phi_1, \phi_2) = \begin{cases} 0, & \min(|\phi_1|, |\phi_2|) > \delta \\ 1, & \min(|\phi_1|, |\phi_2|) \leq \delta \end{cases}$$

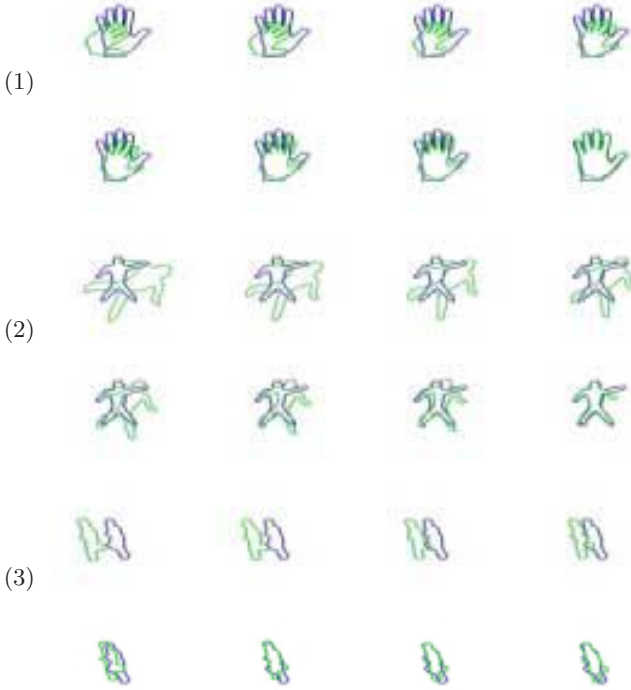
that has a simple interpretation. All pixels (isophotes), within a range of distance  $\delta$  from the actual shape are considered in the optimization process.

As a consequence, the registration is done in an augmented shape-driven (level set) space. This space is robust to very local deformations and missing data since the selected representation is obtained through a global procedure (Euclidean distance). Moreover, the proposed framework is invariant to rigid transformations and refers to multiple shape matching (isophotes) between shapes that are clones of the original ones to be registered.

Using the proposed formulation we were able to convert a geometry driven point-correspondence problem into an image-registration application where space as well feature-based (intensity) correspondences are considered.

The optimization of this criterion can be done using a gradient descent method

$$\begin{cases} \frac{d}{dt}\theta = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (\nabla \Phi_{\mathcal{S}} \cdot \nabla_{\theta} A^T) [(s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^T))] \\ \frac{d}{dt}s = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (\Phi_{\mathcal{D}} + \nabla \Phi_{\mathcal{S}} \cdot \nabla_s A^T) [(s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^T))] \\ \frac{d}{dt}T = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) \left( \nabla \Phi_{\mathcal{S}} \cdot \begin{pmatrix} \nabla_{T_x} A^T \\ \nabla_{T_y} A^T \end{pmatrix} \right) [(s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^T))] \end{cases}$$



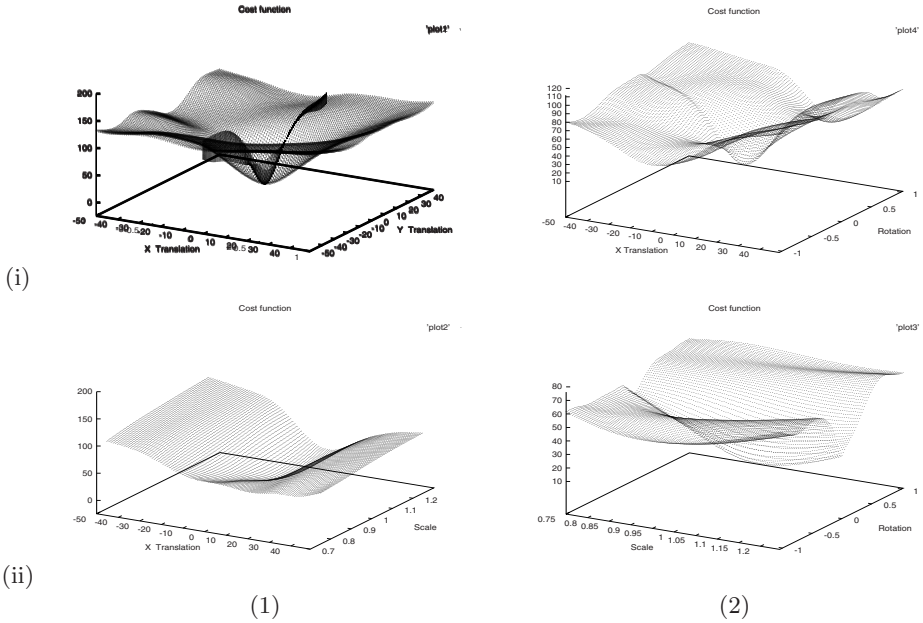
**Fig. 1.** (1) *Rigid Hand Registration (Synthetic Data)*:  $\{s = 1.27, \theta = 71.38^\circ, T_x = -19.65, T_y = 21.32\}$ . (2) *Rigid Body Registration (Synthetic Data)*:  $\{s = 0.63, \theta = 59.94^\circ, T_x = -13.97, T_y = -14.09\}$  (3) *Rigid Body Registration (Real Data)*.

The performance of the proposed module is shown in [FIG. (1)]. The selected representation is powerful and can provide encouraging results using a quite simple optimization criterion like the sum of squared differences. However, a validation of the method is required.

The characteristics of the cost function is a good indicator regarding the stability of the problem. Non-convex optimization criteria (like the one consider in this paper) suffer from the initial conditions. In our approach, we have considered a very strong feature space, the signed distance transforms and therefore one would expect that the performance of the method will be satisfactory.

In order to perform a study on this performance, we can constrain the unknown parameter space in two dimensions. We have considered the examples shown in [FIG. (1.1)]. Then, from the 4-dimensional parameter space we have studied the following cases:

- Scale and rotation are known, translation is to be recovered [FIG. (2.(i.1))],
- Translation in  $x$  and scale are known, translation in  $y$  and rotation are to be recovered [FIG. (2.(i.2))],
- Translation in  $x$  and rotation are known, translation in  $y$  and scale are to be recovered [FIG. (2.(ii.1))],



**Fig. 2.** Empirical evaluation of the cost function: (i.1) Unknown translation  $[x, y]$ , (i.2) Unknown translation  $[x]$  and rotation, (ii.1) Unknown translation  $[x]$  and scale, (ii.2) Unknown scale and rotation.

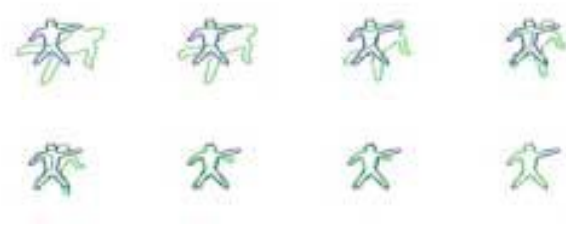
- Translation in  $x$  and translation in  $y$  are known, rotation and scale are to be recovered [FIG. (2.(ii.2))].

Then, we have quantized the search space using uniform sampling (100 samples) for all unknown parameters in each case. Translation transformations in  $(x, y)$  were in the range of  $[-50, 50] \times [-50, 50]$ , scale in  $[0.75, 1.25]$  and rotation in  $[-\frac{\pi}{3}, \frac{\pi}{3}]$ . Then, one can estimate the cost function in the space of two unknown parameters, by considering all possible combinations derived from the sampling strategy (the other two parameters are fixed) [FIG. (2)]. The resulting functional as shown in [FIG. (2)] has some nice properties; it is smooth and exhibits a single global minimum.

One can claim that the cost function [FIG. (2)] has a convex form for all cases with two unknown variables. This convexity cannot be guaranteed when the registration problem is considered in its full dimensionality, the four variables of rigid transformations. However, the form of this function in a reduced variable space is a good indicator for a well-behaved optimization criterion with smooth properties.

### 3.2 Global-to-Local Registration

The framework described in the previous section is limited to global rigid transformation and cannot account for non-rigid objects (shapes) and local deforma-



**Fig. 3.** *Global-to-Local Rigid Body Registration.*  $\{s = 0.63, \theta = 59.94^\circ, T_x = -13.97, T_y = -14.09\}$

tions. In order to overcome this limitation we will incorporate to the method the notion of local deformations by assuming that the observed shape is a rigid transformation  $A$  of the target combined with some local deformations  $(u, v)$ . Under this assumption, we can write:

$$\begin{cases} (s, \theta, T) \\ (u(x, y), v(x, y)) : (x, y) \in \Omega \\ \forall (\mathbf{x}, \mathbf{y}) \in \Omega : s\Phi_{\mathcal{D}}(\mathbf{x}, \mathbf{y}) = \Phi_{\mathcal{S}}(\mathbf{A}^\tau + (\mathbf{u}, \mathbf{v})) \end{cases}$$

Now, we can decompose the previous hypothesis into two terms and define the following registration criterion:

$$\begin{aligned} E(s, \theta, T, (u, v)) &= \alpha \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^\tau))^2 \\ &+ (1 - \alpha) \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^\tau + (u, v)))^2 \end{aligned}$$

that has a simple interpretation: registration errors according to the rigid transformation are corrected using the local deformation field. The performance of this additional term is shown in [fig. (3)]. For demonstration purposes, we consider the same input as the one that was used to validate the performance of the criterion that accounts only for a global rigid transformation.

A natural registration assumption refers to the smoothness of the transformation field. Local deformations cannot be independent in a pixel level and therefore this condition has to be also considered during the recovery of the corresponding field. This can be done by introducing a constraint (in terms of a penalty function) that accounts for smoothness on the field  $(u, v)$ :

$$\begin{aligned} E(s, \theta, T, (u, v)) &= \alpha \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^\tau))^2 \\ &+ (1 - \alpha)\beta \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^\tau + (u, v)))^2 \\ &+ (1 - \alpha)(1 - \beta) \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (u_x^2 + u_y^2 + v_x^2 + v_y^2) \end{aligned}$$

Let us now try to interpret the above functional. The first term aims at finding pixel-wise intensity (level set representation) correspondences using a global



motion model (rigid transformation). The second term, aims at correcting the correspondences in a pixel level using a local deformation model on top of the existing global model, while the third term constrains the deformation field to be locally smooth.

The minimization of this functional is done using a gradient descent:

$$\left\{ \begin{array}{l} \frac{d}{dt}s = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_S)(\Phi_{\mathcal{D}} + \nabla \Phi_S \cdot \nabla_s A^{\tau}) \\ \quad [\alpha (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau})) + \beta(1 - \alpha) (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau} + (u, v)))] \\ \frac{d}{dt}T = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_S) \left( \nabla \Phi_S \cdot \begin{pmatrix} \nabla_{T_x} A^{\tau} \\ \nabla_{T_y} A^{\tau} \end{pmatrix} \right) \\ \quad [\alpha (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau})) + \beta(1 - \alpha) (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau} + (u, v)))] \\ \frac{d}{dt}\theta = 2 \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_S)(\nabla \Phi_S \cdot \nabla_{\theta} A^{\tau}) \\ \quad [\alpha (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau})) + \beta(1 - \alpha) (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau} + (u, v)))] \\ \frac{d}{dt}u = 2(1 - \alpha)(1 - \beta) (u_{xx} + u_{yy}) + \\ \quad 2\beta(1 - \alpha)\nabla_x \Phi_S(A^{\tau} + (u, v)) (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau} + (u, v))) \\ \frac{d}{dt}v = 2(1 - \alpha)(1 - \beta) (v_{xx} + v_{yy}) + \\ \quad 2\beta(1 - \alpha)\nabla_y \Phi_S(A^{\tau} + (u, v)) (s\Phi_{\mathcal{D}} - \Phi_S(A^{\tau} + (u, v))) \end{array} \right.$$

The performance of the complete system is shown in [fig. (4)].

The selection of the  $\alpha$  parameter is a crucial component of the algorithm since it controls global-to-local factor. This is clearly shown in the motion equations where the construction of the local deformation field is delayed (actually it is done very slowly) until the parameters of the global rigid model are properly estimated.

The same property can be obtained through a two stage approach that involves the estimation of the global model first, and then the local estimates of the deformation field. However, such selection involves the use of time measures to control the gradient descent method and cannot be done automatically. One can claim that a similar argument is also valid for the approach presented in this paper. We can argue that the selection of  $\alpha$  parameter does affect the registration parameters (global and local estimates) but not the final registration result since by modifying  $\alpha$ , we theoretically change the estimates of the unknown parameters (global and local deformation measures) but not their joint estimates. This is clearly shown in [fig. (4)] where three different cases have been considered independently:

- A global rigid transformation [fig. (4.1)],
- A global-to-local rigid transformation with local deformations [fig. (4.2)],
- A local (pixel-wise registration) model [fig. (4.3)].

Based on the obtained results, we can claim that the registration performance of this representation for the three different type of transformations is comparable/similar. However, in order to overcome this limitation one can consider

modifying the objective function to penalize the formation of a rich local deformation field. The use of the local deformation field magnitude is a common selection to perform this task leading to the following example:

$$\begin{aligned}
E(s, \theta, T, (u, v)) = & \alpha \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^{\tau}))^2 \\
& + (1 - \alpha)\beta \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) (s\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A^{\tau} + (u, v)))^2 \\
& + (1 - \alpha)(1 - \beta) \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) [(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \gamma(u^2 + v^2)]
\end{aligned}$$

However, it is important to note that the introduction of local deformations cannot guarantee the proper handling of the shape to be registered. For example several points or parts of the source shape may be mapped to the same point or part of the target even in the case where scale variations are not present [fig. (4)]. Thus, the transformed shape may be quite different or even an open structure compared with the original one. In order to deal with this issue, we can consider the use membership functions for each pixel that enforces a "one-to-one" pixel-wise correspondence between the current and the target shape or

### 3.3 Supervised Registration and Joint Learning

We now consider the registration problem in more complex scenario where the target is not a shape, but a shape model with local degrees of variability. We assume the existence of this model that refers to a distance function  $[\Phi_{\mathcal{S}}(x, y)]$  associated with some variability measurements  $[\sigma_{\mathcal{S}}(x, y)]$  :

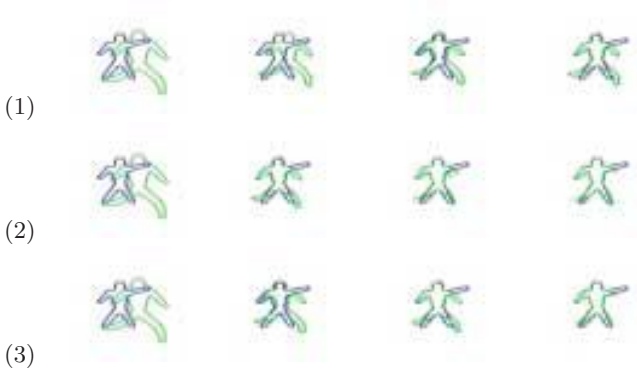
$$\Phi_{\mathcal{S}}(x, y) = \begin{cases} 0, & (x, y) \in \mathcal{S} \\ +D((x, y), \mathcal{S}) > 0, & (x, y) \in \mathcal{R}_{\mathcal{S}} \\ -D((x, y), \mathcal{S}) < 0, & (x, y) \in [\Omega - \mathcal{R}_{\mathcal{S}}] \end{cases}$$

Then, for a given pixel location  $(x, y)$  and a given value  $\phi$  the conditional probability of having this value at the location in  $\mathcal{S}$  is given by:

$$p_{\mathcal{S}(x, y)}(\phi) = \frac{1}{\sqrt{2\pi}\sigma_{\mathcal{S}}(x, y)} e^{-\frac{(\phi - \Phi_{\mathcal{S}}(x, y))^2}{2\sigma_{\mathcal{S}}^2(x, y)}}$$

The construction of this model can be done using variational framework that assumes the existence of a training set of shapes that are registered with respect to an arbitrary element of this set [17].

Given this model, a more challenging and promising task is to register a given shape  $\mathcal{D}$  by maximizing the maximum likelihood density between the model and the input shape. If we assume that the conditional densities of the model are independent across pixels, then the optimization criterion is equivalent with the



**Fig. 4.** (1) *Global*  $\{s = 0.79, \theta = 1.34^\circ, T_x = -16.34, T_y = -15.76\}$ , (2) *Global-to-Local*  $\{s = 0.81, \theta = 2.07^\circ, T_x = -14.97, T_y = -15.43\}$ , (3) *Local Registration with regularization constraints*.

minimization of:

$$\begin{aligned}
 E(s, \theta, T, (u, v)) = & \alpha \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) \left( \log(\sigma_{\mathcal{S}}(A)) + \frac{(\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A))^2}{2\sigma_{\mathcal{S}}^2(A)} \right) \\
 & + (1 - \alpha)\beta \iint_{\Omega} N_{\delta}(\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) \left( \log(\sigma_{\mathcal{S}}(A + (u, v))) + \frac{(\Phi_{\mathcal{D}} - \Phi_{\mathcal{S}}(A + (u, v)))^2}{2\sigma_{\mathcal{S}}^2(A + (u, v))} \right) \\
 & + (1 - \alpha)(1 - \beta) \iint_{\Omega} N_{\delta}(s\Phi_{\mathcal{D}}, \Phi_{\mathcal{S}}) [u_x^2 + u_y^2 + v_x^2 + v_y^2 + \gamma(u^2 + v^2)]
 \end{aligned}$$

One can easily interpret this function. Shape components are considered according to their variability (denominator factor) in the registration process. Therefore, the global registration component will be recovered using the rigid parts of the shape model, while local deformations will be estimated accordingly in a less significant manner due to the contribution of the variability estimates.

## 4 Conclusions

In this paper, we have proposed a novel simple framework based on variational principles for global to local shape registration. The proposed framework makes use of a powerful mathematical tool, the level set representations that is integrated with variational (sum of squared differences) and stochastic principles (Supervised Registration and Joint Learning) resulting in a scale, translation and rotation invariant paradigm for shape registration. The extension of the method to deal with structures of higher dimension is a straight-forward step that is currently under investigation, as well as to deal with open structures.

The efficiency of the proposed formulation is demonstrated using a very simple optimization criterion, the sum of squared differences. Therefore, we can presume that the selected shape representation is very powerful, and has strong discrimination power. Very encouraging [fig. (5)] experimental results were obtained using all variations of the proposed framework.



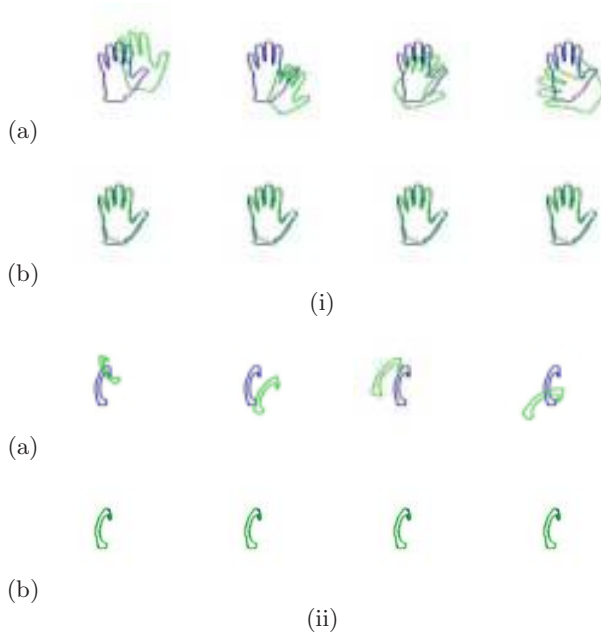
**Fig. 5.** *Global Registration for non-rigid objects: (1,2,3,4,5,6) Different examples. (a) Input Shapes, (b,c) Intermediate Results, (d) Registration Result.*

Conventional methods were considered to implement the obtained PDEs in this paper. Therefore, small time steps are required to guarantee stability and convergence. The estimation of the rigid transformation can be done in real time, due to the small number of involved parameters once an appropriate numerical approximation methods are used [25]. The current implementation takes up to a couple of seconds modulo the initial position of shapes. Recovering the complete local deformations field is very expensive.

In order to validate our approach, we have considered four different examples [fig. (6,7)] to validate the performance of the method. Cases with severe local deformations and topological changes resulting to missing and occluded parts have been also investigated. Towards this end, two fingers were progressively removed from the hand input shape [fig. (7)].

In order to perform reliable tests, the input shape has been globally and locally deformed. We have considered a four dimensional random variable  $(s, \theta, T_x, T_y)$  defined in the following space:

$$(\theta, s, T_x, T_y) \in \left( \left[ -\frac{\pi}{3}, \frac{\pi}{3} \right], [0.8, 1.2], [-30, 30], [-30, 30] \right)$$

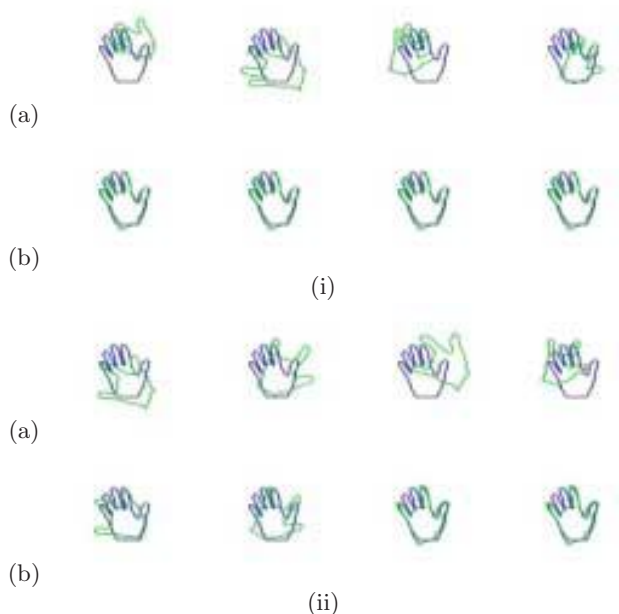


**Fig. 6.** *Empirical Evaluation (a) Initial Condition, (b) Registration Result. Performance (Registration Ratio): (i) 100 %, (ii) 100 %.*

For all examples we have ran a 100 trials using a random generation process for the parameters of the rigid transformation. The obtained results are shown in [fig. (6,7)]. Each column corresponds to a random trial. The first row (a), refers to the initial condition and the second row (b) to the final registration result. The registration performance of the algorithm is also shown.

As far the future directions of the proposed approach are concerned, several issues remain open. The acceleration of the method to perform real time registration is a natural step to be done. Moreover, the integration of image/intensity features into the shape registration algorithm can further improve the performance of the proposed framework. An hybrid approach that makes use primarily of the shape information and secondly of the image features can be a valuable element to the segmentation of medical structures where the global shape variability is not discriminant. Also, the investigation of more complex motion structures and global transformations is step to be done. Shape recognition is also an interesting application. One can consider (after alignment) the use of these representations as feature space to the recognition process.

The proposed framework can be considered static since it is based on already extracted shapes. In many cases, solving the segmentation problem cannot be done trivially and therefore shape registration becomes more complicated. To overcome this constraint, a joint optimization approach has to be investigated where both, the problems of segmentation and registration are treated simultaneously.



**Fig. 7.** *Empirical Evaluation (a) Initial Condition, (b) Registration Result. Performance (Registration Ratio): (i) 100 %, (ii) 77 %.*

**Acknowledgments.** The authors would like to thank C. Xu for fruitful discussions, the anonymous reviewers for their suggestions and Alok Gupta, the director of the IM Department at Siemens Corporate Research for encouraging the exploration of the topic and providing financial support to Mikael Rousson. We are also thankful to T. Sebastian and B. Kimia from Brown University for providing their shape database [19] and Professor Yunmei Chen, Department of Mathematics, Dr. Richard W. Briggs, Department of Radiology and Brain Institute, University of Florida and his group for providing the medical data.

## References

1. S. Belongie, J. Malik, and J. Puzicha. Matching Shapes. In *IEEE ICCV*, pages 456–461, Vancouver, Canada, 2001.
2. A. Blake and M. Isard. *Active Contours*. Springer-Verlag Press, 1997.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IEEE ICCV*, pages 694–699, Boston, USA, 1995.
4. C. Chef d’Hotel, G. Hermosillo, and O. Faugeras. A Variational Approach to Multi-Modal Image Matching. In *IEEE Workshop on Variational and Level Set Methods*, pages 21–28, 2001.
5. H. Chui and A. Rangarajan. A New Algorithm for Non-Rigid Point Matching. In *IEEE CVPR*, pages II: 44–51, Hilton Island, USA, 2000.

6. I. Cohen and I. Herlin. Curve Matching Using Geodesic Paths. In *IEEE CVPR*, pages 741–746, Santa Barbara, USA, 1998.
7. T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active Shape Models - their traing and applications. *CVGIP: Image Understanding*, 61, 1995.
8. A. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. volume 2, pages 411–420, 2001.
9. S. Joshi and M. Miller. Ladmark Matching via Large Deformation Diffeomorphism. *IEEE TIP*, 9:1357–1370, 2000.
10. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *IEEE ICCV*, pages 261–268, 1987.
11. B. Kimia, A. Tannenbaum, and S. Zucker. Shocks and Deformations i: The Components of two-dimensional Shape and the reaction-diffusion space. *IJCV*, 15:189–224, 1995.
12. D. Kozinska, O. Tretiak, J. Nissanov, and C. Ozturk. Multidimensional Alignment Using the Euclidean Distance Transform. *Graphical Models and Image Processing*, 6:373–385, 1997.
13. S. Lavalley and R. Szilinski. Recovery of the Position and Orientation of free-form Objects from Image Contours using 3D Distance Maps. *IEEE PAMI*, 17:378–390, 1995.
14. J. Maintz and M. Viergever. A Survey for Medical Image Registration. *Medical Image Analysis*, 2:1–36, 1998.
15. R. Malladi and J. Sethian. A unified framework for shape segmentation representation, and recognition. Technical Report LBL-36069 UC-405, Lawrence Berkeley Laboratory, Berkeley, 1994.
16. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : algorithms based on the hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
17. N. Paragios and M. Rousson. Shape Priors for Level Set Representations. Copenhagen, Denmark, 2002.
18. A. Rosenfeld and J. Pfaltz. Distance Functions on Digital Pictures. *Pattern Recognition*, 1:33–61, 1968.
19. T. Sebastian, P. Klein, and B. Kimia. Recognition of Shapes by Editing Shock Graphs. In *IEEE ICCV*, pages 755–762, Vancouver, Canada, 2001.
20. J. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
21. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shocks Graphs and Shape Matching. *IJCV*, 35:13–32, 1999.
22. K. Siddiqi, A. Shokoufandeh, S. Dikinson, and S. Zucker. Shock Graphs and Shape Matching. In *IEEE ICCV*, pages 222–229, Bombay, India, 1998.
23. R. Veltkamp and M. Hagedoorn. State-of-the-art in Shape Matching. Technical Report UU-CS-1999-27, Utrecht University, Sept. 1999.
24. P. Viola and W. Wells. Aligment by Maximization of Mutual Information. In *IEEE ICCV*, pages 16–23, Boston, USA, 1995.
25. J. Weickert, B. M. t. Haar Romeny, and M. Viergener. Efficient and Reliable Scheme for Non-Linear Diffusion and Filtering. *IEEE TIP*, 7:398–410, 1998.
26. S. Zhu and A. Yuille. FORMS: A flexible object recognition and modeling system. *IJCV*, 20:187–212, 1996.

# Shape from Shading and Viscosity Solutions

Emmanuel Prados<sup>1</sup>, Olivier Faugeras<sup>1</sup>, and Elisabeth Rouy<sup>2</sup>

<sup>1</sup> INRIA, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France  
{emmanuel.prados,olivier.faugeras}@sophia.inria.fr

<sup>2</sup> Institut National des Sciences Appliquées de Lyon, MAPLY - UMR5585,  
Laboratoire de Modélisation Mathématique et de Calcul Scientifique, Bâtiment 401,  
20 Avenue Albert Einstein, 69621 VILLEURBANNE Cedex, France  
Elisabeth.Rouy@ec-lyon.fr

**Abstract.** This article presents an approach to the shape from shading problem which is based upon the notion of viscosity solutions to the shading partial differential equation, in effect a Hamilton-Jacobi equation. The power of this approach is twofolds: 1) it allows nonsmooth, i.e. nondifferentiable, solutions which allows to recover objects with sharp troughs and creases and 2) it provides a framework for deriving a numerical scheme for computing approximations on a discrete grid of these solutions as well as for proving its correctness, i.e. the convergence of these approximations to the solution when the grid size vanishes.

Our work extends previous work in the area in three aspects. First, it deals with the case of a general illumination in a simpler and a more general way (since they assume that the solutions are continuously differentiable) than in the work of Dupuis and Oliensis [9]. Second, it allows us to prove the existence and uniqueness of “continuous” solutions to the shading equation in a more general setting (general direction of illumination) than in the work of Rouy and Tourin [24], thereby extending the applicability of shape from shading methods to more realistic scenes. Third, it allows us to produce an approximation scheme for computing approximations of the “continuous” solution on a discrete grid as well as a proof of their convergence toward that solution.

**Keywords:** Shape from Shading, viscosity solutions, existence and uniqueness of a solution, Hamilton-Jacobi equations, dynamic programming principle, approximation and numerical schemes.

## 1 Introduction

Shape from shading has been a central problem in the field of computer vision since the early days. The problem is to compute the three-dimensional shape of a surface from the brightness variations in a black and white image of that surface. The work in our field was pioneered by Horn who was the first to pose the problem as that of finding the solution of a nonlinear first-order partial differential equation called the brightness equation [15]. This initial idea was limited by the particular numerical method that was used (the method of characteristics) and was enriched by posing the problem as a variational problem [14] within which



additional constraints such as those provided by occluding contours [17] can be taken into account. The book [13] contains a very nice panorama of the research in this area up to 1989. Questions about the existence and uniqueness of solutions to the problem were simply not even posed at that time with the important exception of the work of Bruss [4]. These questions as well as those related to the convergence of numerical schemes for computing the solutions became central in the last decade of the 20th century. Brightness equations that do not admit continuously differentiable solution were produced [3,16], Durou and his co-workers showed that some well-known numerical schemes were in fact almost never convergent [10] and exhibited a continuous family of ambiguous solutions [11]. A breakthrough was achieved by people who realized that control theory could be brought to bear on this problem. Dupuis and Oliensis showed that this theory provided a way of constructing numerical schemes with provable convergence properties in the case where a continuously differentiable solution existed [9]. More significantly perhaps, P.-L. Lions, Rouy and Tourin used the theory of viscosity solution of Hamilton-Jacobi equations to characterize the existence and uniqueness of weak solutions to the brightness equation and to come up with provably convergent numerical schemes to compute them [24,21]. In doing so, they considerably generalized the applicability of shape from shading since solutions could be only continuous and they opened the way to the mathematically well-posed use of such constraints as occluding edges and shadows as well as general light sources.

In this article we revisit one of the simplest versions of the shape from shading problem, the idea being that the tools that we develop here will be extendable to more general and realistic situations. We therefore assume that the camera performs an orthographic projection of the scene (hence a simple affine camera model as opposed to a pinhole), that the scene is illuminated by a single point source at infinity, that its reflectance is Lambertian and its albedo constant and equal to 1. We also assume that there are no shadows and no occluding boundaries and that the distance from the camera to the scene is known on the boundary of the image. Admittedly, these hypotheses may appear a bit restrictive. In fact they are not in the sense that they can be generalized without drastically changing the mathematical analysis that is done in this paper. But this will be the subject of another paper.

We denote by  $u$  the distance of the points in the scene to the camera,  $I$  the image intensity,  $\mathbf{L} = (\alpha, \beta, \gamma)$  the unit vector representing the direction of the light source ( $\gamma > 0$ ), and  $\mathbf{l} = (\alpha, \beta)$ . The image is modelled as a function from the closure  $\overline{\Omega}$  of an open set  $\Omega$  of  $\mathbb{R}^2$  into the closed interval  $[0, 1]$ ,  $I : \overline{\Omega} \rightarrow [0, 1]$ . Given our hypotheses, the shape from shading problem is, given  $I$  and  $\mathbf{L}$ , to find a function  $u : \overline{\Omega} \rightarrow \mathbb{R}$  satisfying the brightness equation:

$$\forall x \in \Omega, \quad I(x) = \frac{-\nabla u(x) \cdot \mathbf{l} + \gamma}{\sqrt{1 + |\nabla u(x)|^2}}, \quad (1)$$

$$\text{with the Dirichlet boundary conditions } \forall x \in \partial\Omega, \quad u(x) = \varphi(x), \quad (2)$$

$\varphi$  being continuous on  $\partial\Omega$ .

Note that in the case where the light source is in the same direction as the direction of projection (it is the case considered in [24]) we have  $\mathbf{L} = (0, 0, 1)$ , and the PDE (1) is equivalent to an Eikonal equation:

$$|\nabla u(x)| = \sqrt{\frac{1}{I(x)^2} - 1}. \quad (3)$$

Note also that (1) is a Hamilton-Jacobi equation and can be rewritten as  $H(x, \nabla u(x)) = 0$ , where  $H$  is the Hamiltonian (see section 5).

The paper is organized as follows. In section 2 we briefly introduce the notion of viscosity solutions of Hamilton-Jacobi solutions. In sections 3 and 4 we give classical theorems of existence and uniqueness of a (viscosity) solution. In section 5 we apply these theorems to the shape from shading problem. In section 6 we propose a numerical scheme for solving the brightness equation and prove its convergence to the continuous solution. In section 7 we show some experimental results on synthetic and real images and conclude in section 8.

## 2 Viscosity Solutions

The notion of viscosity solutions of Hamilton-Jacobi equations has been introduced by Crandall and Lions [6,20,8,7]. It is a very nice way of making quantitative and operational the intuitive idea of weak solutions of first-order (and for that matter, second-order) Partial Differential Equations (PDEs). In the context of the shape from shading problem we are only concerned with first-order PDEs.

Let us consider a Hamilton-Jacobi equation, i.e.:

$$H(x, u(x), \nabla u(x)) = 0 \quad \text{in } \Omega, \quad \text{an open set of } \mathbb{R}^n, \quad (4)$$

where  $H$  is a continuous function from  $\overline{\Omega} \times \mathbb{R} \times \mathbb{R}^n$  to  $\mathbb{R}$ . We start with the definition of viscosity subsolution and supersolution:

**Definition 1** *A viscosity subsolution of equation (4) is a function  $u$  satisfying*

$$\forall \phi \in C^1(\Omega), \forall x_0 \in \Omega \text{ local maximum of } (u - \phi), \quad H(x_0, u(x_0), \nabla \phi(x_0)) \leq 0.$$

*A viscosity supersolution of equation (4) is a function  $u$  satisfying*

$$\forall \phi \in C^1(\Omega), \forall x_0 \in \Omega \text{ local minimum of } (u - \phi), \quad H(x_0, u(x_0), \nabla \phi(x_0)) \geq 0.$$

*Finally,*

- $u \in BUC(\Omega)$  (the set of bounded uniformly continuous functions) is a continuous viscosity solution of equation (4) if it is both a subsolution and a supersolution.
- $u$  is a discontinuous viscosity solution if  $u^*$  (respectively  $u_*$ ) is a subsolution of  $H_*$  (respectively a supersolution of  $H^*$ ):  $u^*$  (respectively  $u_*$ ) is the upper semicontinuous (respectively lower semicontinuous) envelope of  $u$ ; Of course, in this case, we do not suppose  $H$  continuous on  $\overline{\Omega}$ .

As we see in the definition, viscosity solutions are not necessarily differentiable! In this sense they are weak solutions of (4). Nonetheless, the following proposition ensures the coherence between continuous viscosity and classical solutions.

**Proposition 1** *Let  $u$  be differentiable in  $\Omega$  and a classical solution of (4). If  $u \in BUC(\Omega)$ , then it is a viscosity solution.*

*Let  $u$  be a viscosity solution of (4). If  $u$  is differentiable in  $\Omega$ , then it is a classical solution.*

For more details on the notion of viscosity solutions, the interested reader is referred to [20,2].

In the shape from shading application we consider in this article, we are interested in a special case of equation (4) where the Hamiltonian  $H$  does not depend upon the function  $u$ :

$$H(x, \nabla u(x)) = 0 \quad \text{in } \Omega, \quad \text{an open set of } \mathbb{R}^2, \quad (5)$$

We restrict ourselves from now on to this case and give general results of existence and uniqueness of viscosity solutions of (5).

### 3 Uniqueness of a Viscosity Solution

The uniqueness results depend upon the type of boundary conditions that we impose. In this article we deal with the Dirichlet condition. For the continuous case, Rouy and Tourin give the following uniqueness result for Hamiltonians  $H$  which do not depend upon  $u$  (see [24]). For more general conditions, see [21].

**Theorem 1** *Let  $H : \overline{\Omega} \times \mathbb{R}^2 \rightarrow \mathbb{R}$  where  $\Omega$  is a bounded open set of  $\mathbb{R}^2$ . If*

1.  $\forall x, y \in \Omega, \forall p \in \mathbb{R}^2, |H(x, p) - H(y, p)| \leq w(|x - y|(1 + |p|))$ , where  $w$  is a continuous nondecreasing function such that  $w(0) = 0$ ,
2.  $H$  is continuous in  $\overline{\Omega} \times \mathbb{R}^2$  and convex with respect to  $\nabla u$ ,
3. there exists a strict viscosity subsolution  $\underline{u} \in C^1(\Omega) \cap C(\overline{\Omega})$  of (5) (i.e. such that  $H(x, \nabla \underline{u}(x)) < 0$  for all  $x$  in  $\Omega$ ),

*then there exists at most one continuous viscosity solution of (5) verifying  $u = \varphi$  in  $\partial\Omega$ , where  $\varphi \in C(\partial\Omega)$ .*

For the discontinuous case, we have a "hard" uniqueness result (see [2]):

**Theorem 2** *if  $H$  satisfies the hypotheses "HNCL" (defined below) and if there exists a constant  $C > 0$  such that for all  $p$  in  $\mathbb{R}^2$ ,*

- $\forall x \in \partial\Omega, H(x, p + \lambda\eta(x)) \leq 0 \implies \lambda \leq C(1 + |p|)$ ,  $\eta(x)$  is the unit outward pointing normal vector to  $\partial\Omega$ .
- $H(x, p - \lambda\eta(x)) \rightarrow +\infty$  uniformly with respect to  $x$ , when  $\lambda \rightarrow +\infty$ ;

*then for all subsolution  $u$  and supersolution  $v$  (in discontinuous sense),  $u \leq v$  in  $\Omega$ .*

We then have the uniqueness of discontinuous solution on  $\Omega$  (but not in  $\overline{\Omega}$ !). The hypotheses "HNCL" are:

1. There exists a function  $m_R$  which goes to zero at zero, such that  $\forall x, y \in \Omega, \forall p \in \mathbb{R}^2, |H(x, p) - H(y, p)| \leq m_R(|x - y|(1 + |p|))$
2. For all  $x$  in  $\Omega$ ,  $H(x, p)$  is convex with respect to  $p$ .
3. There exists a function  $\Phi$  of class  $C^1$  in  $\Omega$ , continuous in  $\overline{\Omega}$  such that  $H(x, D\Phi(x)) \leq \delta < 0$  in  $\Omega$
4. There exists a function  $m_R$  which goes to zero at zero, such that  $\forall x \in \partial\Omega, \forall p, q \in \mathbb{R}^2, |H(x, p) - H(x, q)| \leq m_R(|p - q|)$ .

## 4 Existence of Viscosity Solution, Value Functions, and Boundary Compatibility Conditions

We recall here theorems of existence of viscosity solutions in the special case where the Hamiltonian  $H$  appearing in equation (5) is convex with respect to  $\nabla u$ . We also consider the case of Dirichlet boundary conditions. In the case of continuous solutions on  $\overline{\Omega}$ , the existence theorem can be interpreted as giving compatibility constraints for the boundary conditions.

We note  $H^*$  the Legendre transform of  $H$ :

$$H^*(x, q) = \sup_{p \in \mathbb{R}^2} \{p \cdot q - H(x, p)\} \leq +\infty.$$

The following theorem is a special case of theorem 5.3 in [20].

**Theorem 3** *If*

1.  $H \in C(\overline{\Omega} \times \mathbb{R}^2)$  is convex with respect to  $\nabla u$  for all  $x$  in  $\overline{\Omega}$ ,
2.  $H(x, p) \rightarrow +\infty$  when  $|p| \rightarrow +\infty$  uniformly with respect to  $x \in \overline{\Omega}$ ,
3.  $\inf_{p \in \mathbb{R}^2} H(x, p) \leq 0$  in  $\overline{\Omega}$ ,
4.  $\forall x, y \in \partial\Omega, \varphi(x) - \varphi(y) \leq L(x, y);$   
( where  $L(x, y) = \inf_{\xi \in C_{x,y}} \{ \int_0^{T_0} H^*(\xi(s), -\xi'(s)) ds$  and  $C_{x,y} = \{ \xi \mid \xi(0) = x, \xi(T_0) = y, \forall t \in [0, T_0], \xi(t) \in \overline{\Omega}, \xi' \in L^\infty(0, T_0) \}$  )

then the function  $u$  defined in  $\overline{\Omega}$  by:

$$u(x) = \inf_{\xi \in C_{x,y}, y \in \partial\Omega} \left\{ \int_0^{T_0} H^*(\xi(s), -\xi'(s)) ds + \varphi(\xi(T_0)) \right\}$$

is a continuous viscosity solution of equation (5) verifying  $u = \varphi$  in  $\partial\Omega$ .

Remark: if the hypotheses of theorem 1 are satisfied, we can reformulate its conclusion as follows: *The equation (5) has a unique viscosity solution iff*

$$\forall x, y \in \partial\Omega, \quad \varphi(x) - \varphi(y) \leq L(x, y). \quad (6)$$

At last, for discontinuous solutions, we have the theorem (see theorem 4.13 of [1])

**Theorem 4** *If  $H(x, p) = \sup_{q \in Q} \{-f(x, q) \cdot p - l(x, q)\}$ , with  $Q \subset \mathbb{R}^2$  compact,  $f : \mathbb{R}^2 \times Q \rightarrow -\mathbb{R}^2$  Lipschitz continuous in  $x$  uniformly in  $a$ , with  $l : \mathbb{R}^2 \times Q \rightarrow -\mathbb{R}$  continuous such that there exist constants  $\omega_l$  and  $M$  such that  $|l(x, q) - l(y, q)| \leq \omega_l(|x - y|)$  and  $|l(x, q)| \leq M, \quad \forall x, y \in \mathbb{R}^2$  and  $q \in Q$ ; then  $u$  (defined above) is a viscosity solution of (5) with Dirichlet condition (2) in the discontinuous sense.*

One very nice thing about these theorems is that not only do they guarantee the existence of viscosity solutions, but they also show that the solution is the value function of an optimal control problem and provide a way of constructing one explicitly.

## 5 Application to the Shape from Shading Problem

As shown in section 1, the shape from shading equation is of the form  $H(x, \nabla u) = 0$ , where  $H(x, p) = I(x)\sqrt{1 + |p|^2} + p \cdot \mathbf{l} - \gamma$ . It is easy to verify that if  $I(x)$  does not reach the value 1 within  $\Omega$ , i.e.  $0 \leq I(x) < 1 \quad \forall x \in \Omega$ , the function  $\tilde{u} : (x, y) \mapsto \frac{1}{\gamma}(-\alpha x - \beta y)$  is a strict viscosity subsolution of (5). If  $I$  is Lipschitz continuous, the other hypotheses of theorem 1 are also verified (as the reader will readily check) and we conclude that there exists at most one continuous viscosity solution of the PDE

$$\begin{cases} I(x)\sqrt{1 + |\nabla u(x)|^2} + \nabla u(x) \cdot \mathbf{l} - \gamma = 0 & \forall x \in \Omega \\ u(x) = \varphi(x) & \forall x \in \partial\Omega \end{cases} \quad (7)$$

Otherwise, under these same hypotheses, theorem 2 easily applies oneself as soon as  $I > |\mathbf{l}|$ ; thus in this case, we have also the uniqueness of discontinuous solution on  $\Omega$  (but not on  $\bar{\Omega}$ !). In practice,  $I$  can reach the value 1 in an arbitrary compact set in  $\bar{\Omega}$ . This implies that there does not exist a strict viscosity subsolution and we lose uniqueness. The loss of uniqueness is completely characterized in [21] and we summarize it here. We assume that there exists a finite collection of disjoint connected compact sets  $K_i, i = 1, \dots, n$  such that  $\{I = 1\} = \cup_{i=1}^n K_i$ . The main points are that the viscosity solutions of (7) are constant over the  $K_i$ 's and, when  $n > 1$ , we only need to specify the differences of the values of  $u$  in the  $K_i$ 's. This allows us to ignore the set  $\{I = 1\}$  and to work in the open set  $\Omega' = \Omega - \{I = 1\}$ . In other words, when the set  $\{I = 1\}$  is not empty we consider the problem

$$\begin{cases} I(x)\sqrt{1 + |\nabla u(x)|^2} + \nabla u(x) \cdot \mathbf{l} - \gamma = 0 & \forall x \in \Omega' \\ u(x) = \varphi(x) & \forall x \in \partial\Omega' \end{cases} \quad (8)$$

rather than (7). Like this, as soon as  $I$  is Lipschitz continuous, we have a uniqueness result for problem (8). The solutions of (7) are then obtained from these by choosing (almost) arbitrary values for  $u$  in the  $K_i$ 's. Another possibility is to choose among all solutions one which possesses an extra property, as in the work of M. Falcone et al. [5,12] where the uniqueness is obtained by choosing the *maximal* solution.

Since  $I$  is continuous,  $H$  is continuous in  $\bar{\Omega} \times \mathbb{R}^2$  and convex with respect to  $p$ ; we easily obtain, taking the derivative:

$$\inf_{p \in \mathbb{R}^2} H(x, p) = \begin{cases} \sqrt{I(x)^2 - |\mathbf{l}|^2} - \gamma & \text{if } I(x)^2 \geq \alpha^2 + \beta^2. \\ -\infty & \text{otherwise.} \end{cases}$$

Since  $I^2 \leq \alpha^2 + \beta^2 + \gamma^2 = 1$ , we have  $\inf_{p \in \mathbb{R}^2} H(x, p) \leq 0$ . Finally, since  $\Omega$  is bounded hypothesis 2 in theorem 3 is satisfied iff  $I > |\mathbf{l}|$  and, if the compatibility

conditions (6) are satisfied on  $\partial\Omega'$ , we have obtained existence and uniqueness of a continuous viscosity solution of problem (8). In the discontinuous case, the theorem 4 applied with  $A = \bar{B}(0, 1)$ ,  $f(x, q) = -\mathbf{1} - I(x)q$  and  $l(x, q) = -I(x)\sqrt{1 - |q|^2} - \gamma$ , is valid as soon as  $I$  is Lipschitz in  $\bar{\Omega}$  (and even if  $I^2 < \alpha^2 + \beta^2$  or  $I = 0$ !).

Finally, in both cases,

$$u(x) = \inf_{\xi} \left\{ \int_0^{T_0} H^*(\xi(s), -\xi'(s)) ds + \varphi(\xi(T_0)) \right\} \quad (9)$$

where  $\xi \in \cup_{y \in \partial\Omega} C_{x,y}$  satisfies  $\forall t \in [0, T_0]$ ,  $-\xi'(t) \in \bar{B}(\mathbf{1}, I(\xi(t)))$ , is a solution, and we can calculate  $H^*$  explicitly through differential calculus:

$$H^*(x, q) = \begin{cases} -\sqrt{I(x)^2 - |q - \mathbf{1}|^2} + \gamma & \text{if } |q - \mathbf{1}| \leq I(x) \\ +\infty & \text{otherwise.} \end{cases}$$

## 6 Numerical Approximation of the Solution

The purpose of this section is to give an algorithm for approximating the viscosity solution of (8). We also prove that our algorithm is convergent in the sense that when the spatial mesh size tends to 0, the discrete solution tends to the viscosity solution.

### 6.1 Approximation Scheme

Let  $\Omega$  be the rectangular domain  $]0, X[ \times ]0, Y[$  of  $\mathbb{R}^2$ . For all mesh size  $\rho = (\Delta x, \Delta y)$ , we want to approximate the value of the solution  $u$  at the points  $(x_{ij}) = (i\Delta x, j\Delta y)$  by  $U_{ij}$  (for  $i = 1..N$  and  $j = 1..M$  where  $N = \lfloor X/\Delta x \rfloor$  and  $M = \lfloor Y/\Delta y \rfloor$ ).

Since  $u$  is a value function, the dynamic programming principle allows to write that for all  $\tau \geq 0$ ,  $u(x)$  is equal to the *inf* of

$$\int_0^{T_0 \wedge \tau} H^*(\xi(s), \xi'(s)) ds + \varphi(\xi(T_0)) \chi_{\{T_0 \leq \tau\}}(T_0) + u(\xi(\tau)) \chi_{\{T_0 > \tau\}}(T_0), \quad (10)$$

where  $a \wedge b$  is the smallest of  $a$  and  $b$ . The *inf* is taken over the set of the  $\xi$ s of  $\cup_{y \in \partial\Omega'} C_{x,y}$  satisfying  $\forall t \in [0, T_0]$   $-\xi'(t) \in \bar{B}(\mathbf{1}, I(\xi(t)))$ .

$\rho$  being supposed fixed, we can choose  $\tau$  sufficiently small so that for all  $x_{ij} \in \Omega'$ :

$$u(x) = \inf_{\xi} \left\{ \int_0^{\tau} H^*(\xi(s), \xi'(s)) ds + u(\xi(\tau)) \right\}. \quad (11)$$

To come up with our numerical scheme we use the following approximations:

$$\int_0^{\tau} H^*(\xi(s), \xi'(s)) ds \cong H^*(\xi(0), \xi'(0))\tau, \quad \xi(\tau) \cong \xi(0) + \xi'(0)\tau,$$

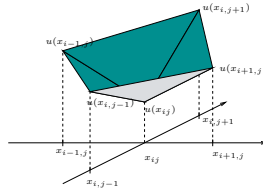
$$\bar{B}(-(\alpha, \beta), I(\xi(t))) \cong \bar{B}(-\mathbf{1}, I(\xi(0)))$$

Since  $\xi(0) = x$ , equation (11) now becomes:

$$\sup_{\xi'(0) \in B_x} \left\{ \frac{u(x) - u(x + \tau \xi'(0))}{\tau} - H^*(x, \xi'(0)) \right\} = 0; \quad (12)$$

where  $B_x = \bar{B}(-(\alpha, \beta), I(x))$ .

The final step is to approximate  $u$  by a piecewise affine function in a neighbourhood of  $x$ , see figure 1. The goal is to obtain a relation between  $x_{ij}$ ,  $x_{i \pm 1, j}$



**Fig. 1.** Piecewise affine approximation of  $u$  in a neighborhood of  $x_{ij}$

and  $x_{i,j \pm 1}$ . In order to reach this goal, we need to compute the sup in (12) and for that matter we partition the set  $Q$  in four subsets  $Q_1 \cup Q_2 \cup Q_3 \cup Q_4$ ,

$$Q_1 = \{q \in B_x | q_x > 0, q_y \geq 0\}, Q_2 = \{q \in B_x | q_x \leq 0, q_y \geq 0\}, \\ Q_3 = \{q \in B_x | q_x \leq 0, q_y < 0\}, Q_4 = \{q \in B_x | q_x > 0, q_y < 0\};$$

over which we independently maximize (12). To simplify matters but without loss of generality, we apply a rotation of axis  $Oz$ , and assume that  $\beta = 0$ . In this case we note that when  $x \in \{x \in \Omega | I(x) \leq |(\alpha, 0)|\}$ ,  $Q_1$  and  $Q_4$  are empty.

Let us detail the steps for  $Q_2$ . The piecewise affine approximation of  $u$  yields:

$$\frac{u(x_{ij}) - u(x_{ij} - \tau(q_x, q_y))}{\tau} \cong -q_x a(\Delta x, x_{ij}, u_{x_{ij}}, u) + q_y d(\Delta y, x_{ij}, u_{x_{ij}}, u),$$

where  $a(\rho, x, t, u) = \frac{t - u(x - (\rho, 0))}{\rho}$  and  $d(\rho, x, t, u) = \frac{t - u(x + (0, \rho))}{\rho}$ . Using convexity and differential calculus we find that the maximum over  $Q_2$  of (12) is equal to: First case: if  $I(x) > \alpha$ :

$$I(x) \sqrt{1 + (\chi^+(a, d^+))^2 + (d^+)^2} + \alpha \chi^+(a, d^+) - \gamma$$

where

$$\chi^-(x, y) = \begin{cases} x & \text{if } x \leq -\alpha \frac{\sqrt{1+y^2}}{\sqrt{I^2 - \alpha^2}} \\ -\alpha \frac{\sqrt{1+y^2}}{\sqrt{I^2 - \alpha^2}} & \text{otherwise.} \end{cases}$$

and

$$\chi^+(x, y) = \begin{cases} x & \text{if } x \geq -\alpha \frac{\sqrt{1+y^2}}{\sqrt{I^2 - \alpha^2}} \\ -\alpha \frac{\sqrt{1+y^2}}{\sqrt{I^2 - \alpha^2}} & \text{otherwise.} \end{cases}$$

Second case: if  $I(x) \leq \alpha$ :

$$I(x)\sqrt{1+a^2+(d^+)^2}+\alpha a-\gamma.$$

We then collect the results of the maximization over  $Q_1$ ,  $Q_2$ ,  $Q_3$  and  $Q_4$ , let  $\Delta x = \Delta y = \rho$ , and obtain the following numerical scheme:

$$S(\rho, x, t, u) = 0, \quad (13)$$

where  $S$  is equal to  $\rho T(\rho, x, t, u)$ , and  $T$  is defined by:

1. If  $x \in \Omega^\rho$  and  $I(x) > \alpha$  then

$$T(\rho, x, t, u) = \max( K_1(a, c, d), K_2(b, c, d) ), \quad (14)$$

where

$$\begin{aligned} K_1(a, c, d) &= I(x)\sqrt{1+(\chi^+(a, M))^2+M^2}+\alpha\chi^+(a, M)-\gamma, \\ K_2(b, c, d) &= I(x)\sqrt{1+(\chi^-(-b, M))^2+M^2}+\alpha\chi^-(-b, M)-\gamma, \\ M &= \max(c^+, d^+), \end{aligned}$$

and

$$\begin{aligned} a &= \frac{t-u(x-(\rho, 0))}{\rho} & b &= \frac{t-u(x+(\rho, 0))}{\rho} & c &= \frac{t-u(x-(0, \rho))}{\rho} \\ & & & & d &= \frac{t-u(x+(0, \rho))}{\rho}. \end{aligned} \quad (15)$$

2. If  $x \in \Omega^\rho$  and  $I(x) \leq \alpha$  then  $T(\rho, x, t, u) = I(x)\sqrt{1+a^2+M^2}+\alpha a-\gamma$ .

3. If  $x \in \partial\Omega^\rho$  then

$$T(\rho, x, t, u) = t - \varphi(x). \quad (16)$$

We have noted  $\Omega^\rho = \{x \in \Omega | x - (\rho, 0), x + (\rho, 0), x - (0, \rho), x + (0, \rho) \in \overline{\Omega}\}$  and  $\partial\Omega^\rho = \overline{\Omega} - \Omega^\rho$ .

## 6.2 Convergence of the Approximation Scheme

The following theorem is proved in [23]:

**Theorem 5** *Let  $T(\rho, x, t, u) = 0$  be an approximation scheme which can be written as  $g(x, a, b, c, d) = 0$ , where  $a, b, c, d$  are defined in (15). Let us assume that the approximation scheme  $S$  satisfies the hypotheses "MSC" (defined below), then for all positive  $\rho$ , the scheme has a solution noted  $u^\rho$ . If furthermore  $H$  satisfies the hypotheses of the hard uniqueness theorem (theorem 2), then  $u^\rho$  converges toward the solution of the corresponding PDE when  $\rho \rightarrow 0$ .*

The hypotheses "MSC" are:



1.  $g$  is increasing with respect to each of the four variables  $a, b, c$  and  $d$ ;
2. there exists a bounded function  $u_0$  such that  $\forall x \in \overline{\Omega}, T(\rho, x, u_0(x), u_0) \leq 0$ ;
3. for all  $\rho, x$  and  $u$ ,  $\lim_{t \rightarrow +\infty} T(\rho, x, t, u) \geq 0$ ;
4.  $g(x, a, b, c, d) \leq 0 \implies a, b, c$  or  $d$  are bounded independently of  $x$ ;
5.  $g$  is continuous in  $\overline{\Omega} \times \mathbb{R}^4$  and  $\forall x \in \overline{\Omega}, \forall \phi \in C_b^\infty(\overline{\Omega})$ ,

$$g(x, \partial_x \phi(x), -\partial_x \phi(x), \partial_y \phi(x), -\partial_y \phi(x)) = H(x, \phi(x), \nabla \phi(x))$$

Let us emphasize the fact that when the scheme is obtained by the process described in subsection 6.1, most “MSC” hypotheses are systematically verified. In particular, it is easily to apply the theorem 5, in the case for all  $x$  in  $\overline{\Omega}$ ,  $I^2(x) > \alpha^2 + \beta^2$ . For more details, see [23].

### 6.3 Algorithm for Computing the Solution

Thanks to theorem 5, we know that when  $\rho = (\Delta x, \Delta y)$  tends to zero, the solutions  $u^\rho$  of the numerical scheme (13) converge to the unique viscosity solution of equation (8). We now describe an algorithm that computes an approximation of  $u^\rho$ , for each value of  $\rho > 0$ . It is important to keep in mind that this algorithm converges toward  $u^\rho$  but not toward the viscosity solution. We assume that the image  $I$  is known on the discrete grid  $(x_{ij})_{i=1..N, j=1..M}$ . The algorithm consists of the following computation of the sequence of the values  $U_{ij}^n, n \geq 0$ :

- Algorithm 1**
1. *Initialisation* ( $n = 0$ ):  $U_{ij}^0 = u_0(x_{ij})$ .
  2. *Choice of a pixel  $x_{ij} \in \Omega'$  and modification (step  $n + 1$ ) of  $U_{ij}^n$ :*  
 We choose  $U^{n+1} = \sup\{V = (V_{k,l})_{(k,l) \in \overline{\Omega}} \text{ such that } \forall (k,l) \neq (i,j), V_{kl} = U_{kl}^n \text{ and } g(\rho, x_{ij}, V_{ij}, V) = 0\}$ .
  3. *Choose the next pixel  $x_{ij} \in \Omega'$  in such a way that all pixels of  $\Omega'$  are regularly visited and go back to 2.*

We have the following

**Theorem 6** *If the hypotheses of theorem 5 are satisfied, the algorithm 1 is well-defined (step 2 always yields a value) and the sequence  $U^n$  is increasing and converges toward  $u^\rho$  when  $n \rightarrow +\infty$ .*

Let us note

$$K_1(t) = I(x) \sqrt{1 + (\chi^+(D_x^- U_{ij}(t), M(t)))^2 + M(t)^2} + \alpha \chi^+(D_x^- U_{ij}(t), M(t)) - \gamma,$$

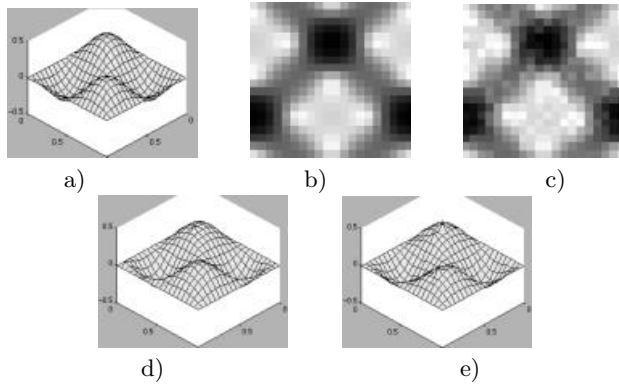
$$K_2(t) = I(x) \sqrt{1 + (\chi^-(D_x^+ U_{ij}(t), M(t)))^2 + M(t)^2} + \alpha \chi^-(D_x^+ U_{ij}(t), M(t)) - \gamma,$$

and

$$K(t) = \text{Max}(K_1(t), K_2(t))$$

where

$$\begin{aligned} D_x^- U_{ij}(t) &= \frac{t - U_{i-1j}}{\Delta x} & D_x^+ U_{ij}(t) &= \frac{U_{i+1j} - t}{\Delta x} \\ D_y^- U_{ij}(t) &= \frac{t - U_{ij-1}}{\Delta y} & D_y^+ U_{ij}(t) &= \frac{U_{ij+1} - t}{\Delta y}; \end{aligned}$$



**Fig. 2.** Results for a synthetic image generated by a sinusoidal surface sampled on a grid of size  $20 \times 20$  with  $\theta = 28^\circ$ : a) original surface, b) original image, c) noisy image; d) surface reconstructed from b):  $n = 40$ ,  $\epsilon_1 = 10.0\%$ ,  $\epsilon_2 = 9.9\%$ ,  $\epsilon_3 = 15.2\%$ ; e) surface reconstructed from c):  $n = 62$ ,  $\epsilon_1 = 12.8\%$ ,  $\epsilon_2 = 10.5\%$ ,  $\epsilon_\infty = 15.7\%$

$$M(t) = \max((D_y^- U_{ij}(t))^+, (D_y^+ U_{ij}(t))^-).$$

We note  $U_{minj} = \min(U_{i,j+1}, U_{i,j-1})$ .

For example, in the case where  $I(x_{ij}) > \alpha$ , Step 2 of algorithm 1 computes the value of  $t$  such that  $K(t) = 0$ , i.e. such that:

$$\left( \underbrace{K_1(t) = 0 \text{ and } K_2(t) \leq 0}_{\text{Case 1}} \right) \quad \text{or} \quad \left( \underbrace{K_2(t) = 0 \text{ and } K_1(t) \leq 0}_{\text{Case 2}} \right)$$

$t$  is therefore either a root of  $K_1$ , or a root of  $K_2$ . Let us detail the procedure for finding the roots of  $K_1$ :

1. Search for a root  $t \geq U_{minj}$

– If  $t$  is such that  $D_x^- U_{ij}(t) \geq -\alpha \frac{\sqrt{1+M(t)^2}}{\sqrt{I^2 - \alpha^2}}$  then  $t$  is a root of

$$I(x) \sqrt{1 + D_x^- U_{ij}(t)^2 + M(t)^2} + \alpha D_x^- U_{ij}(t) - \gamma = 0.$$

– If  $t$  is such that  $D_x^- U_{ij}(t) \leq -\alpha \frac{\sqrt{1+M(t)^2}}{\sqrt{I^2 - \alpha^2}}$  then  $t$  is a root of

$$I(x) \sqrt{1 + \alpha^2 \frac{1 + M(t)^2}{I^2 - \alpha^2} + M(t)^2} - \alpha^2 \sqrt{\frac{1 + M(t)^2}{I^2 - \alpha^2}} - \gamma = 0.$$

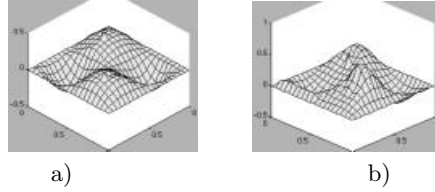
2. Search for a root  $t \leq U_{minj}$ . ( $M(t) = 0$  !) In that case  $t$  is such that  $D_x^- U_{ij}(t) \geq -\frac{\alpha}{\sqrt{I^2 - \alpha^2}}$  and therefore it is the root of

$$I(x) \sqrt{1 + D_x^- U_{ij}(t)^2} + \alpha D_x^- U_{ij}(t) - \gamma = 0.$$

Once we have found the roots  $t_i$ ,  $i = 1, 2$  de  $K_1$  we have to verify that  $K_2(t_i) \leq 0$ , otherwise we must find the roots  $u_i$ ,  $i = 1, 2$  of  $K_2$  such that  $K_1(u_j) \leq 0$ .

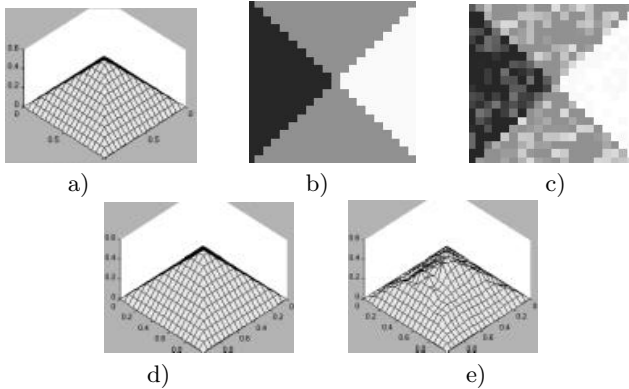
## 7 Experimental Results

We have tested our algorithm with synthetic images generated by shapes with several levels of regularity e.g.  $C^\infty$  (a sinusoid, see figures 2 and 3), or  $C^0$  (a pyramid, see figures 4 and 5), to demonstrate the ability of our method to deal with smooth and nonsmooth objects. We also have tested it with real images, an example is shown in figure 6. In all results, the parameters are  $n$ , the number



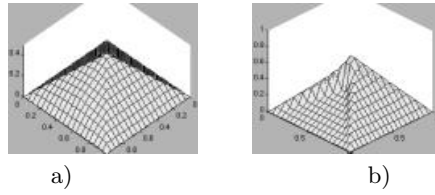
**Fig. 3.** Sinusoidal surface of figure 2 reconstructed with an error on the parameter  $\mathbf{L}$ : a)  $\theta_p = 33^\circ$ ,  $\epsilon_\theta = 5^\circ$ ,  $n = 37$ ,  $\epsilon_1 = 11.4\%$ ,  $\epsilon_2 = 7.6\%$ ,  $\epsilon_\infty = 17.9\%$ ; b)  $\theta_p = 18^\circ$ ,  $\epsilon_\theta = 10^\circ$ ,  $n = 45$ ,  $\epsilon_1 = 18.3\%$ ,  $\epsilon_2 = 13.8\%$ ,  $\epsilon_3 = 41.9\%$ .

of iterations,  $\epsilon_1$ ,  $\epsilon_2$  and  $\epsilon_\infty$  the relative errors of the computed surface measured according to the  $L_1$ ,  $L_2$  and  $L_\infty$  norms, respectively,  $\theta$  the angle of the direction of illumination with the  $z$ -axis.

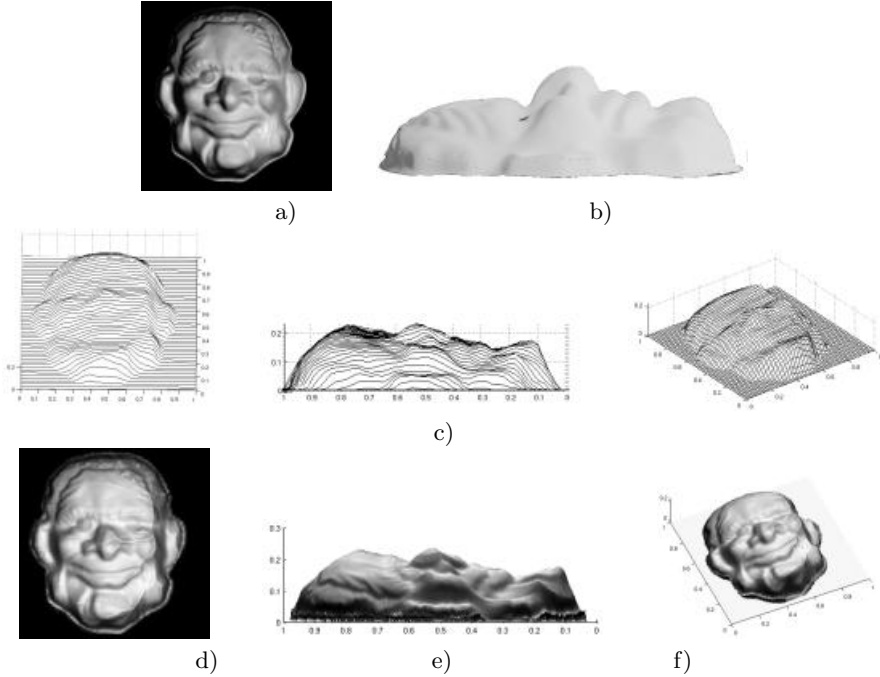


**Fig. 4.** Results for a synthetic image generated by a pyramidal surface sampled on a grid of size  $20 \times 20$  with  $\theta = 36^\circ$ : a) original surface, b) original image, c) noisy image; d) surface reconstructed from b):  $n = 89$ ,  $\epsilon_1 = 0.4\%$ ,  $\epsilon_2 = 0.4\%$ ,  $\epsilon_\infty = 0.8\%$ ; e) surface reconstructed from c):  $n = 91$ ,  $\epsilon_1 = 23.7\%$ ,  $\epsilon_2 = 23.2\%$ ,  $\epsilon_\infty = 26.3\%$ .

In all synthetic cases we show the original object, the input image and the reconstructed surface. We then demonstrate the stability of our method with respect to two types of errors. The first type is image intensity errors due to noise. Uniformly distributed noise has been added to some pixels of the input images and the corresponding reconstructed surfaces are shown (figures 2 and 4). The second type of error is due to an incorrect estimation of the direction of illumination  $\mathbf{L}$  (figures 3 and 5). We start with a smooth sinusoidal object, see



**Fig. 5.** Pyramidal surface of figure 4 reconstructed with an error on the parameter  $\mathbf{L}$ : a)  $\theta_p = 41^\circ$ ,  $\epsilon_\theta = 5^\circ$ ,  $n = 80$ ,  $\epsilon_1 = 16.5\%$ ,  $\epsilon_2 = 15.2\%$ ,  $\epsilon_3 = 19.8\%$ ; b)  $\theta_p = 26^\circ$ ,  $\epsilon_\theta = 10^\circ$ ,  $\epsilon_1 = 25.1\%$ ,  $\epsilon_2 = 14.0\%$ ,  $\epsilon_3 = 28.3\%$ .



**Fig. 6.** Experimental results with a real image: a) Original image of size 200x200, b) Lateral view (image) of the real object, c) Three views of the surface reconstructed from a), d) Synthetic image generated from the surface c), e) and f) reconstructed surface, textured with image d).

figure 2. We introduce an error  $5^\circ$  on the parameter  $\mathbf{L}$ , see figure 3. We note  $\theta_p$  the angle used for computation;  $\epsilon_\theta = |\theta - \theta_p|$ .

As seen from these figures, our algorithm seems to be quite robust, not only to intensity noise (see figures 2.e and 4.e), as in [24], but also to inaccuracies in the estimation of the direction of the light source  $\mathbf{L}$  (see figures 3 and 5). The pyramid example shows the remarkable ability of the numerical scheme to deal with functions which are only continuous. This example also shows the convergence of our algorithm with discontinuous images: through the recent

works of Ostrov [22,18], we hope to extend shortly our theory for discontinuous images.

The real image shown in figure 6.a and 6.b is a photograph of a Halloween mask taken with a standard camera with 70mm focal length. The light source was far from being pointlike and at infinity and ambient lighting due to reflections on the walls was present. The reflectance of the mask was not quite Lambertian (some highlights were visible). Unlike the case of the synthetic examples where the critical points  $x$  such that  $I(x) = 1$  were included in the boundary conditions (i.e. their distances were supposed to be known), these distances must in this case be computed by the algorithm, making the problem ill-posed. Despite this difficulty, the results shown in figure 6.c-f are of good quality.

## 8 Conclusion

We have proposed a rigorous mathematical analysis of the simplest version of the shape from shading problem in the case of a Lambertian object illuminated by a point source at infinity and imaged by an orthographic camera. We have given hypotheses for the existence and uniqueness of a (continuous) viscosity solution, provided an approximation scheme of this solution, an algorithm for computing the solution. Unlike in [19], we have proved the convergence of our numerical scheme. Our implementation of the algorithm described in [19] shows that it fails as soon as the direction of the source of lighting is a bit too far from being parallel to the  $z$ -axis, the original case studied in [24].

We are extending our analysis and algorithm to more general cases.

## References

1. M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser, 1997.
2. G. Barles. *Solutions de Viscosité des Equations de Hamilton-Jacobi*. Springer-Verlag, 1994.
3. M.J. Brooks, W. Chojnacki, and R. Kozera. Shading without shape. *Quarterly of Applied Mathematics*, L(1):27–38, 1992.
4. A.R. Bruss. The eikonal equation: Some results applicable to computer vision. *Journal of Mathematical Physics*, 23(5):890–896, May 1982.
5. F. Camilli and M. Falcone. An approximation scheme for the maximal solution of the shape-from-shading model. *International Conference on Image Processing*, pages 49–52, 1996.
6. M.G. Crandall. Viscosity solutions of Hamilton–Jacobi equations. In *Nonlinear Problems: Present and Future, Proc. 1st Los Alamos Conf., 1981*, volume 61, pages 117–125. North-Holland Math. Stud., 1982.
7. M.G. Crandall, H. Ishii, and P.L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Soc.*, 27:1–67, 1992.
8. M.G. Crandall and P.L. Lions. Viscosity solutions of Hamilton–Jacobi equations. *Trans. AMS*, 277:1–43, 1983.
9. Paul Dupuis and John Oliensis. An optimal control formulation and related numerical methods for a problem in shape reconstruction. *The Annals of Applied Probability*, 4(2):287–346, 1994.

10. J.-D. Durou and H. Maitre. On convergence in the methods of strat and smith for shape from shading. *The International Journal of Computer Vision*, 17(3):273–289, 1996.
11. J.-D. Durou and D. Piau. Ambiguous shape from shading with critical points. *Journal of Mathematical Imaging and Vision*, 12:99–108, 2000.
12. M. Falcone and M. Sagona. An algorithm for the global solution of the shape-from-shading model. *International Conference on Image Analysis and Processing*, 1:596–603, 1997. LNCS 1310.
13. Berthold K. Horn and Michael J. Brooks, editors. *Shape from Shading*. The MIT Press, 1989.
14. Berthold K.P. Horn. Height and Gradient from Shading. *The International Journal of Computer Vision*, 5(1):37–75, August 1990.
15. B.K.P. Horn. Obtaining shape from shading information. In P.H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.
16. B.K.P. Horn, R.S. Szeliski, and A.L. Yuille. Impossible shaded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):166–169, 1993.
17. K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence Journal*, 17:181–184, 1981.
18. J. Kain and D.N. Ostrov. Numerical shape-from-shading for discontinuous photographic images. *The International Journal of Computer Vision*, 44(3):163–173, 2001.
19. R. Kimmel and J.A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(2):237–244, May 2001.
20. P.L. Lions. *Generalized Solutions of Hamilton–Jacobi Equations*. Number 69 in Research Notes in Mathematics. Pitman Advanced Publishing Program., 1982.
21. P.L. Lions, E. Rouy, and A. Tourin. Shape-from-shading, viscosity solutions and edges. *Numer. Math.*, 64:323–353, 1993.
22. D.N. Ostrov. Extending viscosity solutions to eikonal equations with discontinuous spatial dependence. *Nonlinear Anal.*, 42(4):709–736, 2000.
23. E. Prados, O. Faugeras, and E. Rouy. Shape from shading and viscosity solutions. Technical report, INRIA, 2002.
24. E. Rouy and A. Tourin. A Viscosity Solutions Approach to Shape-from-Shading. *SIAM Journal of Numerical Analysis*, 29(3):867–884, June 1992.

# Model Acquisition by Registration of Multiple Acoustic Range Views

Andrea Fusiello, Umberto Castellani, Luca Ronchetti, and Vittorio Murino

Dipartimento di Informatica, Università di Verona  
Strada le Grazie 15, 37134 Verona, Italy  
{fusiello,castellani,murino}@sci.univr.it

**Abstract.** This paper deals with the three-dimensional reconstruction of an underwater environment from multiple acoustic range views acquired by a remotely operated vehicle. The problem is made challenging by the very noisy nature of the data, the low resolution and the narrow field of view of the sensor. Our contribution is twofold: first, we introduce a statistically sound thresholding (the X84 rejection rule) to improve ICP robustness against noise and non-overlapping data. Second, we propose a new global registration technique to distribute registration errors evenly across all views. Our approach does not use data points after the first pairwise registration, for it works only on the transformations. Therefore, it is fast and occupies only a small memory. Experimental results suggest that ICP with X84 performs better than Zhang's ICP, and that the global registration technique is effective in reducing and equalizing the error.

## 1 Introduction

In this paper we address the problem of registration of many three-dimensional (3D) point sets, coming from an acoustic range sensor. Typically, the term *registration* is used for the geometric alignment of a pair or more 3D data point sets, while the term *fusion* indicates the recovery of a single surface representation from registered 3D data sets.

The registration of two point sets is usually addressed by means of the Iterated Closest Point (ICP) algorithm [2,5]. ICP is based on the search of pairs of nearest points in the two sets, and estimating the rigid transformation which align them. Then, the rigid transformation is applied to the points of one set, and the procedure is iterated until convergence. ICP assumes that one point set is a subset of the other. When this assumption is not verified, false matches are created, that negatively influence the convergence of ICP to the solution. In order to overcome this problem, many variants to ICP have been proposed, including the use of closest points in the direction of the local surface normal [5], the use of thresholds to limit the maximum distance between points [25], reject- ing matching on the surface boundaries [23], Least Median of Squares estimation [22].

In this paper we propose to use the X84 outlier rejection rule [10] to discard false correspondences on the basis of their distance. This is an improvement over

[25], because the threshold is set automatically, and a larger basin of attraction is achieved.

A widely used approach to the registration of many views is to sequentially apply pairwise registration until all the views are combined. Chen and Medioni [5] proposed an incremental approach in which two views are registered and merged building a *metaview*. The next view is then registered and merged with the metaview and the process is repeated for all the views. This approach was taken also by [21] and [12]; the latter proposed to incrementally build a surface model, onto which new views can be registered and already registered views can be adjusted. In [4], couples of images are incrementally registered together with a final registration between the first and last view, by using the inverse calibration procedure of the range-finder to relate a point in the 3D space corresponding to a point in the range image.

These schemes do not use all the available information, and do not compute the optimal solution, because of the accumulation of registration errors, as pointed out by [19] and [1]. Multiview registration could exploit information present in the unused overlapping view pairs, distributing the registration error evenly between every pairwise registration. Bergevin et al. [1] registered multiple range images simultaneously, using an extended ICP algorithm. They converted the sequential registration relationship into a star-shaped relationship, and then imposed the *well-balanced network* constraint. A network of range views is well-balanced when the registration error is similar for all transformation matrices, and the transformation matrix between any two views is uniquely defined regardless of the path chosen to link the views. Pulli [18] proposed to use the pairwise alignments as constraints that the multiview step enforces while evenly diffusing the pairwise registration errors. In such a way, computational time is reduced as well as memory storage. He introduces the concept of *virtual mate* to enforce the pairwise alignments as constraints.

Some works focus on computing the global registration given the correspondences among many views (this is called the *N-view point set registration problem*). To this end, in [20] and [8], a force-based optimisation approach is proposed. Assuming the points' correspondences among the data sets are known, interconnections using springs between corresponding points is simulated. Pennec [17] introduces an iterative algorithm based on the concept of *mean shape*. Benjemaa and Shmitt [19] use a quaternion approach similar to [3]. In [9] a statistical model is introduced, and the problem is solved using the EM algorithm. In a recent work, Williams and Bennamoun [24] proposed a new technique for the simultaneous registration of multiple point sets, in which rotations are first computed iteratively, and then translations are obtained as the solution of a linear system. The method have been integrated in a generalized multiview ICP. Three of these techniques, namely [20],[17], and [19], have been compared in [6], and the result is that, not considering speed, Pennec's method is the best one, whereas [19] is the fastest.

All the multi-alignment methods need to keep data of all – or at least some – views in memory at the same time, reducing drastically performance, especially when aligning large data set. Our global registration approach differs from all the



others because we enforce the constraints arising from the pairwise registration directly on the transformation matrices, without the need to process data points again, after the initial pairwise registration between all the overlapping views. The idea comes from [7] where it was applied to the construction of planar mosaics from images. Here we propose to extend the technique to the registration of multiple 3D point sets. In our case we end up with a non-linear system of equations, because of the parametrization of the rotations. Following [16,3,19] we used quaternions to represent rotations, because of their well-known good properties [15].

In the field of 3D registration, the closest work to our approach is [18], because both are based on the simultaneous satisfaction of constraints provided by the pairwise registration, and both do not rely on the solution of the N-view point set registration problem. Our work differs in the formulation of the constraints (which do not use data points) and because we cope with overlapping between any number of sets, not just pairs.

Our final goal is the reconstruction of a scene from a sequence of cluttered, noisy, and low resolution 3D data, aimed at building a 3D model. Such model is then displayed to a human operator of an underwater remotely operated vehicle (ROV), in order to facilitate the navigation and the understanding of the surrounding environment. Our data come from a high frequency acoustic camera, called Echoscope [11] with a typical resolution of 3 cm. Moreover, speckle noise affects data due to the coherent nature of the acoustic signals. With such operative conditions, the registration problem is much more demanding, and, although a large literature addresses this issue, no work dealt with this particular kind of 3D data. Further, due to lack of tight hypotheses about the sensor motion and the direct manipulation of the transformation matrices instead of 3D points, our approach results particularly suitable to real-time applications.

The rest of the paper is structured as follows. In Section 2, the pairwise registration stage is described outlining the X84 outlier rejections rule. Section 3 introduces the multiview registration focusing on the global transformation adjustment. Results are shown in Section 4 for both synthetic and real images and conclusions are drawn in Section 5.

## 2 Robust Pairwise Registration

Pairwise registration was addressed using the classical Iterated Closest Point (ICP) algorithm [2] to which we added an outlier rejection rule (X84) [10] in order to cater for non-overlapping areas between views.

### 2.1 Two View Point Set Registration

Let us suppose that we have two sets of 3-D points which correspond to a single shape but are expressed in different reference frames. We will call one of these sets the model set  $X$ , and the other the data set  $Y$ . Assuming that for each point in the data set the corresponding point in the model set is known, the *point set registration problem* consist in finding a 3-D transformation which, when applied

to the data set  $Y$ , minimizes the distance between the two point sets. The goal of this problem can be stated more formally as follows:

$$\min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^N \|\mathbf{x}_i - (\mathbf{R}\mathbf{y}_i + \mathbf{t})\|^2, \quad (1)$$

where  $\mathbf{R}$  is a  $3 \times 3$  rotation matrix,  $\mathbf{t}$  is a  $3 \times 1$  translation vector, and the subscript  $i$  refers to corresponding elements of the sets  $X$  and  $Y$ . Efficient, non-iterative solutions to this problem were compared in [14], and the one based on Singular Value Decomposition (SVD) was found to be the best, in terms of accuracy and stability.

## 2.2 Iterated Closest Point

In general, when point correspondences are unknown, the Iterated Closest Point (ICP) algorithm may be used. For each point  $\mathbf{y}_i$  from the set  $Y$ , there exists at least one point on the surface of  $X$  which is closer to  $\mathbf{y}_i$  than all other points in  $X$ . This is the *closest point*,  $\mathbf{x}_i$ . The basic idea behind the ICP algorithm is that, under certain conditions, the point correspondence provided by sets of closest points is a reasonable approximation to the true point correspondence. The ICP algorithm can be summarized:

1. For each point in  $Y$ , compute the closest point in  $X$ ;
2. With the correspondence from step 1, compute the incremental transformation  $(\mathbf{R}, \mathbf{t})$ ;
3. Apply the incremental transformation from step 2 to the data  $Y$ ;
4. If the change in total mean square error is less than a threshold, terminate. Else goto step 1.

Besl and McKay [2] proved that this algorithm is guaranteed to converge monotonically to a local minimum of the Mean Square Error (MSE).

ICP can give very accurate results when a set is a subset of the other, but results deteriorate with outliers, created by non-overlapping areas between the two sets. In this case, the overlapping surface portions must start very close to each other to ensure convergence, making the initial position a critical parameter.

Modifications to the original ICP have been proposed to achieve accurate registration of *partially* overlapping point sets [25,23,22]. We implemented a variation similar to the one proposed by Zhang [25], using robust statistics to limit the maximum allowable distance between closest points.

## 2.3 Robust Outlier Rejection

As pointed out by Zhang, the distribution of the residuals for two fully overlapping sets approximates a Gaussian, when the registration is good. The non-overlapped points skew this distribution: they are *outliers*. Therefore, good correspondences can be discriminated by using an outlier rejection rule on the distribution of closest point distances.

To do this, we employ a simple but effective rejection rule, *X84* [10], which use robust estimates for location and scale of a corrupted Gaussian distribution to set a rejection threshold. The median is a robust location estimator, and the Median Absolute Deviation (MAD), defined as

$$\text{MAD} = \text{med}_i \{ |\epsilon_i - \text{med}_j \epsilon_j| \} \quad (2)$$

is a robust estimator of the scale (i.e., the spread of the distribution). It can be seen that, for symmetric (and moderately skewed) distributions, the MAD coincides with the *interquartile range*:

$$\text{MAD} = \frac{\xi_{3/4} - \xi_{1/4}}{2}, \quad (3)$$

where  $\xi_q$  is the  $q$ th quantile of the distribution (for example, the median is  $\xi_{1/2}$ ). For normal distributions we infer the standard deviation from

$$\text{MAD} = \Phi^{-1}(3/4)\sigma \approx 0.6745\sigma. \quad (4)$$

The *X84* rule prescribes to reject values that are more than  $k$  Median Absolute Deviations away from the median. A value of  $k=5.2$ , under the hypothesis of Gaussian distribution, is adequate in practice (as reported in [10]), since it corresponds to about 3.5 standard deviations, and the range  $[\mu - 3.5\sigma, \mu + 3.5\sigma]$  contains more than the 99.9% of a Gaussian distribution. The rejection rule *X84* has a *breakdown point* of 50%: any majority of the data can overrule any minority.

### 3 Multiview Registration

We now turn our attention to the simultaneous registration of several point sets.

#### 3.1 Chaining Pairwise Transformations

Assume that there are  $M$  overlapping point sets (or views)  $V^1 \dots V^M$ , each taken from a different viewpoint. The objective is to find the best rigid transformations  $\mathbf{G}^1 \dots \mathbf{G}^M$  to apply to each set, bringing them a common reference frame where they are seamless aligned.

Let  $\mathbf{G}^{ij}$  be the rigid transformation matrix (in homogeneous coordinates) that registers view  $j$  onto view  $i$ , i.e.,

$$V^i = \mathbf{G}^{ij} V^j \quad (5)$$

where the equality holds only for the overlapping portions of the two points sets  $V^i$  and  $\mathbf{G}^{ij} V^j$ .

If we choose (arbitrarily) view  $k$  as the reference one, then the unknown rigid transformation  $\mathbf{G}^1 \dots \mathbf{G}^M$  are respectively  $\mathbf{G}^{k,1} \dots \mathbf{G}^{k,M}$ . As customary, we will take  $k = 1$ .

These rigid transformations are not independent each other, being linked by the composition relationship:

$$\mathbf{G}^{k,j} = \mathbf{G}^{k,i} \mathbf{G}^{i,j}. \quad (6)$$

We can therefore estimate the alignment  $\mathbf{G}^j$  of image  $V^j$  on the reference view (defined by the image  $V^1$ ), by first registering  $V^j$  onto any view  $V^i$  and then using  $\mathbf{G}^i$  to map the result into the space of  $V^1$

$$\mathbf{G}^j = \mathbf{G}^i \mathbf{G}^{i,j} \quad (7)$$

This relationship, can be used to compute  $\mathbf{G}^i$  when all the matrices  $\mathbf{G}^{i-1,i} \dots \mathbf{G}^{1,2}$  are known, by simply chaining them

$$\mathbf{G}^i = \prod_{j=2}^i \mathbf{G}^{j-1,j} \quad (8)$$

The global registration matrix  $\mathbf{G}^i$  will map  $V^i$  into the space of  $V^1$  (the reference view).

As it is well known, the combination of pairwise registration does not yield the optimal result. For example, if  $\mathbf{G}^{k,i}$  and  $\mathbf{G}^{i,j}$  are optimal on the sense that they minimize the mean square error distance between the respective sets, then  $\mathbf{G}^{k,j}$  computed with Eq. (6) does not necessarily minimizes the mean square error between views  $V^j$  and  $V^k$ . Small registration errors accumulate so that images near the end of a sequence have a large cumulative error.

### 3.2 Global Transformations Adjustment

In order to improve the quality of global registration, let us suppose we have locally registered all spatially overlapping image pairs, in addition to those that are adjacent in the image sequence. Especially for underwater images, in which the ROV moves back and forth we can find significant overlapping also between distant views in the temporal sequence.

The aim of our method is to optimize the information coming from every pairwise registrations, obtained by the alignment of all overlapped range images. The original contribution consists in obtaining a global registration by introducing algebraic constraints on the transformations, instead of data points.

We first perform pairwise registration between every view and each of its overlapping views, thereby computing the  $\mathbf{G}^{i,j}$  whenever it is possible. By considering many equations as (7), we can build a system of equations in which the  $\mathbf{G}^{i,j}$  are known quantities obtained by pairwise image registration, and the matrices  $\mathbf{G}^{i1} = \mathbf{G}^i$  ( $1 \leq i \leq N$ ) are unknowns to be found. By decomposing the homogeneous transformation matrices  $\mathbf{G}$  into a rotation and translation, Eq. (7) splits in two:

$$\begin{cases} \mathbf{R}^j = \mathbf{R}^i \mathbf{R}^{i,j} \\ \mathbf{t}^j = \mathbf{R}^i \mathbf{t}^{i,j} + \mathbf{t}^i \end{cases} \quad (9)$$

where  $\mathbf{R}$  is a rotation matrix and  $\mathbf{t}$  is a translation vector. Although this system of equations is essentially linear, a number of problems arise when formulating solutions that account for the non-linear constraints on the components of  $\mathbf{R}$ . In order to respect these constraints, the rotation matrices must be suitably parametrized, ending up with a system of non-linear equations.

This non-linear least squares problem can be cast as the minimization of the following objective function:

$$\min_{i,j} \sum \left( \left( \frac{\text{angle}(\mathbf{R}^i \mathbf{R}^{i,j} (\mathbf{R}^j)^\top)}{\pi} \right)^2 + \left( \frac{\|\mathbf{R}^i \mathbf{t}^{i,j} + \mathbf{t}^i - \mathbf{t}^j\|}{\|\mathbf{t}^i\|} \right)^2 \right) \quad (10)$$

where  $\text{angle}(\cdot)$  takes a rotation matrix and returns the angle of rotation around a suitable axis. Starting from the global registration obtained by chaining pairwise transformation (Eq. (8)), a solution is found using a Quasi-Newton method.

The estimated transformation  $\mathbf{G}^1 \dots \mathbf{G}^M$  are influenced by all the pairwise observed transformations, and the registration error is distributed over all the estimated transformations. In this sense, the network of the views after the final registration is very close to be *well balanced* as defined in [1].

Moreover, the complexity of the proposed algorithm is *independent from the number of points involved*. Because the objective function includes only the matrix components, the complexity depends only on the number of (overlapping) views.

### 3.3 Dealing With Rotations

A number of techniques have been developed to represent rotations. One of the most convenient is the quaternions representation. They have a number of mathematical properties that make them particularly well suited to requirements of iterative gradient-based search for rotation and translation [15].

Rotations are represented by unit quaternions. Instead of requiring the quaternion  $\mathbf{q} = [u, v, w, s]$  to be a unit vector, we can enforce the constraint that the rotation matrix is orthogonal by dividing the matrix by the squared length of the quaternion [15]:

$$\mathbf{R}(\mathbf{q}) = \frac{1}{\mathbf{q} \cdot \mathbf{q}} \mathbf{R}_u(\mathbf{q}) \quad (11)$$

where  $\mathbf{R}_u(\mathbf{q})$  is the rotation matrix given by

$$\mathbf{R}_u(\mathbf{q}) = \begin{pmatrix} s^2 + u^2 - v^2 - w^2 & 2(uv - sw) & 2(uw + sv) \\ 2(uv + sw) & s^2 - u^2 + v^2 - w^2 & 2(vw - su) \\ 2(uw - sv) & 2(vw + su) & s^2 - u^2 - v^2 + w^2 \end{pmatrix} \quad (12)$$

This constraint is necessary in general to ensure the gradient accurately reflect the differential properties of a change in the quaternion parameters.

### 3.4 Summary of the Algorithm

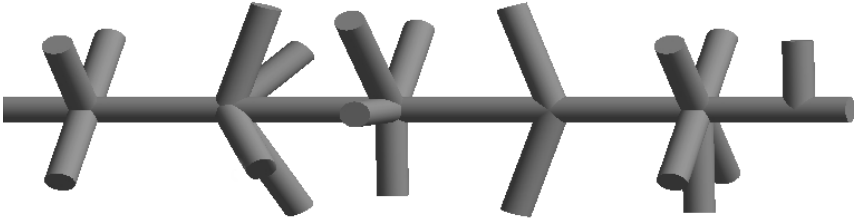
Finally, the multiview registration algorithm is defined as:

- Step 1.** calculate pairwise registration for all the overlapped images using ICP algorithm, with X84 rejection rule;
- Step 2.** if the pairwise registration is good, accept the registration matrix  $\mathbf{G}^{i,j}$ ;
- Step 3.** compute a starting guess for the global registration by chaining pairwise transformation (Eq. (8));
- Step 4.** minimize the objective function defined in Eq. (10) with the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure<sup>1</sup>; At each step enforce orthogonality of rotation matrix with Eq. (11)
- Step 5.** apply the transform defined by  $\mathbf{G}^i$  to the view  $V^i$ ,  $i = 1, \dots, M$ .

Registered sets of points must be fused in order to get a single 3D model. Surface reconstruction from multiple range images can be addressed as the problem of surface reconstruction from a set of unorganized 3D points, disregarding the original organization of the data. We used the public domain implementation of Hoppe and De Rose algorithm [13].

## 4 Results Description

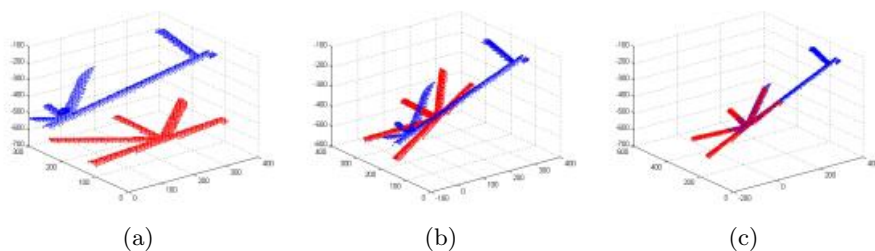
In this section we show results obtained on synthetic and real acoustic images.



**Fig. 1.** 3D synthetic model representing pipes structure

In synthetic experiments we simulated the movement of an underwater ROV around the external part of an offshore rig using the OpenGL library to generate synthetic range images. Given a 3D model of part of the rig (Fig. 1),

<sup>1</sup> MATLAB `lsqnonlin` function



**Fig. 2.** In Figure (a) the two point sets are in the start position, Figure (b) shows the result of Zhang's ICP algorithm and Figure (c) shows the result of ICP with X84

range images are obtained by moving a (virtual) camera and extracting the *z-buffer* for each view. In order to assess the final registration, we made the last view to coincide with the first one.

In real experiments the acoustic data are acquired by an underwater ROV using the Echoscope acoustic camera, which outputs a  $64 \times 64$  range image [11]. The noise corrupts sensibly the acoustic signals and decreases the reliability of the estimated 3D measures. Moreover, there is a trade off between range resolution and field of view. Resolution depends on the frequency of the acoustic signal (it is about 3 cm at 500 KHz): the higher the frequency, the higher the resolution, the narrower field of view. Consequently, we are forced to operate with a limited field of view.

#### 4.1 Pairwise Registration

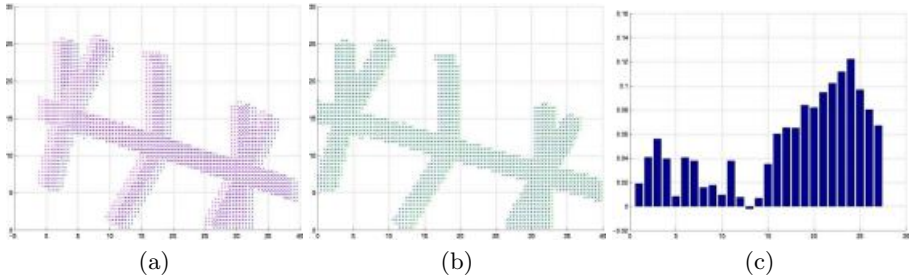
In Fig. 2, we show an example of two point sets that Zhang's ICP fails to align. Instead, our ICP algorithm with X84 rejection rule recovers the correct rigid transformation.

#### 4.2 Global Registration

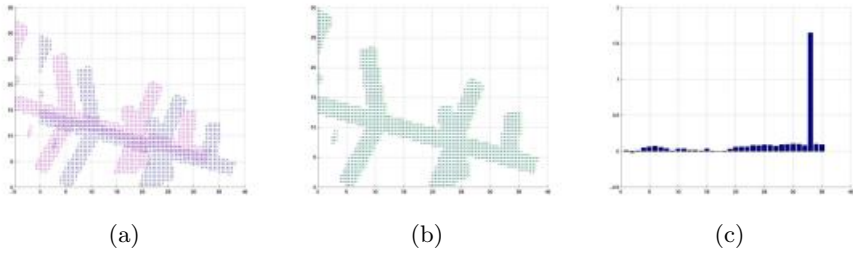
In order to evaluate the performance of our global registration algorithm, we computed the *registration error* as the closest points mean square distance between each view and the mosaic composed by all the already registered views (outliers are discarded according to the X84 rule). The improvement over the chained pairwise alignment is shown as an histogram depicting, for each view, the difference between the registration errors of the two techniques (a positive value means an improvement of our method).

In experiment n. 1 we generated a synthetic sequence, of 29 range images. The benefit brought by the global registration can be appreciated in Fig. 3a-b. The histogram in Fig. 3c shows that the global registration improves especially in near the end of the sequence (as expected).

In experiment n. 2 we generated a sequence composed by 37 range images. We wanted to test the performance of the global registration algorithm in the



**Fig. 3.** Experiment n. 1. Alignment between view n. 1 and view n. 29 for chained pairwise registration (a) and global registration (b). Histogram of the differences of the registration error (c)



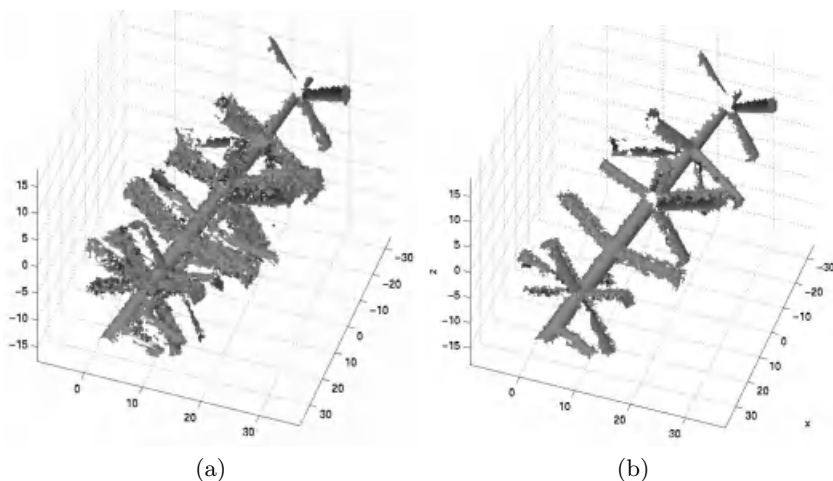
**Fig. 4.** Experiment n. 2. Alignment between view n. 1 and view n. 37 for chained pairwise registration (a) and global registration (b). Histogram of the differences of the registration error (c)

presence of an incorrect pairwise registration (n. 35). In this case the chaining of pairwise transformations inevitably propagates the error. In our global multiview registration, thanks to the information coming from the other pairwise transformations linking (indirectly) view n. 35 and n. 34, the correct registration is achieved, and the error is distributed over the whole sequence. Fig. 4c shows the improvement obtained by optimal global registration, which is concentrated on matrix n. 35, as expected. The improvement brought by the global registration is also clearly visible in the Fig. 4a-b and also in Fig. 5 where the reconstructed surfaces are shown for both techniques.

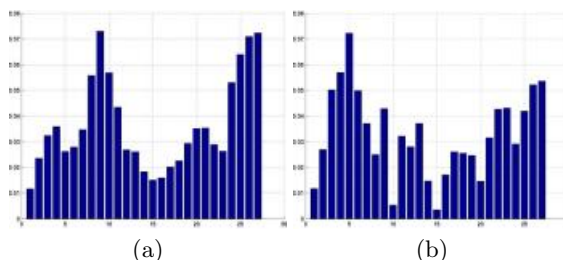
In experiment n. 3 and n. 4 we introduced Gaussian white additive noise with different standard deviation ( $\sigma = 0.02$  and  $\sigma = 0.045$ , respectively) on the synthetic images of experiment n. 1. The relative histograms are shown in Fig. 6.

Experiment n. 5 uses a sequence of 15 real acoustic images (Fig. 7) that are partial views of a tubular structure. The images are rather noisy and the scene is quite difficult to understand from a single view. The histogram shown in Fig. 8c confirms the improvement obtained by the global multiview registration.





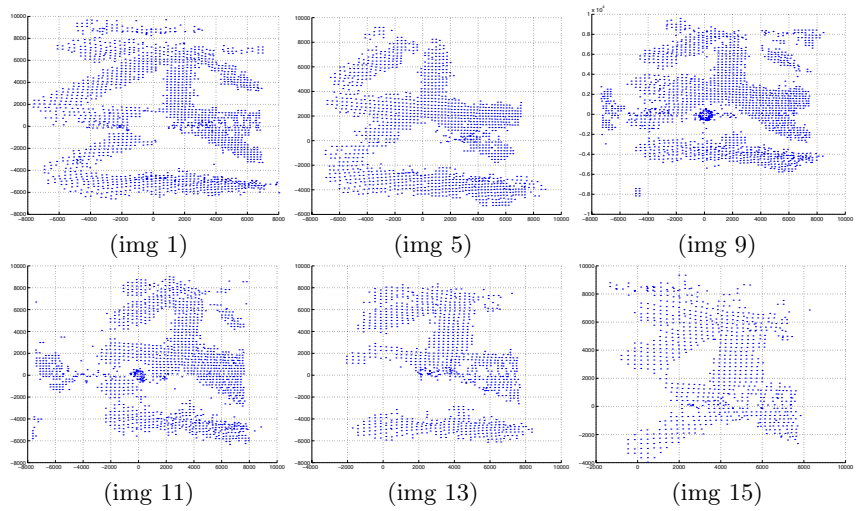
**Fig. 5.** Surface reconstruction using Hoppe and De Rose algorithm. Chained pairwise registration (a) and global multiview registration (b)



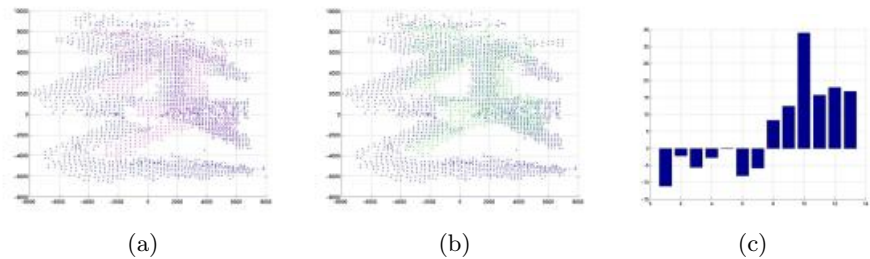
**Fig. 6.** Histogram of the differences between the registration error for the chained pairwise registration and the global registration, in experiment n. 3 (a) and n. 4 (b). A positive value correspond to an improvement over chained pairwise registration.

The light worsening of the first views registration is compensated by the good improvement of the last ones. A more accurate evaluation is not possible in real case because true correspondences are not known (as in synthetic experiments). Nevertheless, Fig. 8a-b visually confirm the effectiveness of the global multiview registration in building the mosaic of the scene.

Tables 1 and 2 summarizes the numerical results obtained in all the experiments. Table 1 reports the average (over the views) registration errors for both algorithms. In Table 2 a more meaningful evaluation is obtained by calculating the registration error (misalignment) between the first and the last view (which should coincide). In this case we know exactly the corresponding points and error evaluation is more accurate.



**Fig. 7.** Experiment n. 5. Selected views from the real sequence.



**Fig. 8.** Experiment n. 5. First and last views of the mosaic. The images are aligned using chained pairwise registration algorithm (a) and global registration algorithm (b). Histogram of the differences of the registration error (c).

**Table 1.** Average registration error (cm).

Experiment	Chained pairwise reg.	Global registration	% difference
exp 1	0.24095	0.19258	20,0740 %
exp 2	0.28960	0.19630	32,2168 %
exp 3	0.36328	0.32936	9,3371 %
exp 4	0.50290	0.47200	6,1443 %
exp 5	15.47955	15.01574	2,9963 %

Our global multiview registration algorithm always improves over pairwise registration. When the noise level was increased in the experiments on synthetic

**Table 2.** Misalignment between the last and the first view (cm).

Experiment	Chained pairwise reg.	Global registration	% difference
exp 1	1.9584	0.1340	93,1577 %
exp 2	29.8662	0.1362	99,5440 %
exp 3	4.0279	2.1979	45,4331 %
exp 4	13.8420	11.1094	19,7413 %

data, our algorithm continued to perform better. Improvement was also seen in the experiment involving real data.

In another set of experiments we reproduced those reported by [19] and [24]. We evaluate the error on the rotations by comparing the result of the global registration with the ground truth. As a measure of the discrepancy between two rotations  $\mathbf{R}^1$  and  $\mathbf{R}^2$  we take  $\text{angle}(\mathbf{R}^1(\mathbf{R}^2)^\top)$ .

A set of six random transformations  $\mathbf{G}^{2,1}, \mathbf{G}^{3,1}, \mathbf{G}^{4,1}, \mathbf{G}^{5,1}, \mathbf{G}^{6,1}$  are generated, as in [24]. By composing these transformations we simulate the output of the pairwise registration of 12 overlapping views, namely 1-2, 2-3, 3-4, 4-1, 1-5, 2-5, 3-5, 4-5, 1-6, 2-6, 3-6, 4-6. The rotation components are perturbed by a random rotation in the range  $[-5,5]$  deg., The pairwise transformations are composed to obtain the perturbed  $\mathbf{G}^{i,1}$ , and the error on the rotations is computed as detailed above. Then we run our algorithm, and compare the average error and its variance over the views. Results, shown in Table 3, demonstrate that our global multiview registration technique is effective in reducing the variance of the error on the rotations, and also it improves the average error.

**Table 3.** Rotation error.

	Chained pairwise reg.	Global registration	% difference
average error	0.0463	0.0381	17.7%
average error variance	0.00243	0.00108	55.6%

## 5 Conclusions

In this paper we propose a technique for 3D object reconstruction from multiple acoustic range views, acquired by an underwater ROV. As data coming from the acoustic sensor is noisy, low resolution, and the field of view is narrow, we want to provide the human operator(s) with a synthetic 3D model of the scene, in order to facilitate the navigation and the understanding of the surrounding environment. To this end, we address the problem of registering many 3D views, starting from pairwise registration between all the overlapping views.

Our contribution is twofold. First we modified Zhang's ICP by introducing the X84 rejection rule, which does not depend on user specified thresholds and is more effective in achieving a larger convergence basin.

Moreover, we propose a new global multiview registration technique to distribute registration errors evenly across all views. Our approach differs from all the others because we enforce the constraints arising from the pairwise registration directly on the transformations, and we do not rely on the solution of the N-view point set registration problem. The complexity of our technique does not depend on the number of points involved, but only on the number of views. The drawback is that the error is only spread among the views, but does not get reduced significantly. Consequently, this technique is well suited for all the application where speed can be traded for accuracy.

Future work will be aimed at automatically detecting the degree of overlap between views and introducing a weight for each term of Equation (10), depending on the amount of overlap.

**Acknowledgment.** This work was supported by the European Commission under the project no. GRD1-2000-25409 ARROV (Augmented Reality for Remotely Operated Vehicles based on 3D acoustical and optical sensors for underwater inspection and survey). Echoscope images are courtesy of Dr. R.K. Hansen of Omnitech A/S (Norway).

## References

1. R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, May 1996.
2. P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
3. B.K.P.Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am.*, 4(4):629–642, 1987.
4. G. Blais and M. D. Levine. Registering multiview range data to create 3-D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.
5. Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.
6. S.J. Cunningham and A. J. Stoddart. N-view point set registration: A comparison. In *British Machine Vision Conference*, Nottingham, UK, 1999.
7. James Davis. Mosaics of scenes with moving objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 354–360, 1998.
8. A.W.Fitzgibbon D.W.Eggert and R.B.Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Computer Vision and Image Understanding*, (69):253–272, March 1998.
9. J. Goldberg. Registration of multiple point sets using the EM algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, Corfu, Greece, September 21-24 1999.

10. F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1986.
11. R. K. Hansen and P. A. Andersen. A 3-D underwater acoustic camera - properties and applications. In P.Tortoli and L.Masotti, editors, *Acoustical Imaging*, pages 607–611. Plenum Press, 1996.
12. H.Hoppe J.A.McDonald K.Pulli H.Jin, T.Duchamp and W.Stuetzle. Surface reconstruction from misregistered data. In *Proc. SPIE: Vision Geometry IV*, pages 324–328, 1995.
13. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, July 1992.
14. A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-D rigid transformations. *Machine Vision and Applications*, 9:272–290, 1997.
15. K.Ikeuchi M.D.Wheeler. Iterative estimation of rotation and translation using quaternion. Technical Report CMU-SC-95-215, Carnegie Mellon University, 1995.
16. O.D.Fougeras and M.Hebert. The representation, recognition, and locating fo 3-d objects. *International Journal Robot. Research*, 5(3):27–52, 1986.
17. X. Pennec. Multiple registration and mean rigid shape: Applications to the 3D case. In *16th Leeds Annual Statistical Workshop*, pages 178–185, Leed, UK, 1996.
18. K. Pulli. Multiview registration for large data sets. In *IEEE Int. Conf. on 3-D Imaging and Modeling, 3DIM '99*, pages 160–168, Ottawa (Canada), 1999.
19. R.Benjema and F.Shmitt. A solution for the registration of multiple 3D point sets using unit quaternions. In *Proceedings of the European Conference on Computer Vision*, pages 34–50, Freiburg, Germany, 1998.
20. A.J. Stoddart and A. Hilton. Registration of multiple point sets. In *Int. Conf. on Pattern Recognition*, pages 40–44, Vienna, 1996.
21. T.Masuda. Generation of geometric model by registration and integration of multiple range images. In *IEEE Int. Conf. on 3-D Imaging and Modeling, 3DIM 2001*, pages 254–261, Quebec City (Canada), 2001.
22. E. Trucco, A. Fusiello, and V. Roberto. Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters*, 20(9):889–898, September 1999.
23. Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 311–318. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
24. J. Williams and M. Bennamoun. Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–141, January 2001.
25. Z. Zhang. Iterative point matching of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

# General Trajectory Triangulation<sup>\*</sup>

Jeremy Yirmeyahu Kaminski and Mina Teicher

Bar-Ilan University  
Department of Mathematics and Statistics  
Ramat-Gan, Israel.  
{kaminsj,teicher}@math.biu.ac.il

**Abstract.** The multiple view geometry of static scenes is now well understood. Recently attention was turned to dynamic scenes where scene points may move while the cameras move. The triangulation of linear trajectories is now well handled. The case of quadratic trajectories also received some attention.

We present a complete generalization and address the Problem of general trajectory triangulation of moving points from non-synchronized cameras. Our method is based on a particular representation of curves (trajectories) where a curve is represented by a family of hypersurfaces in the projective space  $\mathbb{P}^5$ . This representation is linear, even for highly non-linear trajectories. We show how this representation allows the recovery of the trajectory of a moving point from non-synchronized sequences. We show how this representation can be converted into a more standard representation. We also show how one can extract directly from this representation the positions of the moving point at each time instant an image was made. Experiments on synthetic data and on real images demonstrate the feasibility of our approach.

**Keywords:** Structure from motion.

## 1 Introduction

The theory and practice of multiple view geometry is well understood when the scene consists of static point and line features. A summary of the past decade of work in this area can be found in [14,9]. However recently a new body of research has appeared which considers configurations of independently moving points, first with the pioneering work [1], which introduced the first algorithm for trajectory triangulation, and then with [6,11,12,17,19,20,21,22]. A common assumption of these works is that the motion must occur along a straight line or a conic section. When the motion is linear and at constant velocity, the recovery of the trajectory is done linearly. However for quadratic trajectories, the computations are non linear.

---

<sup>\*</sup> This work is partially supported by the Emmy Noether Institute for Mathematics and the Minerva Foundation of Germany, by the Excellency Center of the Israel Science Foundation "Group Theoretic Methods in the Study of Algebraic Varieties" and by EAGER (European network in Algebraic Geometry).

In order to deal with more general situations, we introduce a new representation of curves embedded in  $\mathbb{P}^3$ , which will be used to model arbitrary trajectories of moving points. This new representation, which was very briefly introduced in [16] in another context, has the great advantage that when considering the important question of trajectory triangulation from a weakly-calibrated sequence, the equations are linear in the unknowns, even when dealing with non-linear curves. More precisely we address the following problem:

*Consider a 3D point either static or moving along an arbitrary trajectory. The motion is seen by either static or moving non-synchronized cameras whose motions can be general but known. The problem is to give an algebraic representation of the motion of the point, where the unknowns appear linearly. Then we use this representation to recover either the positions of the point at each time instant an image was taken or to recover a more standard representation of the whole trajectory.*

Experiments on both synthetic and real data are provided demonstrating the validity of our approach.

We now proceed to introduce this special representation of curves.

## 2 Curve Representation

The trajectories are handled as algebraic curves. From a practical point of view, one might be interested in giving a piece-wise algebraic approximation of the point motion. Algebraic curves admit several representations. In order to appreciate the compactness and the efficiency of the new representation that we will introduce, let us consider the problem. Given a set of known cameras, we want to recover the trajectory of a moving point. We need to find an appropriate representation for that task.

We will start with the classical approach. An algebraic curve embedded in  $\mathbb{P}^3$  is the intersection of at least two surfaces. Therefore there exists a family of homogeneous polynomials  $\{F_i\}_{i=1,\dots,n}$ , where  $n \geq 2$ , such as a point  $\mathbf{P}$  lies on the curve if and only if any polynomial of the family vanishes on it.

This representation suffers from some drawbacks. The number of polynomials  $n$  is not necessarily 2. It is an open question whether any algebraic curve in  $\mathbb{P}^3$  can be described as the intersection of exactly two algebraic surfaces. Moreover the degree of each polynomial in the family is not known. Indeed it is clear that any polynomial combination  $\sum_i A_i F_i$  ( $A_i$  are polynomials) vanishes on the curve too. Thus several families define the same curve. Therefore the question of knowing the degree of each  $F_i$  is not an easy one.

This shows that this standard representation is not appropriate to our purpose. Moreover, our measurements are made from pixel locations over image planes. From these measurements the only available 3D information is made by the optical rays they generate. Thus we would like a representation of curves embedded in  $\mathbb{P}^3$  which satisfies the following conditions:

- The representation is in terms the lines meeting the curve and not the curve itself.

- The representation determines the curve entirely and uniquely.
- The recovery of the standard representation of the curve as the zero set of a system of equations should be very simple.
- The computations of the representation given a sufficiently general distribution of lines intersecting the curve is linear.

It turns out that such a representation exists.

However we shall now proceed to a more formal discussion. We will need the following definitions:

**Definition 1** A point  $\mathbf{P}$  on the curve, defined by  $\{F_i\}$ , is said to be singular if the gradient of  $F_i$  at  $\mathbf{P}$  vanishes,  $\text{grad}_{\mathbf{P}}(F_i) = 0$ , for all  $i$ . Otherwise it is called a regular point. A curve is said to be smooth if it has only regular points.

The foundations of algebraic geometry can be found in [10,13,7,8,15]. In our context, we will need to refer to the concept of the *irreducible component*, which is mathematically a maximum closed subset of an algebraic variety. It is enough for us to refer to the intuitive concept of the irreducible component. A component of a curve is a *subcurve*, a closed piece of it.

**Definition 2** A curve is said to be irreducible if it contains only one component.

**Definition 3** The degree of a curve embedded in  $\mathbb{P}^3$  is the number of points of intersection of the curve with a generic plane.

All our computations are done in the field of complex numbers, which is algebraically closed. However we are only interested in the real locus of the involved varieties.

## 2.1 Curve Representation in $\mathbb{G}(1, 3)$

Now we turn to present the curve representation that fits our requirements. We will represent a curve embedded in  $\mathbb{P}^3$  by a hypersurface in  $\mathbb{P}^5$ . This is related to the fact that a line in  $\mathbb{P}^3$  can be represented by its Plücker coordinates as a point of  $\mathbb{P}^5$ . If  $[L_1, L_2, L_3, L_4, L_5, L_6]$  are the Plücker coordinates of a line, then the following relation must hold:

$$L_1L_6 - L_2L_5 + L_3L_4 = 0. \quad (1)$$

Furthermore the coordinates of a point in  $\mathbb{P}^5$  that satisfy this equation are the Plücker coordinates of some line embedded in  $\mathbb{P}^3$  [2,9,13]. This shows that the set of lines in  $\mathbb{P}^3$  is isomorphic to a quadric surface of  $\mathbb{P}^5$  called the *Grassmannian* of lines of  $\mathbb{P}^3$ , denoted by  $\mathbb{G}(1, 3)$  and defined by equation 1. Therefore we shall denote by  $\mathbf{L}$  a line in  $\mathbb{P}^3$  and by  $\hat{\mathbf{L}}$  its Plücker coordinates which makes it a point of  $\mathbb{P}^5$ . A point in  $\mathbb{P}^3$  also has a representation in  $\mathbb{P}^5$  via the set of lines passing through it. Three lines passing through a point are sufficient and necessary to linearly span the space of all the lines through that point. Therefore a point in  $\mathbb{P}^3$  correspond to a plane included in the Grassmannian  $\mathbb{G}(1, 3)$ .

Now we proceed to show that a curve in  $\mathbb{P}^3$  can be represented as a subvariety of  $\mathbb{G}(1, 3)$ . A detailed treatment can be found in [13].





**Fig. 1.** A curve in  $\mathbb{P}^3$  with some lines cutting it. A point is on the curve if any line passing through it cuts the curve.

**Construction.** A smooth irreducible curve  $X$  of degree  $d$  and embedded in  $\mathbb{P}^3$  is entirely determined by the set of lines meeting it. See figure 1. We define the following set of lines:

$$\Lambda = \{\mathbf{L} \subset \mathbb{P}^3 \mid \mathbf{L} \cap X \neq \emptyset\}$$

The following facts are well known [13]:

1.  $\Lambda$  is an irreducible subvariety of  $\mathbb{G}(1, 3)$ .
2. There exists a homogeneous polynomial  $\Gamma$ , of degree  $d$ , such that:

$$Z(\Gamma) \cap \mathbb{G}(1, 3) = \Lambda,$$

where  $Z(\Gamma) = \{\hat{\mathbf{L}} \in \mathbb{P}^5 \mid \Gamma(\hat{\mathbf{L}}) = 0\}$ .

3.  $\Gamma$  is defined modulo the  $d$ -th graded piece of the ideal defining  $\mathbb{G}(1, 3)$ , i.e. modulo  $I(\mathbb{G}(1, 3))_d$ .
4. The dimension of the space of all possible  $\Gamma$  for a given curve, i.e. the dimension of the  $d$ -th graded piece of the homogeneous coordinate ring of  $\mathbb{G}(1, 3)$ ,  $S(\mathbb{G}(1, 3))_d$ , is for  $d \geq 2$ :

$$N_d = \binom{d+5}{d} - \binom{d-2+5}{d-2}.$$

5. It is sufficient to pick  $N_d - 1$  generic points on  $\Lambda$  to find  $\Gamma$  modulo  $I(\mathbb{G}(1, 3))_d$ . Every such point  $\hat{\mathbf{L}}$  yields one linear equation on  $\Gamma$ :

$$\Gamma(\hat{\mathbf{L}}) = 0, \tag{2}$$

and  $\Gamma$  is also defined modulo multiplication by a non-zero scalar.

If we put this high-flown terminology down to earth, it says this. The subvariety  $\Lambda$  is the intersection of  $\mathbb{G}(1, 3)$  with some hypersurface defined by a polynomial  $\Gamma$ . There exists a whole family of hypersurfaces intersect in  $\mathbb{G}(1, 3)$  exactly over  $\Lambda$ . This means that the polynomial  $\Gamma$  is not uniquely defined and is

computable only modulo the equation 1 defining  $\mathbb{G}(1, 3)$ . That is two polynomials  $F_1$  and  $F_2$  such that  $\Lambda = Z(\Gamma_1) \cap \mathbb{G}(1, 3) = Z(\Gamma_2) \cap \mathbb{G}(1, 3)$  satisfy an equation as follows:

$$F_2(L_1, \dots, L_6) = F_1(L_1, \dots, L_6) + R(L_1, \dots, L_6)(L_1L_6 - L_2L_5 + L_3L_4),$$

for some polynomial  $R$ . In order to recover the original curve  $X$  in  $\mathbb{P}^3$  it is sufficient to compute one of the polynomials  $\Gamma$ . This justifies the following definition.

**Definition 4** *Any element of the equivalence class of  $\Gamma$  is said to be the Chow polynomial of the curve  $X$ .*

Moreover, it is sufficient to find  $N_d$  linearly independent equations to find the Chow polynomial of  $X$ .

## 2.2 Working with the Chow Polynomial

The previous properties provide us with a way to compute  $\Gamma$  from a set of discrete measurements extracted from image sequences. Each image point generates with the camera center an optical ray  $\mathbf{L}$  with yields one linear constraint on  $\Gamma$  by equation 2.

Now we proceed to introduce two original algorithms which show in a very concrete way the equivalence between the standard representation of  $X$  as the intersection of surfaces in  $\mathbb{P}^3$  and the Chow polynomial.

Let  $\{F_i\}_i$  be a family of polynomials defining a curve in  $\mathbb{P}^3$ . The computation of  $\Gamma$  is done as follows:

*Eliminate  $X, Y, Z$  from the following system:*

$$\begin{aligned} F_i(X, Y, Z, 1) &= 0 \\ \widehat{\mathbf{L}} \vee \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} &= 0, \end{aligned}$$

where  $\widehat{\mathbf{L}} = [L_1, L_2, L_3, L_4, L_5, L_6] \in \mathbb{P}^5$  represents a line meeting the curve and  $\vee$  is the join operator (see [2,9]). The join of  $\widehat{\mathbf{L}}$  and the point  $\mathbf{P} = [X, Y, Z, 1]^T$  vanishes to express the fact that  $\mathbf{P}$  lies on the line represented by  $\widehat{\mathbf{L}}$ . When given  $\Gamma$  as a result of the previous elimination, we shall compute its normal form [3, 5] (to get a canonical representation of  $\Gamma$ ) with respect to equation 1 defining  $\mathbb{G}(1, 3)$ .

We give an example of this computation done with Maple. Let  $X$  be a conic defined by  $X+Y+Z+T=0$  and  $X^2+Y^2-9T^2=0$ . Then the Chow polynomial of  $X$  is given by:

$$\begin{aligned} \Gamma(L_1, \dots, L_6) &= 2L_2L_1 + 2L_3L_1 + 2L_3L_2 + L_4^2 \\ &\quad - 9L_6^2 - 2L_1L_5 - 18L_3L_6 - 8L_3^2 + L_2^2 \\ &\quad - 18L_6L_5 - 18L_3L_5 - 2L_1L_4 - 8L_5^2 + \\ &\quad 2L_4L_5 + 2L_1^2. \end{aligned}$$

Now we show how the usual point based representation of  $X$  can be extracted from  $\Gamma$ .

*Follow the following procedure:*

1. Pick three generic points,  $Q_1, Q_2, Q_3$  on the plane at infinity (last coordinate zero).
2. Consider a point  $P = [X, Y, Z, 1]^T$  in the affine piece of  $\mathbb{P}^3$ . The point  $P$  is lying on  $X$  if any linear combination of the three lines  $(PQ_i)_{i=1,2,3}$  is a zero of the Chow polynomial. This yields  $\binom{d+2}{d}$  equations defining the curve  $X$  completely.

This simple algorithm makes use of the fact that a point  $\mathbf{P}$  of  $\mathbb{P}^3$  is represented as a plane included in  $\mathbb{G}(1, 3)$  generated by three lines passing through  $\mathbf{P}$ . Hence a point is lying on the curve if the plane that represents it in  $\mathbb{G}(1, 3)$  is included into the hypersurface defined by the Chow polynomial of the curve.

In many applications, the sequences of locations of a moving point at each time instant an image was made is much more useful than the standard representation of the trajectory. Therefore we show how the Chow polynomial can be directly used for computing the intersection of the trajectory with a line. This line will be an optical ray in practice.

As before we shall denote the Chow polynomial by  $\Gamma$ . Each 2D point  $\mathbf{p}$  generates an optical ray  $\mathbf{L}_\mathbf{p}$  which is a zero of  $\Gamma$ .  $\mathbf{L}_\mathbf{p}$  is given by  $\widehat{\mathbf{M}}\mathbf{p}$  where  $\widehat{\mathbf{M}}$  is a  $6 \times 3$  matrix, which is a polynomial function of the camera matrix  $\mathbf{M}$ .  $\widehat{\mathbf{M}}$  maps an image point to the optical ray it generates with the optical center [9]. In order to compute where this optical ray meets the curve, we use the following procedure:

1. Pick three random and arbitrary points  $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$  in  $\mathbb{P}^3$ .
2. Give a parametric representation of  $\mathbf{L}_\mathbf{p}$ :  $\mathbf{P}(\lambda) = \mathbf{A} + \lambda\mathbf{B}$ .
3. Build the following parametric lines  $\mathbf{L}_i(\lambda) = \mathbf{P}(\lambda) \vee \mathbf{R}_i$ , for  $i = 1, 2, 3$ .
4. For every triplet  $\alpha, \beta, \gamma$ , we have  $\Gamma(\alpha\mathbf{L}_1(\lambda) + \beta\mathbf{L}_2(\lambda) + \gamma\mathbf{L}_3(\lambda)) = 0$ , for  $\mathbf{P}(\lambda)$  to be located on the curve. This yields  $\frac{1}{2}(d+1)(d+2)$  equations on  $\lambda$ .
5. Find the solution of the previous system. Let  $\lambda_0$  be that solution. Note that for solving the previous system it is enough to solve the first uni-variate polynomial equation and then to check which root is common to all the equations.
6. The point  $\mathbf{P}(\lambda_0) = \mathbf{A} + \lambda_0\mathbf{B}$  is the 3D point that we are looking for.

Note that in the previous algorithm in some very special cases, there might be more than one solution to the system in  $\lambda$ . This will be the case if the optical ray meets the curve at more than one point. This happens if an image was made when the moving point passed through either a point of the trajectory that occludes another point or a point of tangency of the curve with a line passing through the camera center. Those two cases are singular so they appear with probability zero. However in case of a singular point, the ambiguity can be solved by looking at neighboring points.

### 3 Trajectory Recovery with Known Cameras

Now we turn to consider our first problem. A moving point is viewed in one of several sequences. The camera matrices are assumed to be known over each sequence. However the cameras are *not assumed to be synchronized*. We want to recover the trajectory of the moving point. The trajectory recovery is then naturally done in the Grassmannian of lines  $\mathbb{G}(1, 3)$ . Let  $\Gamma$  be a Chow polynomial of the curve  $X$  generated by the motion of the point and let be  $d$  its degree. For now we will assume that  $d$  is known. We shall show in section 3.3 how this assumption can be addressed.

We first generate the number of independent measurements necessary to recover the point trajectory. Then we analyze carefully to which extent the measurements extracted from a particular sequence are independent and provide enough constraints to recover the trajectory. Finally we generalize our results to a general framework for trajectory recovery.

#### 3.1 How Many Measurements Are Necessary?

Each 2D point extracted from the images contributes one linear equation in  $\Gamma$ :

$$\Gamma(\widehat{\mathbf{M}}\mathbf{p}) = 0, \quad (3)$$

where  $\mathbf{M}$  is the camera matrix and  $\widehat{\mathbf{M}}$  is the  $6 \times 3$  matrix mapping each image point to its optical ray, as before. Thus the following result is immediate:

**Proposition 1** *The recovery of the trajectory of a moving point can be done linearly using at least  $k \geq \frac{1}{12}d^4 + \frac{2}{3}d^3 + \frac{23}{12}d^2 + \frac{7}{3}d$  independent measurements.*

*Proof.* The number of degrees of freedom of the Chow polynomial that must be constrained is  $N_d - 1 = \binom{d+5}{d} - \binom{d-2+5}{d-2} = \frac{1}{12}d^4 + \frac{2}{3}d^3 + \frac{23}{12}d^2 + \frac{7}{3}d$ .

The lower bounds on  $k$  for few examples:

1. for a moving point on a conic locus,  $k \geq 19$ ,
2. for a moving point on a cubic,  $k \geq 49$ ,
3. for a moving point on a quartic,  $k \geq 104$ .

A natural question is to know how many independent equations of type 3 each camera provides. This is the issue described in the next section.

#### 3.2 Which Measurements are Actually Independent?

We start by the simple case of a static camera.

**Proposition 2** *A static camera provides  $\binom{d+2}{d} - 1$  constraints on  $\Gamma$ .*

*Proof.* All the optical rays generated by a static camera belong to a plane included into the Grassmannian  $\mathbb{G}(1, 3)$ . Therefore a static camera allows to recover the intersection with a plane of the subvariety of  $\mathbb{G}(1, 3)$  we are looking for. This intersection is a curve of degree  $d$ . Therefore a static camera yields  $\binom{d+2}{d} - 1$  constraints on the Chow polynomials of the curve.

Note that when the camera is static, all the optical rays pass through the same point, the camera center. Hence the space of curves of a given degree constrained to pass through the optical rays contains the space of curves of the same degree passing through the camera center. This has a practical consequence. If several static cameras are viewing a moving point, the recovery of the trajectory might lead to a parasite solution, i.e. a curve passing through the camera centers. If such a parasite solution exists it must be eliminated.

Let us now consider a moving camera over a trajectory modeled by an algebraic curve of degree  $k$ . The question is to know how many independent measurements this camera provides on a point moving over a trajectory of degree  $d$ . When the number of measurements is large enough, this can be viewed as computing the number of degrees of freedom of the family of curves of degree  $d$  over the surface generated by the optical rays generated by the tracked point and the camera center. This is a question of algebraic geometry. We give here the result and postpone the proof to the appendix.

### Theorem 1 Fundamental Theorem

*A camera moving along a curve of degree  $k \geq 1$  provides  $H(k, d)$  constraints on the Chow Polynomial  $\Gamma$  of the trajectory of degree  $d \geq 1$  a moving point, where*

$$\begin{aligned} H(1, 1) &= 4 \\ H(2, 1) &= 5 \\ H(1, 2) &= 12 \\ H(2, 2) &= 17 \\ H(k, 2) &= 18 \text{ for } k \geq 3 \\ H(k, d) &= \begin{cases} N_d - \binom{d-k+5}{5} + \binom{d-k+3}{5} - 1 & \text{if } k \leq d-2 \\ N_d - 7 & \text{if } k = d-1 \\ N_d - 1 & \text{if } k \geq d \end{cases} \text{ for } d \geq 3 \end{aligned}$$

If several independently moving cameras are viewing a moving point, then the number of constraints this whole camera rig provides is the sum of each  $H(k, d)$  for each camera until the  $N_d - 1$  constraints are obtained.

### 3.3 A General Framework for Trajectory Recovery from Known Cameras

At this point we are in a position to propose a general framework for trajectory recovery. A set of *non-synchronized* cameras  $\mathbf{M}_i, i = 1, \dots, m$  which are either static or moving is viewing at a set of points  $\mathbf{P}_j, j = 1, \dots, n$  either static or moving. Since each camera is regarded as a dynamic system, the camera matrices are time dependent. Hence the camera matrix  $i$  at time  $k_i$  will be denoted by

$\mathbf{M}_{ik_i}$ . Note that the cameras are independent and in particular they are not supposed to be synchronized. Therefore the time samples are different between every two cameras. This means that the indices  $k_i$  are independent between two cameras. Let  $\mathbf{p}_{ijk_i}$  be the projection of the point  $\mathbf{P}_j$  onto the camera  $i$  at time  $k_i$ . All the  $\mathbf{M}_{ik_i}$  are known for all  $i$  and all  $k_i$ . This can be achieved during a preprocess by tracking static points over each sequence.

For a given point  $\mathbf{P}_j$ , the optical rays  $\mathbf{L}_{ijk_i} = \widehat{\mathbf{M}}_i \mathbf{p}_{ijk_i}$ , for all  $i$  and  $k_i$ , meet the trajectory of  $\mathbf{P}_j$ . Then according to the geometric entity generated for all  $i$  and  $k_i$  by those rays, the motion of  $\mathbf{P}_j$  can be recovered. Here we provide a table that gives the correspondence between the motion of the point and the geometry of the optical rays.

Motion of $\mathbf{P}_j$	Geometric entity generated by $\{\mathbf{L}_{ijk}\}$
Static point	Plane in $\mathbb{P}^5$ included in $\mathbb{G}(1, 3)$
Point moving on a line	Hyperplane section of $\mathbb{G}(1, 3)$
Point moving on a conic	Intersection of a quadric of $\mathbb{P}^5$ with $\mathbb{G}(1, 3)$
...	...
Point moving on a curve of degree $d$	Intersection of a hypersurface of degree $d$ with $\mathbb{G}(1, 3)$

Therefore this framework provides us with a way of segmenting static points from moving points and then to reconstruct the location of the former and the trajectory of the latter. Note that the determination of the trajectory degree is now handled in the context of fitting the Chow polynomial from the optical rays.

This framework can be seen as a complete generalization of [1], where only the case of moving points on a line was presented using the formalism of the linear line complex.

## 4 Experiments

The computation of the Chow Polynomial involves an estimation problem. However as mentioned above, the Chow polynomial is not uniquely defined. In order to get a unique solution, we have to add some constraints to the estimation problem which do not distort the geometric meaning of the Chow polynomial. This is simply done by imposing to the Chow polynomial to vanish over  $W_d$  additional arbitrary points of  $\mathbb{P}^5$  which do not lie on  $\mathbb{G}(1, 3)$ . The number of additional points necessary to get a unique solution is  $W_d = \binom{d+5}{d} - N_d$ , where  $d$  is the degree of the Chow polynomial.

#### 4.1 Synthetic Trajectory Triangulation

Let  $\mathbf{P} \in \mathbb{P}^3$  be a point moving on a cubic, as follows:

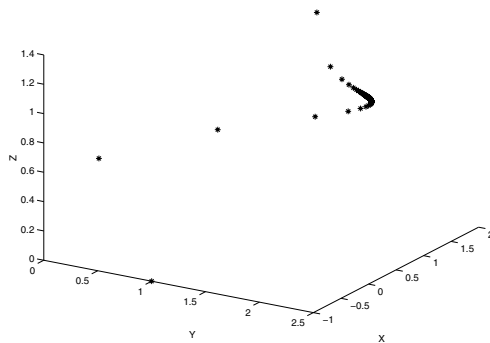
$$\mathbf{P}(t) = \begin{bmatrix} t^3 \\ 2t^3 + 3t^2 \\ t^3 + t^2 + t + 1 \\ t^3 + t^2 + t + 2 \end{bmatrix}$$

It is viewed by a moving camera. At each time instant a picture is made, we get a 2D point  $\mathbf{p}(t) = [x(t), y(t)]^T = [\frac{\mathbf{m}_1^T(t)\mathbf{P}(t)}{\mathbf{m}_3^T(t)\mathbf{P}(t)}, \frac{\mathbf{m}_2^T(t)\mathbf{P}(t)}{\mathbf{m}_3^T(t)\mathbf{P}(t)}]^T$ , where  $\mathbf{M}^T(t) = [\mathbf{m}_1(t), \mathbf{m}_2(t), \mathbf{m}_3(t)]$  is the transpose of the camera matrix at time  $t$ .

Then we build the set of optical rays generated by the sequence. The Chow polynomial is then computed and given below:

$$\begin{aligned} \Gamma(L_1, \dots, L_6) = & -72L_2^2L_3 + L_1^3 - 5L_1L_4L_5 - \\ & 18L_1L_3L_6 + 57L_2L_3L_5 + 48L_2L_4L_5 - 43L_1L_2L_4 - \\ & 10L_1L_3L_5 + 21L_1L_5L_6 - 30L_1L_4L_6 - 108L_2L_3L_6 + \\ & 41L_1L_2L_5 + 69L_1L_2L_6 - 26L_1L_2L_3 - 36L_2L_4^2 - \\ & 21L_2L_5^2 + 3L_3L_5^2 - 9L_3^2L_5 - 12L_4^2L_5 + 6L_4L_5^2 + \\ & 4L_1^2L_4 + 20L_2^3 - 13L_3^3 + 8L_4^3 - L_5^3 + 108L_2^2L_6 - \\ & 120L_2^2L_5 + 27L_3^2L_6 - 25L_1^2L_6 + 57L_2L_3^2 + \\ & 84L_2^2L_4 + 7L_1L_3^2 - L_1^2L_5 + 31L_1L_2^2 + \\ & 5L_1^2L_3 + L_1L_5^2 - 11L_1^2L_2 + 7L_1L_4^2 \end{aligned}$$

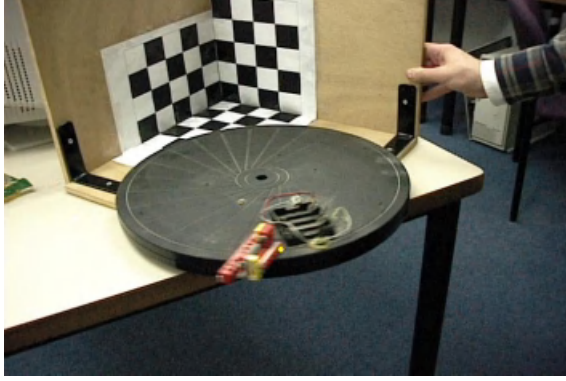
At this point we perform the algorithm described in section 2.2 and get exactly the sequence of locations of the moving point  $\mathbf{P}(t)$ . We show in figure 2 the recovered discrete locations of the point in 3D.



**Fig. 2.** The 3D locations of the point

## 4.2 Experiment on Real Images

A point is moving over a conic section. Four static non-synchronized cameras are looking at it. We show on figure 3 one image of one sequence.



**Fig. 3.** A moving point over a conic section

The camera matrices are computed using the calibration pattern. Every 2D measurement  $\mathbf{p}(t)$  is corrupted by additive noise, which we consider as an isotropic Gaussian noise  $\mathcal{N}(0, \sigma)$ . The variance is estimated to be about 2 pixels.

For each 2D point  $\mathbf{p}(t)$ , we form the optical ray it generates  $\mathbf{L}(t) = \widehat{\mathbf{M}}\mathbf{p}(t)$ . Then the estimation of the Chow polynomial is made using the optical rays  $\mathbf{L}(t)$ . In order to avoid the problem of scale, the Plücker coordinates of each line are normalized such that the last coordinate is equal to one. Hence the lines are represented by vectors in a five-dimensional affine space, denoted by  $\mathbf{L}_a(t)$ . Hence if  $\theta$  is a vector containing the coefficient of the Chow polynomial  $\Gamma$ ,  $\theta$  is solution of the following problem:

$$Z(\mathbf{L}_a(t))^T \theta = 0, \text{ for all } t,$$

with  $\|\theta\| = 1$  and  $Z(\mathbf{L}_a(t))$  is a vector which coordinates are monomials generated by the coordinates of  $\mathbf{L}_a(t)$ . Following [4,18], in order to obtain a reliable estimate, the solution  $\theta$  is computed using a maximum likelihood estimator. This allows to take into account the fact that each  $Z(\mathbf{L}_a(t))$  has a different covariance matrix, or in other terms that the noise is *heteroscedastic*. More precisely, each  $Z(\mathbf{L}_a(t))$  has the following covariance matrix:

$$\mathbf{C}_{\mathbf{L}(t)} = \mathbf{J}_\phi \mathbf{J}_n \widehat{\mathbf{M}} \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \widehat{\mathbf{M}}^T \mathbf{J}_n^T \mathbf{J}_\phi^T,$$

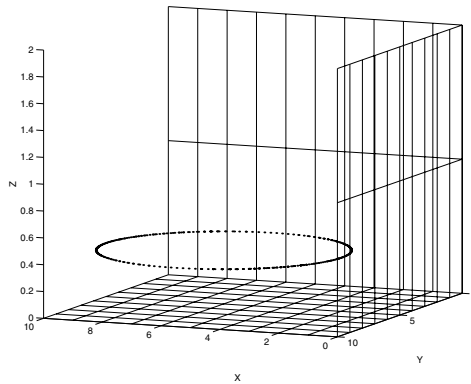
where  $\mathbf{M}$  is the camera matrix and  $\mathbf{J}_n$  and  $\mathbf{J}_\phi$  are respectively the Jacobian matrices of the normalization of  $\mathbf{L}(t)$  and of the map sending  $\mathbf{L}_a(t)$  to  $Z(\mathbf{L}_a(t))$ .



That is for  $\mathbf{L}(t) = [L_1, L_2, L_3, L_4, L_5, L_6]^T$ , we have:

$$\mathbf{J}_n = \begin{bmatrix} \frac{1}{L_6} & 0 & 0 & 0 & 0 & -\frac{L_1}{L_6^2} \\ 0 & \frac{1}{L_6} & 0 & 0 & 0 & -\frac{L_2}{L_6^2} \\ 0 & 0 & \frac{1}{L_6} & 0 & 0 & -\frac{L_3}{L_6^2} \\ 0 & 0 & 0 & \frac{1}{L_6} & 0 & -\frac{L_4}{L_6^2} \\ 0 & 0 & 0 & 0 & \frac{1}{L_6} & -\frac{L_5}{L_6^2} \end{bmatrix},$$

and  $\mathbf{J}_\phi$  is similarly computed. Then we use the method presented in [4] to perform the estimation. The result is stable where starting with a good initial guess. In order to handle more general situation we further stabilize it by incorporating some extra constraints that come some our *a-priori* knowledge of the form of the solution. The final result is presented in figure 4.



**Fig. 4.** The trajectory rendered in the calibration pattern

## 5 Conclusion and Future Work

In this paper, we presented a general method for triangulating arbitrary trajectories of moving points in dynamic scenes. Our approach is based on a new (to computer vision) representation of curves. Within this representation the unknowns appear linearly. Hence an arbitrary trajectory can be approximated by algebraic piece-wise approximation. Experiments also showed the feasibility of our approach.

Further research is needed to get complete reliable estimate of the Chow polynomial because of the heteroscedasticity of the noise and the high dimensionality of the solution space.

Moreover, our approach gives rise to interesting questions that will be investigated in future work. Those questions are related to dual extensions of the present work:

- Moving camera recovery from known moving points or equivalently static cameras recovery from known curves;
- Structure and motion recovery in dynamic environments when some assumptions on the nature of the motions can be made.

## A Appendix: The Fundamental Theorem

We prove the theorem 1.

*Proof.* The proof is based on cohomological computation [15]. Let  $X$  (resp.  $Y$ ) be the point (resp. camera center) trajectory. Each observation generates an optical ray joining the camera center and the point. Let  $\mathbf{L}_1, \dots, \mathbf{L}_n$  be these  $n$  lines joining  $X$  and  $Y$ . Let  $\Gamma_X$  and  $\Gamma_Y$  be the Chow polynomial of  $X$  and  $Y$  respectively. We shall denote by  $Z(\Gamma_X)$  and  $Z(\Gamma_Y)$  the sets where they vanish. Let  $V = Z(\Gamma_X) \cap Z(\Gamma_Y) \cap \mathbb{G}(1, 3)$ . For  $n \gg 1$ , we have

$$\begin{aligned} \{ \Gamma \in H^0(\mathbb{P}^5, \mathcal{O}_{\mathbb{P}^5}(d)) : \Gamma(L_i) = 0, i = 1, \dots, n \} = \\ \{ \Gamma \in H^0(\mathbb{P}^5, \mathcal{O}_{\mathbb{P}^5}(d)) : \Gamma|_V \equiv 0 \} = \mathcal{I}_{V, \mathbb{P}^5}(d). \end{aligned}$$

So, we want to compute  $\dim(\mathcal{I}_{V, \mathbb{P}^5}(d))$ , which is the dimension of the space of  $\Gamma_X$ , or, equivalently,  $h^0(V, \mathcal{O}_V(d)) = h^0(\mathcal{O}_{\mathbb{P}^5}(d)) - \dim(\mathcal{I}_{V, \mathbb{P}^5}(d))$ . Since  $V$  is a complete intersection of degree  $(d, k, 2)$  in  $\mathbb{P}^5$ , the dimension of  $\mathcal{I}_{V, \mathbb{P}^5}(d)$  should be equal to

$$h^0(\mathcal{O}_{\mathbb{P}^5}(d-2)) + h^0(\mathcal{O}_{\mathbb{P}^5}(d-k)) - h^0(\mathcal{O}_{\mathbb{P}^5}(d-k-2)) + 1.$$

As a consequence

$$h^0(V, \mathcal{O}_V(d)) = N_d - (h^0(\mathcal{O}_{\mathbb{P}^5}(d-k)) - h^0(\mathcal{O}_{\mathbb{P}^5}(d-k-2)) + 1).$$

## References

1. S. Avidan and A. Shashua, Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):348-357, 2000.
2. M. Barnabei, A. Brini and G.C. Rota, On the exterior calculus of invariant theory. *Journal of Algebra*, 96, 120-160(1985)
3. B. Buchberger and F. Winkler, Gröbner Bases and Applications. Cambridge University Press, 1998.
4. W. Chojnacki, M. Brooks, A. van den Hengel and D. Gawley, On the Fitting of Surfaces to Data with Covariances. *PAMI*, vol. 22, Nov. 2000.
5. D. Cox, J. Little and D. O'Shea Ideals, Varieties and Algorithms, 2nd. Springer-Verlag, 1996.
6. A Multibody Factorization Method for Independent Moving Objects. *International Journal Of Computer Vision*, Kluwer, Vo. 29, Sep. 1998.
7. D. Eisenbud, Commutative Algebra with a view toward algebraic geometry. Springer-Verlag, 1995.

8. D. Eisenbud and J. Harris, *The Geometry of Schemes*. Springer-Verlag, 2000.
9. O.D. Faugeras and Q.T. Luong, *The Geometry Of Multiple Images*. MIT Press, 2001.
10. W. Fulton, *Algebraic Curves: An Introduction to Algebraic Geometry*. Addison-Wesley Publishing Company: The Advanced Book Program.
11. A.W. Fitzgibbon and A. Zisserman, Multibody Structure and Motion: 3D Reconstruction of Independently Moving Objects. In *Proceedings of European Conference on Computer Vision*, pages 891-906, June 2000.
12. M. Han and T. Kanade, Reconstruction of a Scene with Multiple Linearly Moving Points. In *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, June 2000.
13. J. Harris, *Algebraic Geometry, a first course*. Springer-Verlag, 1992.
14. R.I. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*. Cambridge University Press, 2000.
15. R. Hartshorne, *Algebraic Geometry*. Springer-Verlag, 1977.
16. J.Y. Kaminski, M. Fryers, A. Shashua and M. Teicher, Multiple View Geometry of Non-planar Algebraic Curves. In *Proceedings of International Conference on Computer Vision*, July 2001
17. R.A. Manning C.R. Dyer, Interpolating view and scene motion by dynamic view morphing. In *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, pages 388-394, June 1999.
18. B. Matei and P. Meer, A General Method for Errors-in-variables Problems in Computer Vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, 2000.
19. D. Segal and A. Shashua, 3D Reconstruction from Tangent-of-Sight Measurements of a Moving Object Seen from a Moving Camera. In *Proceedings of European Conference on Computer Vision*, pages 507-521, June 2000.
20. A. Shashua and L. Wolf, Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Points. In *Proceedings of European Conference on Computer Vision*, pages 507-521, June 2000.
21. Y. Wexler and A. Shashua, On the synthesis of dynamic scenes from reference view. In *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, June 2000.
22. L. Wolf and A. Shashua, On Projection Matrices  $\mathbb{P}^k \rightarrow \mathbb{P}^2$ ,  $k = 3, \dots, 6$ , and their Applications in Computer Vision. In *Proceedings of IEEE International Conference on Computer Vision*, July 2001.

# Surviving Dominant Planes in Uncalibrated Structure and Motion Recovery

Marc Pollefeys, Frank Verbiest, and Luc Van Gool

Center for Processing of Speech and Images (PSI), K.U.Leuven, Belgium,  
`firstname.lastname@esat.kuleuven.ac.be`

**Abstract.** In this paper we address the problem of uncalibrated structure and motion recovery from image sequences that contain dominant planes in some of the views. Traditional approaches fail when the features common to three consecutive views are all located on a plane. This happens because in the uncalibrated case there is a fundamental ambiguity in relating the structure before and after the plane. This is, however, a situation that is often hard to avoid in man-made environments. We propose a complete approach that detects the problem and defers the computation of parameters that are ambiguous in projective space (i.e. the registration between partial reconstructions only sharing a common plane and poses of cameras only seeing planar features) till after self-calibration. Also a new linear self-calibration algorithm is proposed that couples the intrinsics between multiple subsequences. The final result is a complete metric 3D reconstruction of both structure and motion for the whole sequence. Experimental results on real image sequences show that the approach yields very good results.

## 1 Introduction

There has been a lot of progress in uncalibrated structure and motion (USaM) recovery over the last decade. Faugeras [3] and Hartley et al. [9] have shown that starting from an uncalibrated image pair a projective reconstruction was possible. The use of robust statistics for the computation of the epipolar geometry made it possible to obtain good results on real image data [27,21]. These approaches were later extended to image sequences (e.g. [2,15]). In parallel with these developments the possibility to upgrade a projective reconstruction to metric (i.e. Euclidean up to scale) based on constraints on the intrinsics was shown [4]. Over the years many different methods have been proposed for constant [11,16,24] and varying intrinsics [17]. Therefore, starting from an uncalibrated image sequence it became possible to retrieve a metric 3D reconstruction. Compared to the more traditional structure and motion recovery approaches where the camera is calibrated separately, USaM recovery offers an important increase in flexibility.

However, an important –but often ignored– problem of the uncalibrated approach is that it breaks down in the case of a planar scene. The relative pose between views can not be determined when all common features are located in a plane. In fact this is a specific case of the more general problem of critical surfaces (e.g. [12]). However, other cases are much less probable to be encountered in practice. Recently, there has been quite some work on dealing with planes in USaM recovery. Liu et al. [14] and Bartoli et al. [1] looked at architectural scenes containing planes. Note, however, that these techniques

require multiple planes or general structure and would therefore fail on the cases treated by this paper. Another interesting approach was proposed by Rother and Carlsson [19]. In this case a linear solution is obtained when a single plane can be seen in all views. Nevertheless, in each view at least two points not located on the plane are required.

In fact, the work that comes closest to solving the problem was carried out by Torr et al. In [23] a robust model selection criterion to differentiate between general 3D structure and planar structure was proposed. This allows to automatically identify the views where the structure is not sufficiently general and to deal with them accordingly (i.e. estimating a homography instead of the epipolar geometry). Although some possibilities were sketched on how this could be used to solve the planarity induced ambiguities in the recovery of USaM, the paper mostly focuses on the model selection and feature tracking issues. No general solution is provided to solve the ambiguity between the structure and motion of subsequences only sharing a single plane.

The main subject of this paper consists of proposing a complete approach to uncalibrated structure and motion recovery that can deal with dominant planes. The approach starts by extending the work by Torr et al. [23] to the 3-view case (which is necessary as will be seen later) so that the difference can be made between subsequences observing sufficiently general structure and subsequences where the tracked features are all located on a single plane. The next step consists of independently recovering the projective structure of the different 3D subsequences. Then the reconstruction for the 3D subsequences is extended with the reconstruction of the planes. Once this is done self-calibration is used to recover the metric structure. To improve the accuracy and robustness the approach couples the intrinsics between the different subsequences. This is especially important to allow successful self-calibration of shorter subsequences. These results are refined using a bundle adjustment that couples the intrinsics for all the subsequences. At this stage a pose estimation algorithm can be used to determine the motion of the camera over the planar parts. The different parts are also assembled (by aligning the overlapping planes). Finally, a global bundle adjustment is carried out to obtain a maximum likelihood estimation of the metric structure and motion for the whole sequence.

The paper is organized as follows. In the next section a traditional uncalibrated structure and motion approach is reviewed. Then, the problem caused by dominant planes is described and the approach for detecting the problem is described. The actual approach to solve it is described in Section 5 (partial projective USaM recovery), Section 6 (coupled self-calibration) and Section 7 (complete metric SaM recovery). In the final sections results and conclusions are presented.

## 1.1 Notations

Points are represented by homogeneous 4-vectors  $M$  in 3-space, and by homogeneous 3-vectors  $m$  in the image. A plane is represented by a homogeneous 4-vector  $\Pi$  and a point  $M$  is on a plane if  $\Pi^T M = 0$ . A point  $M$  is mapped to its image  $m$  through perspective projection, represented by a  $3 \times 4$  projection matrix  $P$  as  $m \sim PM$ . The symbol  $\sim$  indicates equality up to a non-zero scale factor. In a metric coordinate system the matrix  $P$  can be factorized in intrinsic and extrinsic camera parameters:  $P = K[R \ t]$  where the upper-triangular matrix  $K$  is given by the following equation:

$$\mathbf{K} = \begin{bmatrix} f & s & u \\ rf & v \\ 1 \end{bmatrix} \quad (1)$$

with  $f$  the focal length (measured in pixels),  $r$  the aspect ratio,  $(u, v)$  the coordinates of the principal point and  $s$  a factor that is zero when the pixels are rectangular. To deal with radial distortion, the perspective projection model is extended to  $\mathbf{K}\mathcal{R}([\mathbf{R} \ \mathbf{t}]\mathbf{M})$  with  $\mathcal{R}([x \ y \ 1]^\top) \sim [x \ y \ w]^\top$  and  $w^{-1} = (1 + k_1 r^2 + k_2 r^4)$  and  $r^2 = x^2 + y^2$  and  $k_1$  and  $k_2$  are parameters of radial distortion. The fundamental matrix  $\mathbf{F}$  and the two image homography  $\mathbf{H}$ , are both  $3 \times 3$  homogeneous matrices. A point  $\mathbf{m}$  located in the plane corresponding to the homography  $\mathbf{H}$  is transferred from one image to the other according to  $\mathbf{m}' \sim \mathbf{H}\mathbf{m}$ . A more complete description of these concepts can be found in [10].

## 2 General Projective Structure and Motion Recovery

Starting from an uncalibrated image sequence the first step consists of relating the different images to each other. This is not an easy problem. In general a restricted number of corresponding points is sufficient to determine the epipolar geometry between the images. Since not all points are equally suited for matching (e.g. pixels in a homogeneous region), the first step consist of selecting feature points [8] that are suited for automated matching. Features of consecutive views are compared and a number of potential correspondences are obtained. From these the epipolar geometry can be computed. However, the initial set of corresponding points is typically contaminated with an important number of outliers. In this case, a traditional least-squares approach will fail and therefore a robust method is used [21,27,5]. Once the epipolar geometry has been obtained it can be used to guide the search for additional correspondences. These can then in turn be used to further refine the epipolar geometry.

The relation between the views and the correspondences between the features can then be used to retrieve the structure of the scene and the motion of the camera. The approach that is used is related to [2] but is fully projective and therefore not dependent on any approximation. This is achieved by strictly carrying out all measurements in the images, i.e. using only reprojection errors. At first two images are selected and an initial projective reconstruction frame is set-up [3,9]. Then the pose of the camera for the other views is determined in this frame and for each additional view the initial reconstruction is refined and extended. Once the structure and motion has been determined for the whole sequence, the results are refined through a projective bundle adjustment [26]. To minimize the presence of a consistent bias in the reconstruction, this bundle adjustment takes into account radial distortion (around the image center). Then the ambiguity is restricted to metric through self-calibration. A modified version of [17] is used (see Section 6). Finally, a metric bundle adjustment is carried out to obtain an optimal estimation for both structure and motion.

## 3 Problems with Planes

The projective structure and motion approach described in the previous section assumes that both motion and structure are general. When this is not the case, the approach can fail.

In the case of motion this will happen when the camera is purely rotating. A solution to this problem was proposed in [23]. Here we will assume that care is taken during acquisition to not take multiple images from the same position so that this problem doesn't occur<sup>1</sup>.

Scene related problems occur when (part of) the scene is purely planar. In this case it is not possible anymore to determine the epipolar geometry uniquely. If the scene is planar, the image motion can be fully described by a homography. Since  $\mathbf{F} = [\mathbf{e}']_{\times} \mathbf{H}$  (with  $[\mathbf{e}']_{\times}$  the vector product with the epipole  $\mathbf{e}'$ ), there is a 2 parameter family of solutions for the epipolar geometry. In practice robust techniques would pick a random solution based on the inclusion of some outliers.

Assuming we would be able to detect this degeneracy, the problem is not completely solved yet. Obviously, the different subsequences containing sufficient general 3D structure could be reconstructed separately. The structure of subsequences containing only a single plane could also be reconstructed as such. These planar reconstructions could then be inserted into the neighboring 3D projective reconstructions. However, there remains an ambiguity on the transformation relating two 3D projective reconstruction only sharing a common plane. The plane shared by the two reconstructions can be uniquely parameterized by three 3D points ( $3 \times 3$  parameters) and a fourth point in the plane (2 free parameters) to determine the projective basis within the plane. The ambiguity therefore has  $15 - 11 = 4$  degrees of freedom. An illustration is given on the left side of Figure 1. Note also that it can be very hard to avoid this type of degeneracy as can be seen from the right side of Figure 1. Many scenes have a configuration similar to this one.

## 4 Detecting Dominant Planes

The first part of the solution consists of detecting the cases where only planar features are being matched. The Geometric Robust Information Criterion (GRIC) model selection approach proposed in [22] is briefly reviewed. The GRIC selects the model with the lowest score. The score of a model is obtained by summing two contributions. The first one is related to the goodness of the fit and the second one is related to the parsimony of the model. It is important that a robust Maximum Likelihood Estimator (MLE) be used for estimating the different structure and motion models being compared through GRIC. GRIC takes into account the number  $n$  of inliers plus outliers, the residuals  $e_i$ , the standard deviation of the measurement error  $\sigma$ , the dimension of the data  $r$ , the number  $k$  of motion model parameters and the dimension  $d$  of the structure:

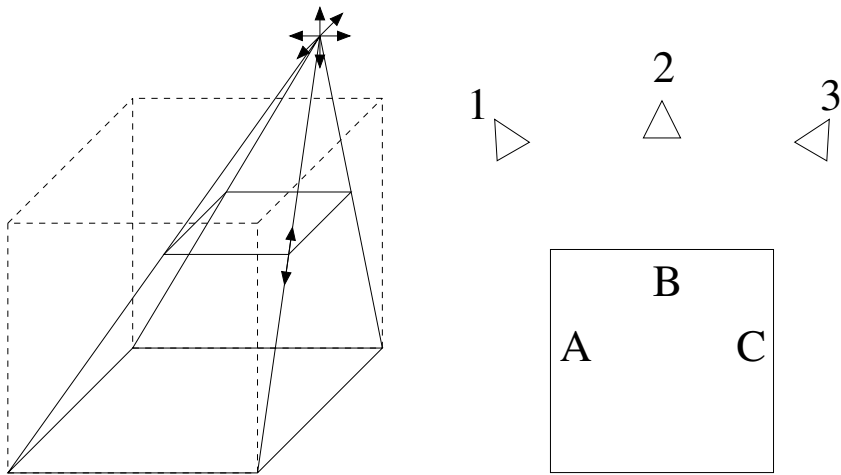
$$\text{GRIC} = \sum \rho(e_i^2) + (nd \ln(r) + k \ln(rn)) . \quad (2)$$

where  $\rho(e^2)$

$$\rho(e^2) = \min \left( \frac{e^2}{\sigma^2}, 2(r - d) \right) . \quad (3)$$

---

<sup>1</sup> Note that the approach would still work if the pure rotation takes place while observing a planar part.

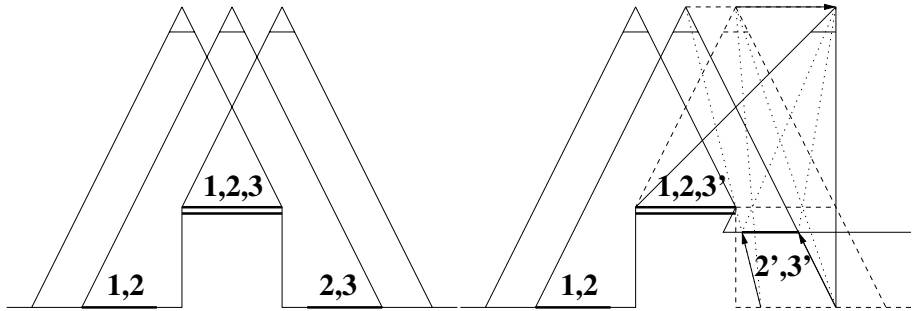


**Fig. 1.** Left: Illustration of the four-parameter ambiguity between two projective reconstructions sharing a common plane. If the base of the cube is shared, a projective transformation can still affect the height of the cube and the position of the third vanishing point. Right: A fundamental problem for many (man-made) scenes is that it is not possible to see A,B and C at the same time and therefore when moving from position 1 to position 3 the planar ambiguity problem will be encountered.

In the above equation  $nd \ln(r)$  represents the penalty term for the structure having  $n$  times  $d$  parameters each estimated from  $r$  observations and  $k \ln(rn)$  represents the penalty term for the motion model having  $k$  parameters estimated from  $rn$  observations.

For each image pair  $\text{GRIC}(\mathbf{F})$  and  $\text{GRIC}(\mathbf{H})$  can be compared. If  $\text{GRIC}(\mathbf{H})$  yields the lowest value it is assumed that most matched features are located on a dominant plane and that a homography model is therefore appropriate. On the contrary, when  $\text{GRIC}(\mathbf{F})$  yields the lowest value one could assume, as did Torr [23], that standard projective structure and motion recovery could be continued. In most cases this is correct, however, in some cases this might still fail. An illustration of the problem is given on the left side of Figure 2 where both  $\mathbf{F}_{12}$  and  $\mathbf{F}_{23}$  could be successfully computed, but where structure and motion recovery would fail because all features common to the three views are located on a plane. Estimating the pose of camera 3 from features reconstructed from views 1 and 2 or alternatively estimating the trifocal tensor from the triplets would yield a three-parameter family of solutions. However, imposing reconstruction 1–2 and reconstruction 2–3 to be aligned (including the center of projection for view 2) would reduce the ambiguity to a one-parameter family of solutions. This ambiguity is illustrated on the right side of Figure 2. Compared to the reference frame of cameras 1 and 2 the position of camera 3 can change arbitrarily as long as the epipole in image 2 is not modified (i.e. motion along a line connecting the center of projections of image 2 and 3). Since intersection has to be preserved and the image of the common plane also has to be invariant, the transformation of the rest of space is completely determined. Note –as seen in Figure 2– that this remaining ambiguity could still cause an important distortion.





**Fig. 2.** Left: Although each pair contains non-coplanar features, the three views only have coplanar points in common. Right: Illustration of the remaining ambiguity if the position of the center of projection for view 2 corresponds for structure 1–2 and 2–3.

For the reason described above we propose to use the GRIC criterion on triplets of views ( $r = 6$ ). On the one hand we have  $\text{GRIC}(\mathbf{PPP})$  based on a model containing 3 projection matrices (up to a projective ambiguity) with  $k = 3 \times 11 - 15 = 18$  and  $d = 3$  (note that using a model based on the trifocal tensor would be equivalent), on the other hand we have  $\text{GRIC}(\mathbf{HH})$  based on a model containing 2 homographies with  $k = 2 \times 8 = 16$  and  $d = 2$ . To efficiently compute the MLE of both  $\mathbf{PPP}$  and  $\mathbf{HH}$  the sparse structure of the problem is exploited (similar to bundle adjustment). We can now differentiate between two different cases: Case A:  $\text{GRIC}(\mathbf{PPP}) < \text{GRIC}(\mathbf{HH})$ : three views observe general 3D structure. Case B:  $\text{GRIC}(\mathbf{PPP}) > \text{GRIC}(\mathbf{HH})$ : common structure between three views is planar. Note that it does not make sense to consider mixed cases such as  $\mathbf{HF}$  or  $\mathbf{FH}$  since for structure and motion recovery triplets are needed which in these cases would all be located on a plane anyway.

Note that in addition, one should verify that a sufficient number of triplets remain (say more than 50) to allow a reliable estimation. When too few points are seen in common over three views, the sequence is also split up. In a later stage it can be reassembled (using the procedure laid out in Section 7). This avoids the risk of a (slight) change of projective basis due to an unreliable estimation based on too few points. Note that it is important to avoid this, since this would mean that different transformations would be required to bring the different parts of the recovered structure and motion back to a metric reference frame. In practice this causes self-calibration to fail and should therefore be avoided.

## 5 Partial Projective Structure and Motion Recovery

The sequence is first traversed and separated in subsequences. For subsequences with sufficient 3D structure (case A) the approach described in Section 2 is followed so that the projective structure and motion is recovered. When a triplet corresponds to case B, only planar features are tracked and reconstructed (in 2D). A possible partitioning of an image sequence is given in Table 1. Note that the triplet 3-4-5 would cause an approach based on [23] to fail.

**Table 1.** Example on how a sequence would be partitioned based on the different cases obtained in the model selection step. Underlined F correspond to cases that would not be dealt with appropriately using a pairwise analysis.

case	AABAABBBBAAA
3D	PPPP PPPPP
2D	HH HHHHHH
3D	PPPP
	<u>FFFFFHHHHFFFF</u>

Suppose the plane  $\Pi$  is labeled as a dominant plane from view  $i$  based on features tracked in views  $(i - 1, i, i + 1)$ . In general, some feature points  $\mathbf{M}_\Pi$  located on  $\Pi$  will have been reconstructed in 3D from previous views (e.g.  $i$  and  $(i - 1)$ ). Therefore, the coefficients of  $\Pi$  can be computed from  $\mathbf{M}_\Pi^\top \Pi = 0$ . Define  $\mathbf{M}_\Pi$  as the right null space of  $\Pi^\top$  ( $4 \times 3$  matrix).  $\mathbf{M}_\Pi$  represents 3 supporting points for the plane  $\Pi$  and let  $\mathbf{m}_{\Pi i} = \mathbf{P}_i \mathbf{M}_\Pi$  be the corresponding image projections. Define the homography  $\mathbf{H}_{i\Pi} = \mathbf{m}_{\Pi i}^{-1}$ , then the 3D reconstruction of image points located in the plane  $\Pi$  are obtained as follows:

$$\mathbf{M}_i = \mathbf{M}_\Pi \mathbf{H}_{i\Pi} \mathbf{m}_i \quad (4)$$

Similarly, a feature  $\mathbf{m}_j$  seen in view  $j(> i)$  can be reconstructed as:

$$\mathbf{M}_j = \mathbf{M}_\Pi \mathbf{H}_{i\Pi} (\mathbf{H}_{ij}^\Pi)^{-1} \mathbf{m}_j \quad (5)$$

where  $\mathbf{H}_{ij}^\Pi = \mathbf{H}_{i(i+1)}^\Pi \cdots \mathbf{H}_{(j-1)j}^\Pi$ .

## 6 Coupled Self-Calibration

Once the projective structure and motion has been computed for each subsequence, standard self-calibration approaches could be used on the subsequences. However, some of these could be too short to obtain good results.

In this section a self-calibration approach is proposed that couples the camera intrinsics for the different subsequences containing general 3D structure. The approach is based on the approach proposed in [17], but was adapted to better reflect a priori expectations for the unknowns. The approach is based on the projection equation for the absolute quadric [24]:

$$\mathbf{K}\mathbf{K}^\top \sim \mathbf{P}\mathbf{\Omega}^*\mathbf{P}^\top \quad (6)$$

where  $\mathbf{\Omega}^*$  represents the absolute quadric. In metric space  $\mathbf{\Omega}^* = \text{diag}(1, 1, 1, 0)$ , in projective space  $\mathbf{\Omega}^*$  is a  $4 \times 4$  symmetric rank 3 matrix representing an imaginary disc-quadric. By transforming the image so that a typical focal length (e.g. 50mm) corresponds to unit length in the image and that the center of the image is located at the origin, realistic expectations for the intrinsics are  $\log(f) = \log(1) \pm \log(3)$  (i.e.  $f$  is typically in the range [17mm, 150mm]),  $r = \log(1) \pm \log(1.1)$ ,  $u = 0 \pm 0.1$ ,  $v = 0 \pm 0.1$ ,  $s = 0$ . These expectations can be used to obtain a set of weighted self-calibration equations

from Equation (6):

$$\begin{aligned} \frac{1}{9\nu} \left( P_1 \Omega^* P_1^\top - P_3 \Omega^* P_3^\top \right) &= 0 & \frac{1}{0.01\nu} \left( P_1 \Omega^* P_2^\top \right) &= 0 \\ \frac{1}{9\nu} \left( P_2 \Omega^* P_2^\top - P_3 \Omega^* P_3^\top \right) &= 0 & \frac{1}{0.1\nu} \left( P_1 \Omega^* P_3^\top \right) &= 0 \\ \frac{1}{0.2\nu} \left( P_1 \Omega^* P_1^\top - P_2 \Omega^* P_2^\top \right) &= 0 & \frac{1}{0.1\nu} \left( P_2 \Omega^* P_3^\top \right) &= 0 \end{aligned} \quad (7)$$

where  $P_i$  is the  $i$ -th row of a projection matrix and  $\nu$  a scale factor that is initially set to 1 and later on to  $P_3 \Omega^* P_3^\top$  with  $\Omega^*$  the result of the previous iteration. In practice iterating is not really necessary, but a few iterations can be performed to refine the initial result. Experimental validation has shown that this approach yields much better results than the original approach described in [17]. This is mostly due to the fact that constraining all parameters (even with a small weight) allows to avoid most of the problems due to critical motion sequences [20,13] (especially the specific additional case for the linear algorithm [18]).

When choosing  $\mathbf{P} = [\mathbf{I} | \mathbf{0}]$  for one of the projection matrices it can be seen from Equation (6) that  $\Omega^*$  can be written as:

$$\Omega^* = \begin{bmatrix} \mathbf{K}\mathbf{K}^\top & \mathbf{a} \\ \mathbf{a}^\top & b \end{bmatrix} \quad (8)$$

Now the set of equations (7) can thus be written as:

$$[\mathbf{C} \ \mathbf{D}] \begin{bmatrix} \mathbf{k} \\ \mathbf{a} \\ b \end{bmatrix} \quad (9)$$

where  $\mathbf{k}$  is a vector containing 6 coefficients representing the matrix  $\mathbf{K}\mathbf{K}^\top$ ,  $\mathbf{a}$  is a 3-vector and  $b$  a scalar and  $\mathbf{C}$  and  $\mathbf{D}$  are matrices containing the coefficients of the equations. Note that this can be done independently for every 3D subsequence.

If the sequence is recorded with constant intrinsics, the vector  $\mathbf{k}$  will be common to all subsequences and one obtains the following coupled self-calibration equations:

$$\begin{bmatrix} \mathbf{C}_1 & \mathbf{D}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_2 & \mathbf{0} & \mathbf{D}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{C}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_n \end{bmatrix} \begin{bmatrix} \mathbf{k} \\ \mathbf{a}_1 \\ b_1 \\ \mathbf{a}_2 \\ b_2 \\ \vdots \\ \mathbf{a}_n \\ b_n \end{bmatrix} \quad (10)$$

As will be seen in the experiments this approach is very successful. The most important feature is that through the coupling it allows to get good results even for the shorter subsequences. For each subsequence a transformation to upgrade the reconstruction from projective to metric can be obtained from the constraint  $\mathbf{T}_i \Omega_i^* \mathbf{T}_i^\top = \text{diag}(1, 1, 1, 0)$  (through Cholesky factorization). This result is then further refined through a metric bundle adjustment that also couples the intrinsics of the different subsequences.

## 7 Combined Metric Structure and Motion Recovery

Now that the metric structure of the subsequences has been recovered, the pose of the camera can also be determined for the viewpoints observing only planar points. Since the intrinsics have been computed, a standard pose estimation algorithm can be used. We use Grunert's algorithm as described in [7]. To deal with outliers a robust approach was implemented [5].

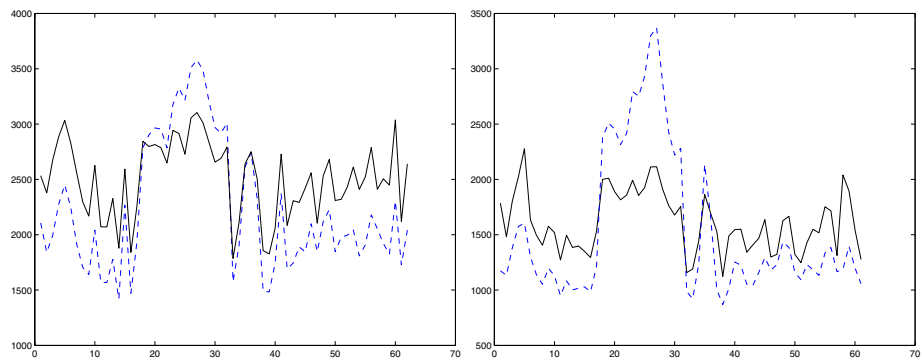
Finally, it becomes possible to align the structure and motion recovered for the separate subsequences based on common points. Note that these points are all located in a plane and therefore some precautions have to be taken to obtain results using linear equations. However, since 3 points form a basis in a metric 3D space, additional points out of the plane can easily be generated (i.e. using the vector product) and used to compute the relative transform using linear equations. Here again a robust approach is used.

Now that all structure and motion parameters have been estimated for the whole sequence. A final bundle adjustment is carried out to obtain a globally optimal solution.

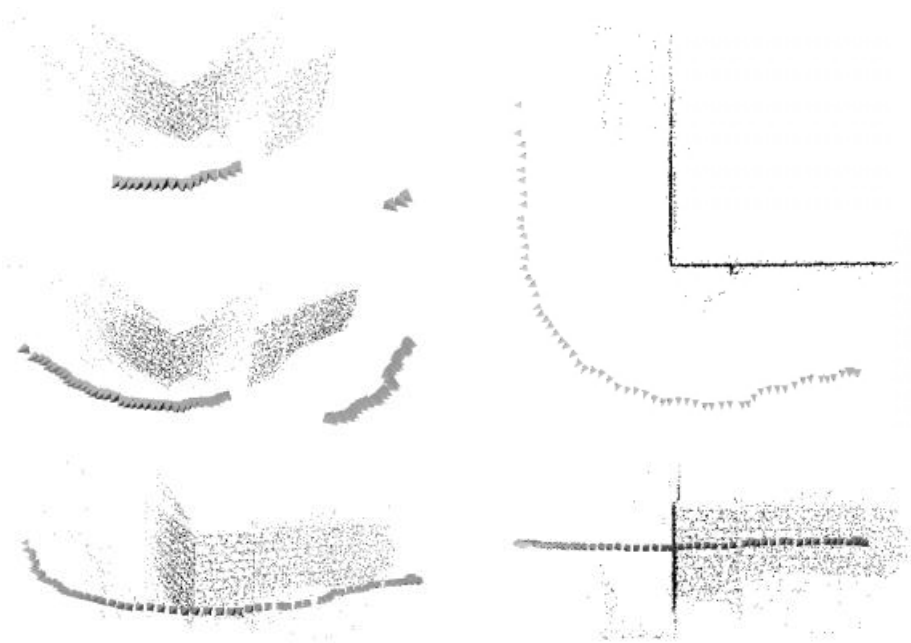


**Fig. 3.** Some of the 64 images of the *corner* sequence.

Figure 5 shows results for different stages of our approach. At the top-left the recovered metric structure and motion for the two subsequences that contain sufficiently general structure is given (after coupled self-calibration). Then, both structure and motion are extended over the planar parts. This can be seen in the middle-left part of the figure. At the bottom-left the complete structure and motion for the whole sequence is shown after bundle adjustment. On the right side of the figure orthogonal top and front views are shown.



**Fig. 4.** Left: GRIC(**F**) (solid/black line) and GRIC(**H**) (dashed/blue line). Right: GRIC(**PPP**) (solid/black line) and GRIC(**HH**) (dashed/blue line).



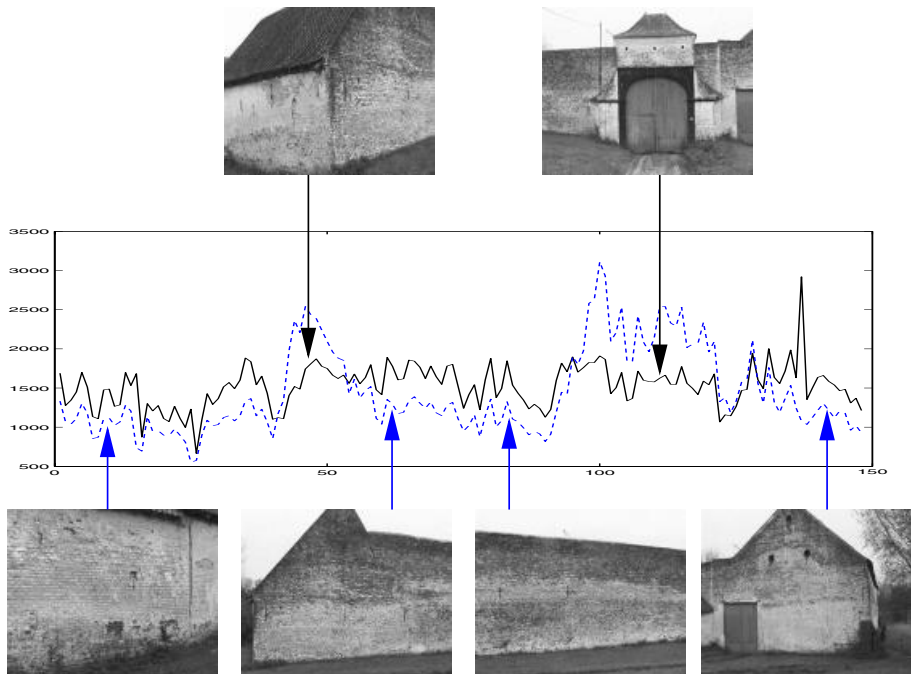
**Fig. 5.** Left: different stages of the structure and motion recovery, Right: orthogonal views of the final result.

## 8 Results

In this section results of our approach on two real image sequences are shown. The first image sequence was recorded from a corner of our institute. The *corner* sequence contains 64 images recorded using a Sony TRV900 digital camcorder in progressive scan mode. The images therefore have a resolution of  $720 \times 576$  (PAL). Some of the images are shown in Figure 3. Note that the images contain quite some radial distortion.

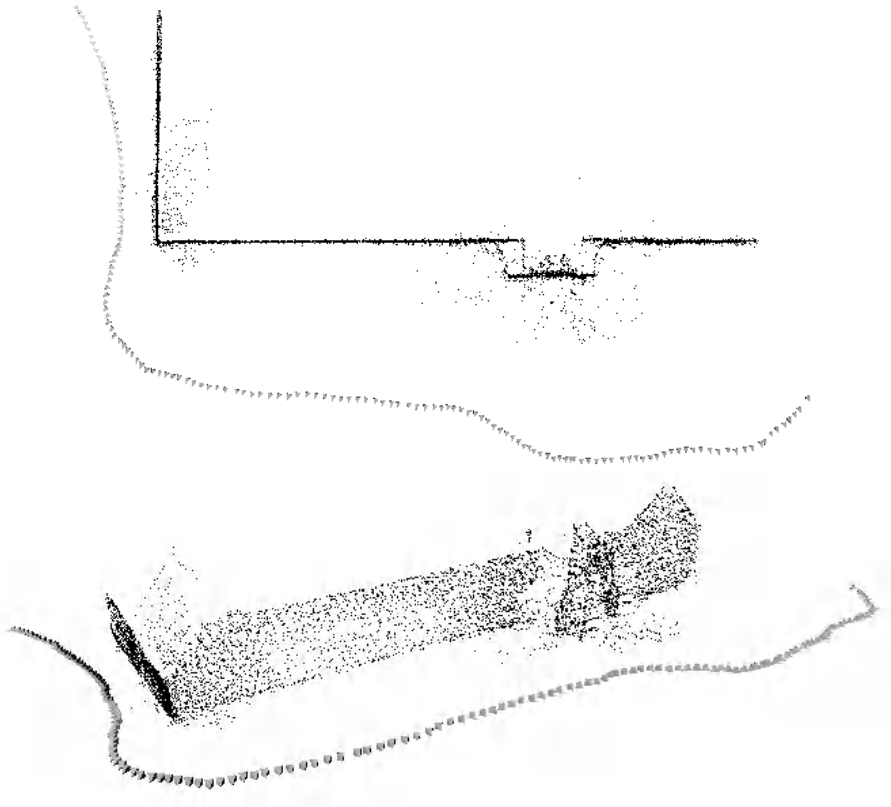
In Figure 4 the GRIC values are given for **F** and **H** as well as for **PPP** and **HH**. It can clearly be seen that –besides dealing with additional ambiguities– the triplet based analysis in general provides more discriminant results. It is also interesting to note that triplet 34-35-36 is clearly indicated as containing sufficiently general structure for the triplet-based approach while the pair-based approach marginally prefers to use the plane based model. The USaM approach reconstructs the structure for this triplet (including some points seen in the background of the lower left picture of Figure 3) and successfully integrates them with the rest of the recovered structure and motion.

The second sequence consists of 150 images of an old *farmhouse*. It was recorded with the same camera as the first sequence. In Figure 6 the GRIC values are plotted and for some of them the corresponding images are shown. As can be seen the approach successfully discriminates between the planar parts and the others. In Figure 7 the computed



**Fig. 6.** Some images of the *farmhouse* sequence together with GRIC(**PPP**) (solid/black line) and GRIC(**HH**) (dashed/blue line).

structure and motion is shown. In Figure 8 some views of a dense textured 3D model are shown. This model was obtained by computing some depth maps using a stereo algorithm and the obtained metric structure and motion. Note that the whole approach from image sequence to complete 3D model is fully automatic.



**Fig. 7.** Combined structure and motion for the whole *farmhouse* sequence.

## 9 Conclusion

In this paper we have presented an approach that successfully deals with dominant planes in uncalibrated structure and motion recovery. This is an important problem that limited the practical applicability of uncalibrated approaches, especially in man-made environments. The solution proposed in this paper yields very good results on real image sequences. The approach uses the Geometric Robust Information Criterion to detect if features seen in common by three views are all in a plane. Subsequences containing sufficiently general structure are reconstructed and then extended with the planar



**Fig. 8.** Textured 3D model of the *farmhouse*

parts. A new linear self-calibration algorithm couples the intrinsics between the different subsequences so that even for short sequences good results can be obtained. Once the reconstruction has been upgraded to metric, the pose is estimated for the cameras observing planar parts and the reconstructions for the different subsequences are assembled. Finally a global bundle adjustment provides an optimal estimate of both structure and motion. A key factor for the success of the proposed approach is the consistent use of robust maximum likelihood estimation through efficient bundle adjustment and robust estimation (i.e. RANSAC) at most of the stages of the computations.

**Acknowledgment.** Marc Pollefeys is a post-doctoral fellow of the Fund for Scientific Research - Flanders (Belgium). The financial support of the FWO project G.0223.01 and the IST projects ATTEST and INVIEW are also gratefully acknowledged.

## References

1. A. Bartoli and P. Sturm, "Constrained Structure and Motion from N Views of a Piecewise Planar Scene", VAA'01 - *In Proceedings of the International Symposium on Virtual and Augmented Architecture*, Dublin, Ireland, pp. 195-206, June 2001.
2. P. Beardsley, P. Torr, and A. Zisserman. "3D model acquisition from extended image sequences". In *Proc. European Conf. on Computer Vision*, LNCS 1064, Vol. 2, Springer-Verlag, pages 683-695, 1996.
3. O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 563-578, 1992.
4. O. Faugeras, Q.-T. Luong and S. Maybank. "Camera self-calibration: Theory and experiments", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 321-334, 1992.



5. M. Fischler and R. Bolles, "RANdom SAMpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography", *Commun. Assoc. Comp. Mach.*, 24:381-95, 1981.
6. A. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences", *Computer Vision – ECCV'98*, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, 1998. pp.311-326, 1998.
7. R. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *International Journal of Computer Vision*, Vol.13, No.3, 1994, pp. 331-356.
8. C. Harris and M. Stephens, "A combined corner and edge detector", *Fourth Alvey Vision Conference*, pp.147-151, 1988.
9. R. Hartley, R. Gupta, T. Chang, "Stereo from uncalibrated cameras". In *Proc. Conf. on Computer Vision and Pattern Recognition*, 1992.
10. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
11. A. Heyden and K. Åström, "Euclidean Reconstruction from Constant Intrinsic Parameters" *Proc. 13th International Conference on Pattern Recognition*, IEEE Computer Soc. Press, pp. 339-343, 1996.
12. W. Hofmann. *Das problem der "Gefährlichen Flächen" in Theorie und Praxis -Ein Beitrag zur Hauptaufgabe der Photogrammetrie*. PhD Thesis, Fakultät für Bauwesen, Technische Universität München, Germany, 1953.
13. F. Kahl, B. Triggs, K. Åström, "Critical Motions for Auto-Calibration When Some Intrinsic Parameters Can Vary", *Journal of Mathematical Imaging and Vision* 13,131-146,2000.
14. Y. Liu, H.-T. Tsui and A. Heyden, "3D Reconstruction of Buildings from an Uncalibrated Image Sequence – A Scene Based Strategy", *Proc. Virtual and Augmented Architecture (VAA'01)*, pp. 231–242, Springer-Verlag, 2001.
15. D. Nister, *Automatic Dense Reconstruction from Uncalibrated Video Sequences*, Ph. D. dissertation, Dept. of Numerical Analysis and Computing Science, KTH Stockholm, 2001.
16. M. Pollefeys and L. Van Gool, "Stratified Self-Calibration with the Modulus Constraint", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 21, No.8, pp.707-724, 1999.
17. M. Pollefeys, R. Koch and L. Van Gool. "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters", *International Journal of Computer Vision*, 32(1), 7-25, 1999.
18. M. Pollefeys, *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*, Ph.D. Thesis, ESAT-PSI, K.U.Leuven, 1999.
19. C. Rother and S. Carlsson. "Linear Multi View Reconstruction and Camera Recovery", *Proc. Eight IEEE International Conference on Computer Vision*, pp. 42-49 ,2001.
20. P. Sturm. "Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1100-1105, 1997.
21. P. Torr, *Motion Segmentation and Outlier Detection*, PhD Thesis, Dept. of Engineering Science, University of Oxford, 1995.
22. P. Torr. "An assessment of information criteria for motion model selection". In *CVPR97*, pages 47–53, 1997.
23. P. Torr, A. Fitzgibbon and A. Zisserman, "The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences", *International Journal of Computer Vision*, vol. 32, no. 1, pages 27-44, August, 1999.
24. B. Triggs, "The Absolute Quadric", *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 609-614, 1997.

25. B. Triggs, "Autocalibration from planar scenes", *Computer Vision – ECCV'98*, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, pp 89-105, 1998.
26. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. "Bundle adjustment – A modern synthesis". In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.
27. Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence Journal*, Vol.78, pp.87-119, October 1995.

# A Bayesian Estimation of Building Shape Using MCMC

A. R. Dick<sup>1</sup>, P. H. S. Torr<sup>2</sup>, and R. Cipolla<sup>1</sup>

<sup>1</sup> Engineering Department  
Cambridge University, Cambridge, UK  
{ard28, cipolla}@cam.ac.uk  
<http://www-svr.eng.cam.ac.uk/~ard28/>

<sup>2</sup> Microsoft Research, 7 JJ Thomson Avenue,  
Cambridge, CB3 0FB, UK  
philtorr@microsoft.com  
<http://research.microsoft.com/~philtorr/>

**Abstract.** This paper investigates the use of an implicit prior in Bayesian model-based 3D reconstruction of architecture from image sequences. In our previous work architecture is represented as a combination of basic primitives such as windows and doors etc, each with their own prior. The contribution of this work is to provide a global prior for the spatial organization of the basic primitives. However, it is difficult to explicitly formulate the prior on spatial organization. Instead we define an implicit representation that favours global regularities prevalent in architecture (e.g. windows lie in rows etc.). Specifying exact parameter values for this prior is problematic at best, however it is demonstrated that for a broad range of values the prior provides reasonable results. The validity of the prior is tested visually by generating synthetic buildings as draws from the prior simulated using MCMC. The result is a fully Bayesian method for structure from motion in the domain of architecture.

## 1 Introduction

Many algorithms (e.g. [1, 3, 16, 14]) have been developed for inferring 3D structure from a set of 2D images. A review of the state of the art in this area can be found at [17]. However there are often cases in which image information is ambiguous or misleading, such as in areas of homogeneous or repeated texture. In such cases extra information is required to obtain a model of the scene.

In the past, dense stereo algorithms have used heuristics favouring “likely” scenarios such as regularization or smoothing in an attempt to resolve these ambiguities (e.g. [5, 16]), but in general these are unsatisfactory (for instance, the smooth surface assumption is violated at occlusion boundaries). It is our belief that maximum likelihood estimates (even regularized) of structure have progressed as much as they are able, and that further research in this area will yield negligible or arguable benefit. Our approach to structure from motion is to develop generic methods to exploit domain-specific knowledge to overcome these ambiguities. This has been successfully done for other 3D reconstruction domains, e.g. heads [9, 18], bodies [15].

Within this paper we explore the reconstruction of generic buildings from images, using strong prior knowledge of building form provided by architects, this is most naturally done in a Bayesian framework. The Bayesian framework provides a rational

method for incorporating prior information into the estimation process [12]. However in complicated scenarios such as the modelling of architecture, there still remains two problems to be resolved. Whilst Bayes provides the basic laws for manipulating probabilities, we still need to resolve the problem of parameterization, and once the problem is parameterized choose the best algorithm to optimize the parameters.

Structure is represented as a collection of planes (corresponding to walls) and primitives (representing windows, doors and so on). Each primitive is defined by several parameters, as listed in Table 1. The advantages of this model-based approach are that it enables the inference of scene structure and geometry where evidence from the images is weak, such as in occluded regions or areas of homogeneous texture, and that it provides an interpretation of the scene as well as its geometry and texture [8]. The representation of the scene as a set of planes and primitives is useful for reasoning about the scene during reconstruction and for subsequent rendering and manipulation of the model. The compactness of the representation also makes recovery of structure and motion more reliable, as demonstrated in [19].

In previous work [8] a framework was defined for model based structure from motion for buildings. In this framework an algorithm for estimating a maximum a posteriori (MAP) estimate of the model based on priors and image likelihood measures was proposed (this is summarized in Section 2). However the spatial prior used in this work applied only to the parameters of each individual primitive, thus ignoring information about their spatial juxtaposition (for instance, that windows are likely to occur in rows and columns). In this paper the spatial prior is expanded to include this sort of information.

The form which the spatial prior should take is far from obvious. Ideally it should admit all plausible buildings while excluding those which are for practical or aesthetic reasons implausible. However the plausibility of a structure can in general only be verified by manual inspection. Thus a crucial step in the formulation of the prior is to test it by drawing sample buildings from it and checking that they appear reasonable. However even with expert knowledge, it is very difficult to explicitly represent the probability density function (pdf) of a suitable prior. What is somewhat easier to do is to express the prior as a scoring function that favours particular configurations, such as windows in rows. One approach is to use a scoring function suggested by an expert and then draw samples from the implicitly defined pdf using an MCMC algorithm. If the samples drawn look like reasonable buildings then the prior must be close to the true prior.

This raises the question of just how close to the true prior our estimate must be to generate reasonable looking models. To answer this empirically the scoring function is varied both on a small scale and a large scale, and the effect on models generated from the prior, and reconstructions obtained using the prior and an image sequence, is observed.

The paper is organized as follows: Section 2 defines an architectural model as a collection of wall planes containing parameterized shapes, and establishes a framework for optimizing it. In Section 3 the spatial prior is discussed and the MCMC algorithm is used to simulate samples from it. Section 4 then presents some reconstructions based

on this prior, and demonstrates the effect of varying the prior on reconstruction. The paper concludes with discussion and ideas for future work in Section 6.

## 2 Problem Formulation

This section briefly recapitulates previous work [8] on the definition of a model for architecture and an algorithm for optimizing it. The projection matrices are initially recovered using point matching and robust methods as described in [2, 20]. With the projection matrices recovered the 3D structure of the scene must be parameterized. An architectural model is formulated as a number of base planes (generally walls), each of which contains a number of offset primitives such as doors and windows. Also modelled are the height of the apex of the roof, and the number of floors in the building, as these will affect the window layout. The model  $\mathbf{M}$  therefore contains parameters  $\theta = \{n, \theta_L, \theta_S, \theta_T, \theta_G\}$ , where  $n$  is the number of primitives in the model,  $\theta_L$  identifies the type of each primitive,  $\theta_S$  are structure parameters which define its shape,  $\theta_T$  are texture parameters describing its appearance, and  $\theta_G$  are global parameters describing such things as the number of floors and the style of the building (e.g. Classical, Gothic). The types of primitive available and their shape parameters are given in Table 1. The texture parameters are intensity variables  $i(\mathbf{x})$  (between 0 and 255) defined at each point  $\mathbf{x}$  on a regular 2D grid covering the model surface<sup>1</sup>.

Our choice of primitive reflects the scale of detail in the model. The model is designed to be used with photographs of architecture taken from ground level. Therefore it models a level of detail consistent with these viewpoints; for instance doors and windows are modelled in addition to the walls of each building. However finer levels of detail, such as the location of individual bricks, door handles and fine ornamentation are not modelled. Similarly little attention is paid to modelling roofs (in fact they are only modelled as a simple pyramid), as most images taken from ground level include very little if any information about the roof structure.

To recover the architectural model we want to maximize

$$\begin{aligned}
 \Pr(\mathbf{M}\theta|\mathbf{DI}) &\propto \Pr(\mathbf{D}|\mathbf{M}\theta\mathbf{I}) \Pr(\mathbf{M}\theta\mathbf{I}) \\
 &= \Pr(\mathbf{D}|\mathbf{M}\theta\mathbf{I}) \Pr(\theta|\mathbf{MI}) \Pr(\mathbf{MI}) \\
 &= \Pr(\mathbf{D}|\mathbf{M}\theta_L\theta_S\theta_T\theta_G\mathbf{I}) \Pr(\theta_L\theta_S\theta_T\theta_G|\mathbf{MI}) \Pr(\mathbf{MI}) \\
 &= \Pr(\mathbf{D}|\mathbf{M}\theta_L\theta_S\theta_T\theta_G\mathbf{I}) \Pr(\theta_T|\theta_L\theta_G\mathbf{MI}) \\
 &\quad \Pr(\theta_S|\theta_L\theta_G\mathbf{MI}) \Pr(\theta_L|\theta_G\mathbf{MI}) \Pr(\theta_G|\mathbf{MI})
 \end{aligned} \tag{1}$$

where  $\mathbf{D}$  is the available data (the images),  $\mathbf{I}$  denotes prior information (the camera calibration and the estimated wall planes), and

- $\Pr(\mathbf{D}|\mathbf{M}\theta_L\theta_S\theta_T\theta_G\mathbf{I})$  is the likelihood of the images given a complete specification of the model. This is determined by the deviation of image intensities from those predicted by the texture parameters.

<sup>1</sup> This allows us to specify the model to super resolution; however this aspect is not dealt with in this paper.

**Table 1.** Some primitives available for modelling classical architecture. Parameters in brackets are optional. The parameters are defined as follows:  $x$ :  $x$  position;  $y$ :  $y$  position;  $w$ : width;  $h$ : height;  $d$ : depth;  $a$ : arch height;  $b$ : bevel (sloped edge);  $dw$ : taper of pillars, buttresses. The NULL model is simply a collection of sparse triangulated 3D points.  $\mathcal{M}_0$  is reserved as the background model (generally a wall).

$\theta_L^i$	Description	Parameters
$\mathcal{M}_1$	Window	$x, y, w, h, d, (b), (a)$
$\mathcal{M}_2$	Door	$x, y, w, h, d, (b), (a)$
$\mathcal{M}_3$	Pediment	$x, y, w, h, d$
$\mathcal{M}_4$	Pedestal	$x, y, w, h, d$
$\mathcal{M}_5$	Entablature	$x, y, w, h, d$
$\mathcal{M}_6$	Column	$x, y, w, h, d, dw$
$\mathcal{M}_7$	Buttress	$x, y, w, h, d, dw$
$\mathcal{M}_8$	Drain pipe	$x, w, h$
$\mathcal{M}_9$	Floor	$y$
$\mathcal{M}_{10}$	Roof	$h$
$\mathcal{M}_{11}$	NULL	$x_1 \dots x_n, y_1 \dots y_n, z_1 \dots z_n$

- $\Pr(\theta_T | \theta_L \theta_G \mathbf{MI})$  is the probability of the texture parameters. This is evaluated using learnt models of appearance, such as the fact that windows are often dark with intersecting mullions (vertical bars) and transoms (horizontal bars), or that columns contain vertical fluting.
- $\Pr(\theta_L | \theta_G \mathbf{MI})$  is the prior probability for each type of primitive. It is used to specify the relative frequency with which primitive types occur, e.g. that windows are more common than doors, or that buttresses appear frequently in Gothic architecture.
- $\Pr(\theta_S | \theta_L \theta_G \mathbf{MI})$  is a prior on shape. The formulation and validation of this prior is described in more detail in Section 3.

The global parameters  $\theta_G$  are generally given rather than estimated (for instance, the style of the model is specified manually) and hence the probability  $\Pr(\theta_G | \mathbf{MI})$  is fixed at 1 for the given parameters, and 0 elsewhere.

## 2.1 Obtaining an Initial Model Estimate

The input to our system is an uncalibrated sequence of 3–6 images, in which corner and line features are automatically detected and matched as in [3] to estimate the structure of the building and the motion of the cameras. This reconstruction is then segmented into planes to obtain an initial estimate of the position and orientation of each wall in the building [7].

Ideally the combined  $\theta_T$ ,  $\theta_S$  and  $\theta_L$  parameter space would then be searched for MAP parameter values. However each primitive may contain thousands of texture parameters, so only the  $\theta_S$  and  $\theta_L$  parameter spaces are searched. This is carried out in two steps: an initial search based on an approximate single image likelihood function locates likely values for a subset  $\theta_{S1} = \{x, y, w, h\}$  of the shape parameters, while

the remaining shape parameters  $d, a, b, dw$  are set to zero. These are then used to seed searches in the full parameter space using the complete likelihood function [8]. Models found using this method are subsequently used as seed points for the MCMC algorithm described in this paper.

### 3 The Shape Prior

In this section an architectural shape prior  $\Pr(\theta_S | \theta_L \text{MI})$  is defined and assessed using a Markov Chain simulation. Ideally this prior should encode information about:

- The scale of each primitive. For instance a door should be tall enough for a person to comfortably walk through. Scale priors can only be used when the absolute scale of the model is known.
- The shape of each primitive. For instance columns are likely to be long and thin, while pedestals are more broad and flat.
- The alignment of primitives. For instance windows are likely to occur in rows corresponding to the floors of a building.
- Other spatial relations such as symmetry about a vertical axis.

It is extremely difficult to explicitly formulate a prior pdf to meet this set of desiderata. When it is required to draw samples from a pdf  $\Pr(\theta)$  which cannot be explicitly defined, a common technique is to simulate the drawing of samples using a Markov process defined on the parameters  $\theta$ . The Markov process is chosen so that over time (as  $t \rightarrow \infty$ ) its transition probability  $\Pr(\theta_t | \theta_{t-1})$  converges to the desired distribution  $\Pr(\theta)$ . Therefore as the number of iterations increases, the values of  $\theta$  visited by the Markov process mimics independent draws from  $\Pr(\theta)$  with increasing accuracy. The group of algorithms which operate on this principle are known as Markov Chain Monte Carlo (MCMC) algorithms [10] (Monte Carlo refers to the fact that Markov processes are seeded at many points in parameter space).

There are a number of ways to generate a Markov process with the desired convergence properties. These include the Metropolis-Hastings class of algorithms, in which transitions or jumps are drawn from a user-defined jumping distribution  $J_t(\theta_t | \theta_{t-1})$  and the updated model is accepted or rejected based on a scoring function  $f(\theta)$ . In particular the Reversible Jump [11] Metropolis-Hastings algorithm is used in this paper, as it allows jumps between parameter spaces of varying dimension, as required when primitives are added or removed from the model. This algorithm is summarized in Algorithm 1.

The behaviour of the Reversible Jump MCMC algorithm depends largely on the choice of both the scoring function and the jumping distribution. These are the subjects of the following two sections.

#### 3.1 The Scoring Function

The scoring function contains terms relating to the scale, shape and alignment of primitives:

$$f_{prior}(\theta) = f_{scale}(\theta) + f_{shape}(\theta) + f_{align}(\theta) + f_{sym}(\theta) \quad (3)$$

**Algorithm 1** Reversible Jump MCMC algorithm.

---

Draw an initial point  $\theta_0$  from a starting distribution  $\text{Pr}_0(\theta)$ .

**for**  $t = 1..T$  **do**

    Draw candidate point  $\theta_*$  from the jumping distribution  $J_t(\theta_*|\theta_{t-1})$ .

    Calculate the ratio

$$r = \frac{f(\theta_*)J_t(\theta_{t-1}|\theta_*)}{f(\theta_{t-1})J_t(\theta_*|\theta_{t-1})} \quad (2)$$

    Set  $\theta_t = \theta_*$  with probability  $\min(r, 1)$ , otherwise set  $\theta_t = \theta_{t-1}$

**end for**

---

The shape and scale terms apply only to individual primitives. In this paper the shape and scale components of the scoring function are given by simple functions, some of which are listed in Table 3. The purpose of these components is mainly to disqualify implausible primitives such as doors which are too thin or short to be practical, windows which extend between floors of the building, or buttresses which do not reach the ground.

The component of the scoring function for the alignment of shapes into rows and columns computes the deviation of the shapes from an aligned grid containing  $R$  rows and  $C$  columns. It is defined as

$$f_{align}(\theta) = \sum_{r=1}^R [\text{Var}(\mathbf{t}_r) + \text{Var}(\mathbf{b}_r) + \text{Var}(\mathbf{r}_r - \mathbf{l}_r)] \quad (4)$$

$$+ \sum_{c=1}^C [\text{Var}(\mathbf{l}_c) + \text{Var}(\mathbf{r}_c) + \text{Var}(\mathbf{t}_c - \mathbf{b}_c)] \quad (5)$$

where  $\mathbf{t}_r, \mathbf{b}_r, \mathbf{l}_r, \mathbf{r}_r$  are the top, bottom, left and right coordinates of the primitives belonging to row  $r$  and  $\mathbf{t}_c, \mathbf{b}_c, \mathbf{l}_c, \mathbf{r}_c$  are similarly defined for primitives belonging to column  $c$ . The function  $\text{Var}(\mathbf{x})$  gives the variance of the elements of  $\mathbf{x}$ . When a wall contains 0 or 1 primitives, it is assigned a fixed score of high variance which encodes a preference for more than one window per wall if the wall is large enough to accommodate it.

The symmetry component of the scoring function is maximized when a row is exactly centred on a wall, and decays quadratically:

$$f_{sym}(\theta) = \sum_{r=1}^R [(l_r - \mathbf{l}) - (r_r - \mathbf{r})]^2 \quad (6)$$

where  $l_r$  is the leftmost point of row  $r$ ,  $r_r$  is the rightmost point of row  $r$ , and  $\mathbf{l}$  and  $\mathbf{r}$  are the left and right coordinates of the wall on which the row appears. The symmetry function is applied only to rows as it was found that columns of shapes are not generally vertically symmetric on a wall.



### 3.2 The Jumping Distribution

As well as a scoring function, an MCMC algorithm requires the specification of a jumping distribution  $J(\theta_t|\theta_{t-1})$ . The jumping distribution is a mixture of several types of jump, which are listed in Table 2.

There are a number of issues bearing on the choice of jump types to use:

- A building should always be a closed structure with walls which intersect at near right angles. Therefore the add/remove/modify wall jumps actually add, remove or modify a closed set of perpendicular walls to the model, effectively adding or removing a room from the reconstruction while maintaining closure.
- For efficiency, it should be easy to sample from the jumping distributions. Each jump should also traverse a significant distance in parameter space, and have a high acceptance rate. Therefore simple jump types which are likely to generate a more probable model should be used.
- Reversible jump MCMC requires that jumps be symmetric; that is, after jumping from  $\{\mathbf{M}_1, \theta_1\}$  to  $\{\mathbf{M}_2, \theta_2\}$  it should always be possible to return to  $\{\mathbf{M}_1, \theta_1\}$  in a single jump. More precisely, given a jump type  $\mathcal{J}_{i1}$  which moves from the parameter space  $\theta_1$  to  $\theta_2$  based on the values of  $\theta_1$  and some extra random variables  $\phi_1$ , there must be a reverse jump type  $\mathcal{J}_{i2}$  which moves from  $\theta_2$  to  $\theta_1$  based on  $\theta_2$  and  $\phi_2$ , where the dimension of  $\theta_1 \oplus \phi_1$  equals that of  $\theta_2 \oplus \phi_2$ . This poses difficulties when considering jumps such as aligning a group of shapes into a regular column. To maintain reversibility, a related jump must be included which can take a column of shapes and perturb each shape so that if the column-aligning jump is applied again, the same column configuration would result.

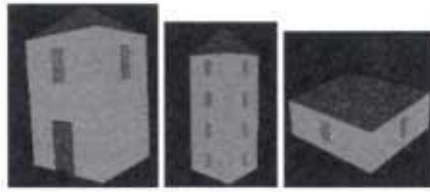
**Table 2.** Jump types available to MCMC algorithm. The parameter  $n$  identifies a single primitive to which the jump is applied. Jump  $\mathcal{J}_{10}$  "regularises" a row or column of shapes by aligning all shapes, making them the same size, and evenly spaced. Jump  $\mathcal{J}_{11}$  is similar but also positions the row so that it is centred in the wall.

Jump type	Description	Parameters
$\mathcal{J}_1$	Add shape	$\mathcal{M}_i, x, y, w, h$
$\mathcal{J}_2$	Remove shape	$n$
$\mathcal{J}_3$	Modify shape	$n, x, y, w, h$
$\mathcal{J}_4$	Add wall	$n, w, h$
$\mathcal{J}_5$	Remove wall	$n$
$\mathcal{J}_6$	Modify wall	$w, h$
$\mathcal{J}_7$	Add window row/col	$n$
$\mathcal{J}_8$	Remove window row/col	$n$
$\mathcal{J}_9$	Modify window row/col	$n$
$\mathcal{J}_{10}$	Regularise window row/col	$n$
$\mathcal{J}_{11}$	Symmetrise window row	$n$
$\mathcal{J}_{12}$	Perturb window row/col	$n, x, y, w, h$

### 3.3 Verifying the Shape Prior

Having specified a scoring function and jumping distribution, the Reversible Jump MCMC algorithm can be used to simulate drawing independent samples from the shape prior that they define. It is important to sample from the shape prior to verify that it generates plausible buildings, and that is not too restrictive, in which case it would produce very similar buildings.

In the following experiment, a total of 9 Markov processes are seeded from one of 3 starting points, shown in Figure 1: a square "hut", a tower, or a bungalow shape. Each of the seed models has one wall containing a door at ground level, and each wall contains 0 or 1 windows. Each Markov chain is iterated for 2000 jumps. After this period, samples



**Fig. 1.** "Hut", "tower" and "bungalow" seed points for the MCMC algorithm.

are drawn at random from each chain and displayed in Figure 2. Samples are drawn at random intervals rather than from consecutive iterations; consecutive iterations tend to be correlated as most jumps entail only a minor change to the model. The samples are displayed as a city of buildings to illustrate both the inter-building variation of the results and the plausibility of each individual structure.



**Fig. 2.** Collection of Classical style buildings generated from the shape prior.

## 4 Results for a Single Wall

Some results of the MCMC algorithm are now shown for a single wall. The scoring function is altered to include image information by adding a likelihood term, so that the

complete score is now given by

$$f(\theta) = \lambda f_{prior}(\theta) + \sum_{i(\mathbf{x})} \sum_{j=1}^m \left( \frac{i(\mathbf{x}^j) - i(\mathbf{x})}{\sigma} \right)^2 \quad (7)$$

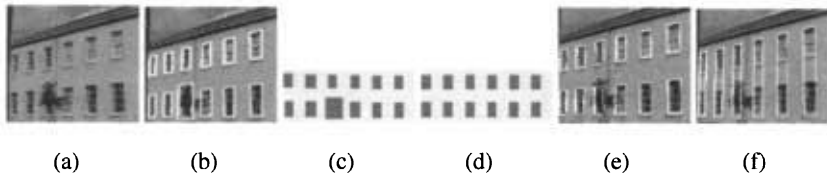
where  $i(\mathbf{x}^j)$  is the projection of the texture parameter  $i(\mathbf{x})$  onto the  $j$ th image,  $\sigma$  is an image variance parameter and  $\lambda$  is a relative scale factor. The global parameter set  $\theta_G$  is reduced to contain simply a style parameter, as the number of floors can be deduced from the images.

Starting points for the algorithm are generated from the results of a previously developed algorithm for finding a MAP model estimate, described in Section 2.1. It is first tested on images of the Downing library, one of which is shown in Figure 3(a), which contains strong information and complies with our priors on regularity and shape. Because the our old algorithm incorporated priors only on individual primitives, the windows estimated, although approximately correct, are slightly misaligned. Furthermore the window obscured by the tree is incorrectly fitted. However the new global shape prior described in this paper significantly improves matters: After running the MCMC algorithm for 2000 iterations, with  $\lambda = 10^4$  and  $\sigma = 10$ , the prior has overridden local likelihood maxima and the MAP model is one in which the windows are properly aligned, even where the window is occluded in some images by a tree (Figure 3(d)). To test the sensitivity of this result to the choice of prior, a Gothic shape prior is substituted for the original Classical shape prior. Although the Gothic prior favours narrower and more arched windows, the same result is obtained (Figure 3(e)). In Figure 3(f) a very different prior is proposed in which very tall narrow windows are strongly favoured (Table 3). This results in a different interpretation of the scene in which vertically aligned windows are merged.

**Table 3.** Relevant shape priors.  $H$  and  $W$  are the height and width of the wall containing the primitive.  $G(\mu, \sigma)$  is a Gaussian distribution with mean  $\mu$  and standard deviation  $\sigma$ .

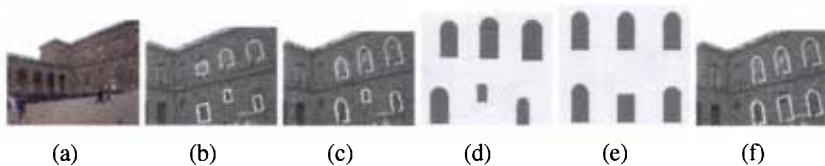
Primitive	Param.	Score
Window (Classical)	$h/w$	$-(G(0.75, 0.25) + G(1.0, 0.25) + G(1.25, 0.25) + G(1.5, 0.25) + G(1.75, 0.25) + G(2.0, 0.25))$
Window (Gothic)	$h/w$	$-(G(2.0, 0.25) + G(2.5, 0.25))$
Window (Narrow)	$h/w$	$-(G(4.0, 1.0) + G(6.0, 1.0))$
Window	$y/H$	0, if $0.1 \leq y/H \leq 0.9$ . Else LARGE.
Window	$x/W$	0, if $0.1 \leq x/W \leq 0.9$ . Else LARGE.
Window	$h/d$	0, if $-0.2 \leq h/d \leq 0.2$ . Else LARGE.

The next image sequence, of the Palazzo Pitti, one image of which is shown in Figure 4(a), is more challenging. The images were taken on a rainy day, and the window



**Fig. 3.** (a) One image of the wall. (b) Model obtained using only Classical single primitive prior. (c) Frontal view of this model (windows are darker, background is light). (d) Model obtained using MCMC and Gothic shape prior. (e) Same model, projected onto image. (f) Incorrect model obtained using prior favouring very narrow windows.

texture is indistinct from wall texture—the windows and wall are a similar colour, and the brick texture of the wall is not easily distinguished from that of the windows. Due to the lack of information in the images, the reconstruction obtained is dependent on whether a Classical or Gothic prior is used. Using Classical priors (Figure 4(b)), only the interior parts of the top windows are detected, whereas a Gothic prior (Figure 4(c)) detects the surrounding arch structure. Applying the full set of shape priors with  $\lambda = 10^4$ ,  $\sigma = 10$  results in a MAP model where the windows are aligned despite this not being the case for the bottom row of windows. To prevent this from occurring,  $\lambda$  can be reduced to 1 in which case the prior has no significant effect on the model.



**Fig. 4.** (a) One image of the Palazzo Pitti. (b) Model obtained using Classical priors. (c) Model obtained using Gothic priors. (d) Primitives before MCMC. (e) After MCMC. (f) Projected onto image.

## 5 Building Results

When jump types involving wall plane parameters are included in the MCMC algorithm, closure of the building is enforced and the reconstruction converges to a symmetric model such as that shown in Figure 5. The texture for this model is cut and pasted from areas of the image identified as a wall, window, columns and so on, and the same texture sample is used for every instance of a type of primitive. Another feature

of using an MCMC algorithm to sample the posterior is that as well as having a MAP model estimate, other probable samples can also be examined. This is useful for identifying ambiguities in the reconstruction. Four of the more marked ambiguities present in this model are shown in Figure 5 (i)–(l).

The operation of this algorithm is shown in Figure 6 for the Trinity Chapel sequence. Note that the entire model is obtained from only 3 images. Although the model is not completely accurate in areas which are not visible in the images, it is a plausible structure, and is obtained automatically except for the prior specification of the structure as being Gothic, and the restriction of the variety and shape of primitives this entails. The width of each part of the building is obtained from the average size of the window or door primitives on visible walls—each segment of the building is made wide enough to accommodate one window of height and width equal to the average height and width of the visible windows, with spacing to either side equal to half the window width. In the absence of image information, this seems a reasonable assumption to make and produces generally plausible architectural models.

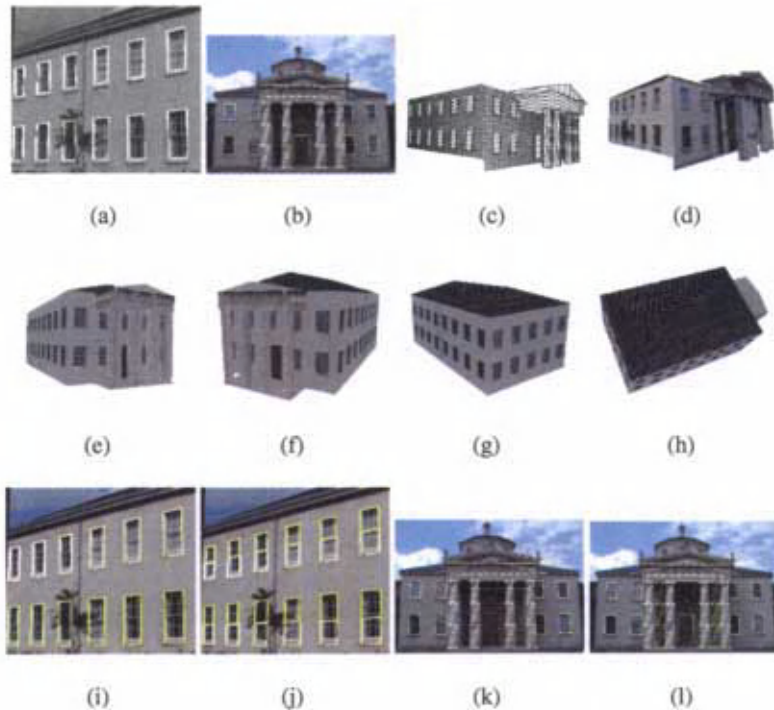
### 5.1 Comparison with Ground Truth

It is difficult to directly compare the results of this algorithm with previous methods, as no other method solves quite the same problem as this one. In the absence of experimental results with which to compare it, some ground truth measurements were taken from the Downing College library, reconstructed in Figure 5.

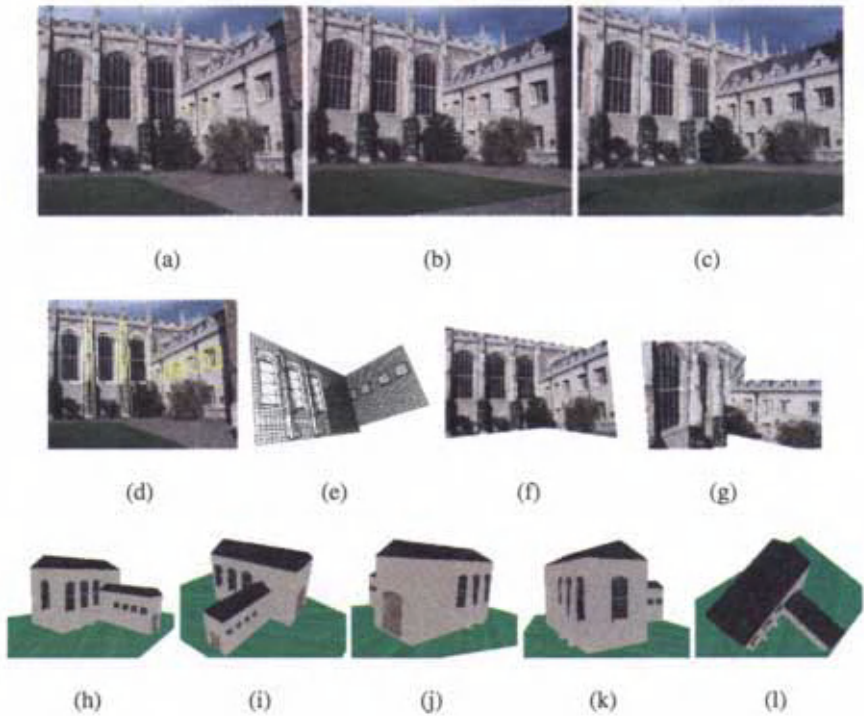
The height, width and depth of a set of windows belonging to this building were measured with a tape measure. The resulting lengths are shown in Figure 7. Because the absolute scale of the model is unknown, only ratios of lengths are compared to ground truth values. It is assumed that there is a  $\pm 1\text{cm}$  error in each measurement. In Table 4, a comparison of the corresponding model values and ground truth measurements is given. The uncertainty in the model values is based on the resolution of the grid of texture parameters on each plane. It can be seen that the ratios of window height to width, and width to depth, are recovered to within the accuracy bounds

**Table 4.** Comparison of ratios of window height ( $h$ ) to width ( $w$ ), width to depth ( $d$ ) wall–column separation ( $d_2$ ) to window width and the circumference of a column ( $c$ ) to window width. The upper and lower bounds are based on  $\pm 1\text{cm}$  accuracy for ground truth measurements, except for the circumference of the base of the column, which is measured to  $\pm 10\text{cm}$ . The accuracy of the model measurements is limited to the resolution of the texture parameters.

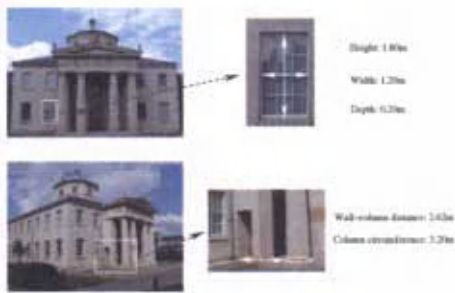
Ratio	Ground truth		Model	
	Lower	Upper	Lower	Upper
$h/w$	1.48	1.52	1.50	1.64
$w/d$	5.67	6.37	5.00	7.40
$d_2/w$	2.22	2.24	2.18	2.28
$c/w$	2.56	2.77	2.74	3.00



**Fig. 5.** (a) MAP model of side wall of Downing library, after 2000 MCMC iterations using the Classical prior. (b) Front wall. Both front and back faces of primitives are drawn, hence the pair of triangles and rectangles for the pediment and entablature. (c)-(d) 3D rendering of MAP Downing model, obtained without using add/remove/delete wall jumps. The textures shown on the model are automatically extracted from the images which are most front-on to each plane. (e)-(h) Four views of the completed model of Downing library, with extra walls added. Even though only two walls are visible, a complete building has been modelled using symmetry. Wall, window, roof and column textures are sampled from the images and applied to the appropriate primitives. (i)-(l) Some ambiguities in the Downing model, chosen from the 20 most probable models visited by the MCMC process. (i) Window sills are included in the window primitives. (j) Windows are represented using two primitives each. (k) The door is omitted. (l) Extra columns are added in between the existing ones.



**Fig. 6.** (a)-(c) 3 original images of Trinity college courtyard. (d) MAP model primitives, super-imposed on image. (e) Wireframe MAP model. (f)-(g) 3D model with texture taken from images. (i)-(m) Five views of the completed model of the north-east corner of Great Court, Trinity College. Only two of the walls are visible in the images.



**Fig. 7.** Some measurements made of the Downing Library scene. The other windows in the scene are the same size as the one shown (to an accuracy of  $\pm 1\text{cm}$ ).



of the ground truth measurements. The error margins are generally greater for model measurements, which are constrained by the resolution of the images. Although high resolution images ( $1600 \times 1200$  pixels) were used for this model, it seems that even more resolution is required to precisely recover fine details such as the depth of each window. The distance from the column to the wall is also identified accurately, but the circumference of the column is slightly underestimated (although the error margins just overlap). The circumference of the column is quite difficult to measure precisely, due to the stonework on its outer rim. Therefore there is an uncertainty of  $\pm 10\text{cm}$  associated with its measurement—this is derived from the fact that there are 20 partitions in the fluting around the base of the column, each of which can be measured to a precision of approximately  $\pm 0.5\text{cm}$ .

## 6 Conclusion

If structure from motion algorithms are to progress they must find ways of incorporating more and higher level prior information concerning the nature of the world. To effectively use this information, recognition of what we are observing will play a crucial rôle. Recovering structure from visual input alone is highly ill conditioned, thus is it envisaged that a robot of the future might carry many prior models in its head and recognize which class of priors is appropriate to reduce the ambiguity in resolving a particular scene. Thus classic geometric structure from motion becomes a blend of learning, classification and geometry. This paper has presented a framework for representing one such spatial prior for the case of architecture. Samples are generated from the prior and shown to be reasonable instances of genuine buildings. A Bayesian framework is a natural way to effectively use the prior information to enhance 3D reconstruction in a variety of ways. Future work includes the parametrization of texture in a similar way to shape, i.e. using only a few texture parameters per primitive, which would also result in a super resolution of the images. As previously mentioned, the roof structure is currently modelled simply as a pyramid; however more complex roof models could improve the appearance of the model from elevated viewpoints.

The general philosophy underlying this paper is that the state of the art has been reached with existing structure from motion methods, and that the best route for progress is to combine structure from motion with recognition. This allows the use of strong prior models of shape, stronger than the markov random fields traditionally used. To some extent this has been done in the past with simple shape models e.g. [4, 6, 13], but these models possess only limited variability. Within this paper we have attempted to present a generic framework which could be used to optimize classes of objects that possess a much greater variability of shape, but that can be decomposed into a ‘lego kit’ of parameterizable parts.

**Software:** It is intended to release a Matlab SFM toolkit to illustrate some of the methods described, please check <http://research.microsoft.com/phillorr/>, as the release is aimed to coincide with ECCV.



## References

1. M.E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR00*, pages II:282–289, 2000. See also <http://city.lcs.mit.edu>.
2. P. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and Cipolla R., editors, *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695. Springer–Verlag, 1996.
3. P.A. Beardsley, A. Zisserman, and D.W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.
4. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE PAMI*, 23(6):681–685, 2001.
5. I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs. A maximum-likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.
6. D. Cremers, C. Schnoerr, and J. Weickert. Diffusion-snakes: Combining statistical shape knowledge and image information in a variational framework. In *1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001.
7. A.R. Dick, P.H.S. Torr, and R. Cipolla. Automatic 3d modelling of architecture. In *Proc. 11th British Machine Vision Conference (BMVC'00)*, pages 372–381, Bristol, 2000.
8. A.R. Dick, P.H.S. Torr, S. Ruffle, and R. Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Proc. IEEE International Conference on Computer Vision*, pages I:268–274, 2001.
9. P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2), July 1999.
10. W. Gilks, S. Richardson, and D. Spiegelhalter (editors). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London., 1996.
11. P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.
12. E. T. Jaynes. *Probability Theory: The Logic of Science*. Unpublished but available online at <http://bayes.wustl.edu/etj/prob.html>, 1996.
13. M. Leventon, E. Grimson, and O. Faugeras. Statistical Shape Influence in Geodesic Active Contours. In *CVPR*, pages 316–323, 2000.
14. D. Nister. Frame decimation for structure and motion. In *2nd European Workshop on 3D Structure from Multiple Images of Large Scale Environments (SMILE 2000)*, pages 17–34, 2000.
15. R. Plänkers and P. Fua. Articulated Soft Objects for Video-based Body Modeling. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.
16. M. Pollefeys, R. Koch, M. Vergauwen, and L. van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *1st European Workshop on 3D Structure from Multiple Images of Large Scale Environments (SMILE 1998)*, pages 139–155, 1998.
17. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002. Evaluation page <http://www.middlebury.edu/stereo/eval/>.
18. Y. Shan, Z. Liu, and Z. Zhang. Model based bundle adjustment with applications to face modeling. In *ICCV Vol 2*, pages 644–651. IEEE, 2001.
19. C.J. Taylor, P.E. Debevec, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *ACM SIGGraph, Computer Graphics*, pages 11–20, 1996.
20. P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int Journal of Computer Vision*, 24(3):271–300, 1997.

# Structure and Motion for Dynamic Scenes – The Case of Points Moving in Planes

Peter Sturm

INRIA Rhône-Alpes, 38330 Montbonnot, France,  
Peter.Sturm@inrialpes.fr,  
<http://www.inrialpes.fr/movi/people/Sturm>

**Abstract.** We consider dynamic scenes consisting of moving points whose motion is constrained to happen in one of a pencil of planes. This is for example the case when rigid objects move independently, but on a common ground plane (each point moves in one of a pencil of planes parallel to the ground plane). We consider stereo pairs of the dynamic scene, taken by a moving stereo system, that allow to obtain 3D reconstructions of the scene, for different time instants. We derive matching constraints for pairs of such 3D reconstructions, especially we introduce a simple tensor, that encapsulates parts of the motion of the stereo system and parts of the scene structure. This tensor allows to partially recover the dynamic structure of the scene. Complete recovery of structure and motion can be performed in a number of ways, e.g. using the information of static points or linear trajectories. We also develop a special self-calibration method for the considered scenario.

## 1 Introduction

Most existing works on structure and motion from images concentrate on the case of rigid scenes. The rigidity constraint allows to derive matching relations among two or more images, represented by e.g. the fundamental matrix or trifocal tensors. These matching tensors encapsulate the geometry/motion of the cameras which took the underlying images, and thus all the geometric information needed to perform 3D reconstruction. Matching tensors for rigid scenes can also be employed for scenes composed of multiple, independently moving objects [1,2], which requires however that enough features be extracted for each object, making segmentation, at least implicitly, possible.

Shashua and Wolf introduced the so-called homography tensors [9] – matching constraints that exist between three views of a planar scene consisting of independently moving points (each point being static or moving on a straight line). Basically, given correspondences of projections of such points in three images, the plane homographies between all pairs of images can be computed from the homography tensor, which would not be possible with only two images of the scene. It is important to note that this does not make any assumption about the camera’s motion, i.e. the camera is indeed allowed to move freely between image takings. So, this work is maybe the first that considers scenarios where *everything* is moving independently: the camera as well as any point in the scene <sup>1</sup>.

---

<sup>1</sup> Of course, if the camera were not moving, two images would be enough to do the job (the plane homography between them, for any plane, is intrinsically known – it is the identity).

Naturally, the question arises if there are other dynamic scenarios that might be interesting to examine. Wolf et al. considered the case of a rigid stereo system taking stereo pairs of a *threedimensional* scene consisting of points moving on straight lines, but independently from each other [11]. From each stereo pair, a 3D reconstruction of the current state of the scene can be obtained (a projective reconstruction if the cameras are not calibrated). Similarly to the above mentioned work on 2D homography tensors, the aim is now to determine 3D homographies between pairs of 3D reconstructions, that would allow to align them. If the stereo system were static, this would again be no problem: the searched for 3D homography is simply the identity transformation. In case of a *moving* stereo system however, Wolf et al. showed that there exist matching tensors, between three 3D reconstructions, representing the state of the scene at three different time instants. From these so-called join tensors, the 3D homographies between all pairs of 3D reconstructions can be recovered, and the reconstructions can be aligned. These 3D homographies represent in fact the stereo system's motions.

Other works along similar lines include that of Han and Kanade [3,4], who consider points moving with constant velocities (thus on linear trajectories), for the case of affine or perspective cameras. Wolf and Shashua [12] consider several dynamic scenarios, and derive matching constraints by embedding the problem representations in higher-dimensional spaces than e.g. the usual projective 3-space for rigid scenes.

The work presented in this paper is inspired by these works. We consider the following scenario: a moving stereo system taking 2D views of a 3D scene consisting of moving points, each point moving arbitrarily in what we call its *motion plane*. In addition, all motion planes are constrained to belong to the same pencil of planes. The most practical instance of this kind of scenario is the case where all motion planes are parallel to each other and, say, horizontal. This scenario covers for example all scenes where objects move on a common ground plane.

For each time instant considered, the stereo system gives a 3D view of the current state of the scene, which would be a projective reconstruction for example, if the system is uncalibrated. In this paper, we derive matching constraints that exist between such 3D views, and examine which amount of 3D motion and structure information can be recovered from the associated matching tensors. We show that there already exists a matching tensor between two 3D views, for two different time instants. This tensor is more or less the analogue to the fundamental matrix between pairs of 2D views. However, it does not allow full recovery of the stereo system's and the 3D points' structure and motion. Full recovery of these requires additional information, e.g. the knowledge that certain points are static, or that certain points move on linear trajectories (if three or more 3D views are available). In the latter case, the join tensors [11] may be applied, but in our more constrained scenario (pencil of motion planes), a simpler matching constraint exists, that can be estimated with fewer correspondences.

For the special case of parallel motion planes, we present a simple self-calibration method that overcomes singularities that exist without the knowledge of parallelism.

## 2 Background and Notation

We will both use standard matrix-vector notations, and tensor notation. In tensor notation, points are specified by superscripts, e.g.  $P^i$ . Transformations mapping points onto points,

have one superscript and one subscript, e.g.  $T_i^m$ . Mapping the point  $P$  by  $T$  gives a point  $Q$  with  $Q^m = T_i^m P^i$ . Transformations mapping points to hyperplanes, are denoted as e.g.  $\mathcal{L}_{ij}$ . Let  $\epsilon$  denote the  $3 \times 3 \times 3$  “cross-product tensor”, which is defined as  $\epsilon_{ijk} a^i b^j c^k = \det A$  where  $a, b$  and  $c$  are the three columns of matrix  $A$ . Among the 27 coefficients of  $\epsilon$ , 21 are zero (all coefficients with repeated indices), the others are equal to  $+1$  or  $-1$ .

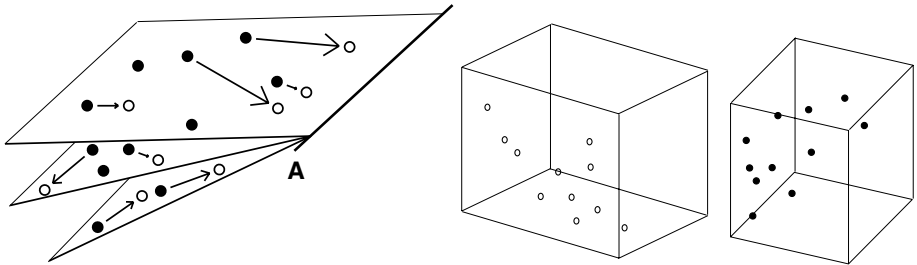
A *linear line complex* in 3D is a set of lines that are all incident with a line  $A$ , the *axis* of the linear line complex [8].

3D lines may be represented via 6-vectors of *Plücker coordinates*. Plücker coordinate vectors are defined up to scale and they must satisfy one bilinear constraint on their coefficients. The Plücker coordinates of a line  $A$  can be determined from any two different points on  $A$ , as follows. Let  $B$  and  $C$  be two points on  $A$  (represented by 4-vectors of homogeneous coordinates). Define  $([A]_\times)^{ij} = B^i C^j - C^i B^j$ . This is a skew-symmetric  $4 \times 4$  matrix and has thus only six different non zero coefficients – these are the Plücker coordinates of  $A$ . There are several possibilities of ordering the coefficients to get a 6-vector, we choose the following:

$$[A]_\times = \begin{pmatrix} 0 & -A_4 & A_6 & -A_2 \\ A_4 & 0 & -A_5 & -A_3 \\ -A_6 & A_5 & 0 & -A_1 \\ A_2 & A_3 & A_1 & 0 \end{pmatrix}.$$

### 3 Problem Statement

We consider a dynamic scene of the type described above. Any point of the scene may move freely<sup>2</sup> inside what we call its *motion plane*. All motion planes form a pencil of planes, whose axis is a 3D line  $A$  (see figure 1). For ease of expression, we also call  $A$  the *horizon* or *horizon line* of the motion (although  $A$  need not be a line at infinity in general). Let the positions of some point at three different time instants be represented by the 4-vectors of homogeneous coordinates,  $P, P'$  and  $P''$ . We call *point motion* the “displacement” of an individual point between different time instants.



**Fig. 1.** Left: the considered scenario – points moving in a pencil of motion planes. Right: 3D views at two time instants.

<sup>2</sup> This includes that a point may actually be static.

We consider that the scene is observed by a *moving* stereo system (consisting of two or more cameras). We suppose that at each time instant, a *3D view* of the current state of the scene can be obtained. In the most general case, this will be a projective reconstruction, based on a weak calibration of the stereo system, for the images taken at the considered instant. The stereo system is considered to be moving<sup>3</sup>, so different 3D views are represented in different coordinate frames (see figure 1). We call *stereo motion* the transformation between these coordinate frames. Let  $T'$  respectively  $T''$  be the transformations mapping points from the second respectively third 3D view, into the frame of the first one. Let  $Q, Q'$  and  $Q''$  be the coordinates inside the 3D views, of a moving point  $P$  at three time instants, i.e. of the points  $P, P'$  and  $P''$ . The basic question dealt with in this paper is, which amount of stereo and point motion (i.e. scene structure) can be reconstructed, given the input of matching 3D views.

We first study this question for the case of two 3D views, by deriving the associated matching tensor and showing what information on stereo and point motion can be extracted from it. We show that in general, i.e. for unconstrained motion of individual points inside their motion planes, a complete reconstruction is not possible, even if arbitrarily many views (for arbitrarily many time instants) are available. Several ways of obtaining a complete reconstruction, are then described. These are based on additional knowledge about point motion, e.g. knowledge that individual points are actually static or that points are moving on linear trajectories.

## 4 Two 3D Views – The Projective Case

### 4.1 The Matching Tensor – A Kind of 3D Epipolar Geometry

The structure of all points, observed at two time instants, may be represented as a linear line complex: the lines spanned by pairs of corresponding points  $P$  and  $P'$ , are all bound to lie in the pencil of motion planes, thus they all intersect the pencil's axis  $A$ .

Let us now consider two 3D views of the dynamic scene, taken at two different time instants, by an uncalibrated stereo system. Hence, the 3D views are projective reconstructions of the scene, at the respective time instants. Let point positions in the first 3D view be denoted as  $Q$  and in the second one, as  $Q'$ . If we knew the stereo motion  $T'$  and  $A$ , the point motion's horizon line in the first 3D view, then, after mapping all  $Q'$  by  $T'$ , the lines spanned by corresponding points  $Q$  and  $T'Q'$ , would form a linear line complex, with  $A$  as axis, as observed above. Let  $B$  and  $C$  be any two points on  $A$ . We must have coplanarity of  $Q, T'Q', B$  and  $C$ , thus:

$$\det \begin{pmatrix} | & | & | & | \\ B & C & Q & T'Q' \\ | & | & | & | \end{pmatrix} = 0 . \quad (1)$$

This equation is bilinear in the coefficients of the reconstructed 3D points  $Q$  and  $Q'$  and we may rewrite it in the following form:

$$Q^i Q'^j \mathcal{L}_{ij} = 0 , \quad (2)$$

<sup>3</sup> Note that it is nowhere required that the stereo system be moving rigidly or the individual cameras have constant intrinsic parameters or the like.

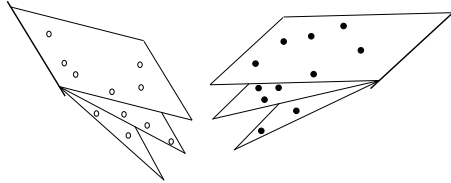
where  $\mathcal{L}$  is a  $4 \times 4$  matrix, that depends on the stereo motion and the point motion's horizon line. We might call  $\mathcal{L}$  a “Linear Line Complex Tensor”, or, L-tensor for short, for the reasons given above. The coefficients of the two points  $B$  and  $C$ , that appear in  $\mathcal{L}$ , can all be contracted to the Plücker coordinates of  $A$ . It is then easy to derive the following decomposition of  $\mathcal{L}$ :

$$\mathcal{L} \sim T'^T [A]_{\times} . \quad (3)$$

In the following, we describe several properties of the tensor and in §4.2 we explain, what information can be extracted from it.

The matrix  $[A]_{\times}$  is of rank two at the most, since its coefficients are Plücker coordinates (they satisfy the constraint  $A_1 A_4 + A_2 A_5 + A_3 A_6 = 0$ ). Hence,  $\mathcal{L}$  too is of rank two at the most. The right and left null spaces of  $\mathcal{L}$  represent nothing else than the horizon line  $A$ : the right null space consists of the 3D points that lie on  $A$ , in the first 3D view, whereas the left null space contains the 3D points lying on the reconstruction of the horizon line  $A'$  in the second 3D view.

In the following, we give some geometric interpretation (cf. figure 2) of the L-tensor, and actually show that there are some analogies to the epipolar geometry between two 2D views of a rigid scene. Let us first consider the action of  $\mathcal{L}$  on a point  $Q$  in the first 3D reconstruction. The product  $([A]_{\times})_{ij} Q^i$  gives the motion plane  $\Pi_j$ , that is spanned by the horizon line  $A$  and the point  $Q$  [8]. The transformation  $T'^T$  maps planes from the first 3D view, onto planes in the second one:  $\Pi'_k \sim (T'^T)^j_k \Pi_j$ . The plane  $\Pi'$  contains the horizon line  $A'$ . The correspondence of a point  $Q'$  with  $Q$  is then expressed as  $Q'$  lying on  $\Pi'$ , or:  $Q'^k \Pi'_k = 0$ .



**Fig. 2.** 3D epipolar geometry.

The analogy to the 2D epipolar geometry is straightforward. The horizon lines  $A$  and  $A'$  (they represent the same “physical” line, but in 3D views taken at different stereo positions) play the role of the epipoles. In each 3D view, there is a pencil of “epipolar motion planes” containing the horizon line, which is analogous to pencils of epipolar lines in 2D views. Concerning the transformation  $T'$ , there is an analogous expression to  $\mathcal{L} \sim T'^T [A]_{\times}$  for the 2D epipolar geometry: any plane homography, multiplied by the skew-symmetrix matrix of an epipole, gives the fundamental matrix. Plane homographies are those 2D homographies that map one epipole onto the other and that map corresponding epipolar lines onto each other. Hence, plane homographies are defined up to three parameters. Here,  $T'$  is a 3D homography. It is constrained to map

epipoles  $A$  and  $A'$  onto each other and to map corresponding motion planes onto each other. A difference compared to the 2D epipolar geometry is that here, the epipoles (the horizon lines) do represent a part of the dynamic scene structure, and not only the camera geometry. Also, for a given 3D view, the epipole with respect to any other 3D view of the same dynamic scene, is always the same, whereas in the 2D case, the epipoles of one view with respect to several other views, are in general different from each other.

## 4.2 What Can Be Extracted from the L-Tensor?

It would be desirable to extract, from the tensor  $\mathcal{L}$ , the stereo system's motion  $T'$  and the point motion's horizon line  $A$ . Unhappily, this is not entirely possible, which is clear when counting parameters:  $\mathcal{L}$  offers at most 11 constraints (it is a rank-2  $4 \times 4$  matrix, defined up to scale), which is not sufficient to cover the 15+4 parameters for  $T'$  and  $A$ .

From the decomposition of  $\mathcal{L}$  in (3), it is clear that the horizon line  $A$  can be extracted via the right nullspace of  $\mathcal{L}$  (the horizon line  $A'$  in the second 3D view is the left nullspace). The question left is, how much information can be gained on the stereo motion  $T'$ ? Let  $H$  be any non-singular 3D homography that maps  $A$  to the line at infinity consisting of all points  $(X, Y, 0, 0)^\top$ . It can be shown that multiplying equation (3) from the right with the inverse of  $H$  leads to:

$$\mathcal{L}H^{-1} \sim T'^\top \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \sim \begin{pmatrix} 0 & 0 & -T'_{41} & T'_{31} \\ 0 & 0 & -T'_{42} & T'_{32} \\ 0 & 0 & -T'_{43} & T'_{33} \\ 0 & 0 & -T'_{44} & T'_{34} \end{pmatrix}.$$

Hence,  $\mathcal{L}$  gives us 7 coefficients of  $T'$  (discarding the scale ambiguity).

Let  $M'$  be any  $4 \times 4$  matrix whose third and fourth rows are the same as that of  $T'$ , but with arbitrary coefficients in the first two rows. Any such  $M'$  maps the horizon line  $A'$  of the second 3D view onto  $A$  in the first 3D view (to be precise, it maps all points on  $A'$  onto points on  $A$ ) and the motion planes of the second 3D view (planes spanned by the  $Q'$  and the line  $A'$ ), onto the corresponding motion planes in the first view. What remains unknown however, is the motion *inside* the individual motion planes.

Mapping the second 3D view by any such  $M'$  will in the sequel be called *partial alignment of 3D views*. Methods for obtaining a *full alignment* are described further below. We now describe one method of performing partial alignment. Since everything is defined up to a global projective transformation, we perform the alignment such that the horizon line becomes the line at infinity, consisting of all points  $(X, Y, 0, 0)^\top$ , which leads to simpler expressions in the sequel. Let the Singular Value Decomposition (SVD) of  $\mathcal{L}$  be given as (remember that  $\mathcal{L}$  is of rank two):

$$\mathcal{L} = U \begin{pmatrix} a & & & \\ & b & & \\ & & 0 & \\ & & & 0 \end{pmatrix} V^\top.$$

Define the following projective transformations:

$$M = \begin{pmatrix} & & \sqrt{a} \\ & \sqrt{a} & \\ -\sqrt{a} & \sqrt{b} & \end{pmatrix} V^T \quad M' = \begin{pmatrix} & \sqrt{a} & \\ \sqrt{a} & \sqrt{a} & \\ & \sqrt{b} & \end{pmatrix} U^T .$$

These transformations are by construction non-singular (unless  $a = 0$  or  $b = 0$ ). Transforming the first 3D view by  $M$  and the second one by  $M'$  leads to points  $MQ$  and  $M'Q'$  that satisfy the following constraint:

$$(M'Q')^T \mathcal{L}' MQ = 0$$

where

$$\mathcal{L}' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

is the L-tensor of the partially aligned 3D views.

Before describing methods for full alignment, which will be done in §6, we consider the specialization of our scenario to the Euclidean and affine cases.

## 5 Two 3D Views – The Euclidean Case

We now consider the case where the 3D views are Euclidean reconstructions, obtained using e.g. a calibrated stereo system. In addition, we concentrate on the case of parallel motion planes, which is probably the most interesting one to study. This means that  $A$  is a line at infinity, thus  $A_4 = A_5 = A_6 = 0$  and

$$\mathcal{L} \sim T'^T [A]_{\times} \sim T'^T \begin{pmatrix} 0 & 0 & 0 & -A_2 \\ 0 & 0 & 0 & -A_3 \\ 0 & 0 & 0 & -A_1 \\ A_2 & A_3 & A_1 & 0 \end{pmatrix} .$$

The vector  $a = (A_1, A_2, A_3)^T$  contains the homogeneous coordinates of the line  $A$ , on the plane at infinity. Thus, it also represents the homogeneous coordinates of the normal direction of all motion planes.

For Euclidean 3D views, the stereo motion is a similarity transformation, i.e. a rigid motion possibly followed by a global scale change, which is needed since the two 3D views might have different scales. Thus:

$$T' = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix}$$



for a rotation matrix  $R$ , a translation vector  $t$  and a scalar  $s$ . The tensor is thus given by:

$$\mathcal{L} = \begin{pmatrix} 0_{3 \times 3} & -sR^T a \\ a^T & -t^T a \end{pmatrix}.$$

It has a particularly simple structure with only 7 non zero coefficients, and no non-linear constraint on them. However, if the global scale  $s$  of the second 3D view, is known in advance, e.g. due to constant stereo calibration in which case  $s = 1$ , then there is one non-linear constraint: the norm of the leading 3-vectors in the 4th column and the 4th row of  $\mathcal{L}$  are the same.

What information on stereo and point motion can be extracted from  $\mathcal{L}$ ? The horizon line can be read off directly, as the leading 3-vector of the 4th row. The scale  $s$  is obtained as the ratio of the norms of the two leading 3-vectors in the 4th column and 4th row. As for  $R$ , it can be seen that it can be determined, up to a rotation about  $a$ , the normal direction of the motion planes (see above). Finally, as for the translation  $t$ , only its amount along the direction  $a$ , can be determined.

Thus, the L-tensor allows, like in the projective case, only partial alignment of 3D views. Here, the ambiguity has three degrees of freedom: let  $T'$  be any similarity transformation doing the partial alignment. Adding any transformation consisting of a rotation about  $a$  and a translation perpendicular to  $a$ , will also result in a valid alignment transformation. Contrary to the projective case, the ambiguous transformation is the same for all motion planes, i.e. if the ambiguity can be cancelled in one motion plane only, then it can be so for the entire 3D scene alignment (in the projective case, full alignment of at least two planes is necessary).

## 6 Three 3D Views

As discussed previously, two 3D views are not sufficient for full alignment. We now examine if and how additional 3D views, obtained at other time instants, allow to reduce the ambiguity. Let us first note that even with three or more 3D views of our scenario, without additional information, full alignment is not possible. Every 3D view can be partially aligned with the others, as described above, but it is easy to see that the ambiguity in the alignment can not be reduced without further information. In the following, we outline a few types of additional information, that indeed may contribute to full alignment of 3D views.

First, suppose that every point has a linear trajectory. Wolf and Shashua have derived the matching constraints for three 3D views of this scenario [11]. The so-called join tensors, or J-tensors for short, allow to perform full alignment of the three 3D views. This holds even if the linear trajectories are in general position, i.e. if they are not bound to lie on a pencil of planes. The drawback of this general case is that a linear solution of the J-tensors requires at least 60 corresponding point triplets. In §6.1, we specialize the J-tensors to our scenario, and show how this allows full alignment using fewer correspondences.

Second, we remind that until now we did not assume that there are more than one point per motion plane. Thus, it might be interesting to study the case of one or several motion

planes containing several points. This can actually be detected after partial alignment, see §7.2. In this case, motion planes with enough moving points on them, can be dealt with individually, e.g. by estimating their homography tensor [9]. Since in our scenario, we already know at least one line correspondence per motion plane (the horizon line), we might consider a simplified version of the H-tensor (see §6.2). Each motion plane for which the H-tensor can be estimated, can thus be fully aligned, and it is easy to show that the alignment of two or more motion planes is sufficient to align the rest of the scene.

Third, knowledge of static points helps of course in the alignment of the 3D views. This will be described briefly in §6.3. Other possibly useful types of additional information could be knowledge of conical trajectories, of motion with constant velocity, of linear trajectories going in the same direction, etc.

## 6.1 Linear Trajectories

The join tensors, introduced for the general case of unconstrained linear trajectories [11], can also be used here of course. However, in our specialized scenario, we can exploit the additional constraint that the trajectories form a linear line complex (they lie in a pencil of motion planes). It is possible to derive tensors that fully encapsulate this constraint, but they are numerous and not very intuitive. Rather, we suppose in the following that partial alignment of the three 3D views has been performed (e.g. the second and third views have been aligned with the first one), as described in §4.2, and derive matching constraints on the already partially aligned 3D views.

We remind that the horizon line  $A$  in the aligned 3D views, is the line at infinity consisting of points  $(X, Y, 0, 0)^T$ . Hence, the motion planes are given by 4-vectors of homogeneous coordinates of the form  $(0, 0, s, -t)^T$ . We are looking for transformations  $T'$  and  $T''$  for the second and third 3D views, that leave the horizon line and all motion planes globally fixed. Hence, the transformations are of the following form:

$$T' = \begin{pmatrix} a' & b' & c' & d' \\ e' & f' & g' & h' \\ 0 & 0 & j' & 0 \\ 0 & 0 & 0 & j' \end{pmatrix} \quad T'' = \begin{pmatrix} a'' & b'' & c'' & d'' \\ e'' & f'' & g'' & h'' \\ 0 & 0 & j'' & 0 \\ 0 & 0 & 0 & j'' \end{pmatrix}. \quad (4)$$

Let  $Q$ ,  $Q'$  and  $Q''$  represent triplets of corresponding points. The matching constraint used here is that  $Q$ ,  $T'Q'$  and  $T''Q''$  have to be collinear, which means that the rank of the  $4 \times 3$  matrix composed of these 3 vectors, is two at the most. This constraint can be expressed by a linear family with four degrees of freedom, of  $4 \times 4 \times 4$  join tensors [11]. In our case, due to the special form of  $T'$  and  $T''$ , the four degrees of freedom remain, but some coefficients are known to be zero for all join tensors, i.e. fewer than the 60 correspondences for the general family of join tensors, are needed here to estimate them.

One problem here, that actually turns out as a benefit, is that the point correspondences available to us in the considered scenario, are constrained – all triplets of points  $Q$ ,  $Q'$ ,  $Q''$  lie in some “horizontal” plane (in the chosen projective frame). Hence, the

estimation of the join tensors is underconstrained<sup>4</sup>. Hence, if we were to estimate general  $4 \times 4 \times 4$  tensors, there would be a family of solutions of degree higher than four.

We now consider matching constraints for triplets of points lying in a motion plane  $(0, 0, s, -t)^\top$ , thus  $Q \sim (X, Y, t, s)^\top$ ,  $Q' \sim (X', Y', t, s)^\top$  and  $Q'' \sim (X'', Y'', t, s)^\top$ . These being collinear after alignment, is expressed by:

$$\text{rank} \left( \begin{array}{c|cc} X & a'X' + b'Y' + c't + d's & a''X'' + b''Y'' + c''t + d''s \\ Y & e'X' + f'Y' + g't + h's & e''X'' + f''Y'' + g''t + h''s \\ t & j't & j''t \\ s & j's & j''s \end{array} \right) \leq 2.$$

In the general case, four join tensors forming a basis for the 4-degree-of-freedom family, might be extracted by expressing that the four possible 3-minor's determinants vanish. With our input, it is clear that the two minors containing both the third and fourth rows of the above matrix, always vanish. Hence, the corresponding join tensors can not be estimated. As for the other two minors, we may write:

$$\begin{aligned} \epsilon_{lpq} (M_i^l Q^i) (M_m^p T_j'^m Q'^j) (M_n^q T_k''^n Q''^k) &= 0 \\ \epsilon_{lpq} (M_i^l Q^i) (M_m^p T_j'^m Q'^j) (M_n^q T_k''^n Q''^k) &= 0 \end{aligned}$$

where the  $3 \times 4$  matrices  $M$  respectively  $M'$  project points  $(X, Y, Z, W)^\top$  to  $(X, Y, Z)^\top$  respectively  $(X, Y, W)^\top$ , i.e.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We thus obtain the following two join tensors:

$$\mathcal{J}_{ijk} = \epsilon_{lpq} M_i^l (M_m^p T_j'^m) (M_n^q T_k''^n) \quad \mathcal{J}'_{ijk} = \epsilon_{lpq} M_i^l (M_m^p T_j'^m) (M_n^q T_k''^n).$$

The slices  $\mathcal{J}_{4jk}$  and  $\mathcal{J}'_{3jk}$  are zero matrices, and  $\mathcal{J}_{3jk}$  and  $\mathcal{J}'_{4jk}$  are identical. As for the other two slices, the coefficients with indices lower than 3 inside the slice, are identical in the two tensors. Among the other coefficients, there are several that are the same in both tensors, but that stand at different places. Each one of  $\mathcal{J}$  and  $\mathcal{J}'$  has only 30 non-zero coefficients. However, again due to the specific type of input correspondences, the tensors can only be estimated up to a 3-degree-of-freedom family of solutions each. Happily, the nature of the ambiguity in the solutions, is known and simple: 24 of the non-zero coefficients for each tensor, can be estimated without ambiguity (up to scale). As for the remaining coefficients, what can be estimated are the following sums:  $\mathcal{J}_{134} + \mathcal{J}_{143}$ ,  $\mathcal{J}_{234} + \mathcal{J}_{243}$ ,  $\mathcal{J}_{334} + \mathcal{J}_{343}$  and  $\mathcal{J}'_{134} + \mathcal{J}'_{143}$ ,  $\mathcal{J}'_{234} + \mathcal{J}'_{243}$ ,  $\mathcal{J}'_{434} + \mathcal{J}'_{443}$ .

So, 26 point correspondences are in general sufficient to obtain a linear solution for the 24 coefficients and the 3 sums of coefficients (per tensor). The following coefficients of the alignment transformations can be read off directly from the estimated tensor

<sup>4</sup> It is important to note that although  $T'$  and  $T''$  conserve motion planes, their join tensors also express the fact that  $Q$ ,  $T'Q'$  and  $T''Q''$  may be collinear, for points  $Q$ ,  $Q'$  and  $Q''$  not lying in the same motion plane.

coefficients (after an arbitrary choice for  $j'$  and  $j''$ ):  $a', b', e', f', a'', b'', e'', f''$ . Having determined them, one can establish, using coefficients of  $\mathcal{J}$  and  $\mathcal{J}'$ , as well as the estimated values of  $a'$  etc., a simple linear equation system, to solve for the remaining 8 unknowns,  $c', d', g', h', c'', d'', g''$  and  $h''$ .

In summary, 26 correspondences are sufficient to determine the alignment transformations  $T'$  and  $T''$ , and it is nowhere required that there be more than a single moving point per motion plane.

## 6.2 Using Homography Tensors

We consider the same scenario as in the previous section, i.e. linear trajectories, but now suppose that there are motion planes carrying several points (which can be detected, see §7.2). In this case, we may deal with each motion plane separately.

Consider one motion plane, represented by  $(0, 0, s, -t)^\top$ . Let  $Q, Q'$  and  $Q''$  represent triplets of corresponding points on that plane. Hence, they have the form given in the previous section. The matching constraint for such triplets, corresponds to the homography tensor, or H-tensor for short, introduced in [9]. In that work, the matching constraint was derived for three 2D views of a dynamic planar scene, obtained by a moving 2D camera. Each such 2D view constitutes a projective reconstruction of the planar scene, at the corresponding time instant. Here, we start with three 3D views, which essentially gives us again three projective reconstructions of the motion plane considered.

In order to compute the homography tensor for a motion plane, we first project the three 3D views of the plane by some projection matrix onto some 2D image plane. Let us define the  $4 \times 4$  projection matrix  $M$ :

$$M = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & t & s \end{pmatrix}$$

whose optical center is guaranteed not to lie on the considered motion plane. We project all three 3D views using  $M$ . For the resulting 2D views, there must exist  $3 \times 3$  transformations  $H'$  and  $H''$  such that all triplets  $MQ, H'MQ'$  and  $H''MQ''$  are collinear. In addition, we know a correspondence of a static line, the motion's horizon line. This line is mapped, by  $M$ , to the line at infinity of the 2D views. Hence,  $H'$  and  $H''$  must be affine transformations of the form:

$$H' = \begin{pmatrix} a' & b' & c' \\ e' & f' & g' \\ 0 & 0 & j' \end{pmatrix} \quad H'' = \begin{pmatrix} a'' & b'' & c'' \\ e'' & f'' & g'' \\ 0 & 0 & j'' \end{pmatrix} .$$

We obtain the following matching equation, in tensorial notation:

$$\epsilon_{lmn} (M_i^l Q^i) (H_p'^m M_j^p Q'^j) (H_q''^n M_k^q Q''^k) = 0$$

We may rewrite the equation:

$$(M_i^l Q^i) (M_j^p Q'^j) (M_k^q Q''^k) \underbrace{(\epsilon_{lmn} H_p'^m H_q''^n)}_{\mathcal{H}_{lpq}} = 0 .$$

We may identify  $\mathcal{H}_{lpq}$  as the  $3 \times 3 \times 3$  homography tensor. It can be shown that, due to the constrained form of  $H'$  and  $H''$ , the tensor has only 19 non-zero coefficients (compared to 27 for the general H-tensor). Hence, a linear solution is possible with 18 or more correspondences. Extracting the individual transformations  $H'$  and  $H''$  from  $\mathcal{H}$  can be done analogously to what is described in [9].

The tensor  $\mathcal{H}$ , for one motion plane, allows to partially determine  $\mathcal{J}$  and  $\mathcal{J}'$  (valid for all motion planes), dealt with in §6.1. Several coefficients of  $\mathcal{H}$  occur identically in  $\mathcal{J}$  or  $\mathcal{J}'$ , and the others give linear equations on coefficients of the join tensors.

It is easy to show that the alignment of two motion planes is sufficient to fully align the entire 3D views: for any 3D point in, say, the second 3D view, which we will call  $Q'$ , let  $D'$  be a line passing through it, but that is not contained in  $Q'$ 's motion plane. Let  $B'$  and  $C'$  be the intersection points of  $D'$  with the two motion planes for which full alignment is possible. We thus may compute the positions  $B$  and  $C$  of the points  $B'$  and  $C'$ , after alignment. The aligned position  $Q$  of  $Q'$  is finally given by the intersection of  $Q'$ 's motion plane, with the line joining  $B$  and  $C$ .

### 6.3 Using Static Points

Given the special form of the alignment matrices (see (4)), it is clear that one static point (that is known to be static), provides two independent equations on each of them. Hence, correspondences associated to four static points in general position, should be sufficient to achieve full alignment of the 3D views (compared to five correspondences that would be required without the specific nature of our scenario). In the Euclidean case, two point correspondences (actually, one and a half) are sufficient for full alignment (compared to three that would be required without the specific nature of our scenario). It would be interesting to study the general case of mixed static and moving points.

## 7 Other Issues

### 7.1 Projections $P^5 \rightarrow P^3$

The derivation of the matching tensor for the two-view scenario (see §4.1), could also be performed in the framework of higher dimensional projection matrices used in [12]. Without loss of generality (we deal with projective space), suppose that the horizon  $A$  is the line at infinity containing all points  $(X, Y, 0, 0)^T$ . Let  $P \sim (X, Y, Z, W)^T$  be a 3D point at the first time instant and let  $P' \sim (X + a, Y + b, Z, W)^T$  be the same point, at the second instant, after having moved in the plane spanned by  $A$  and  $P$ . We may form the following 6-vector that represents  $P$  and its moved version  $P'$ :  $S^T = (X \ Y \ a \ b \ Z \ W)$ . We define two projection matrices  $P^5 \rightarrow P^3$ :

$$M \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad M' \sim \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

We may observe that  $MS \sim P$  and  $M'S \sim P'$ . In our scenario, we do not observe  $P$  and  $P'$  directly, but have projective 3D views of them, i.e.:  $\lambda Q = TP$  and  $\lambda'Q' = T'P'$

for some  $4 \times 4$  projective transformations  $T$  and  $T'$  and scale factors  $\lambda$  and  $\lambda'$ . We may derive the matching constraints for  $Q$  and  $Q'$  in the way shown e.g. in [10]: due to

$$\underbrace{\left( \begin{array}{c|c|c} TM & Q & 0 \\ T'M' & 0 & Q' \end{array} \right)}_{X_{8 \times 8}} \begin{pmatrix} S \\ -\lambda \\ -\lambda' \end{pmatrix} = 0$$

we know that the matrix  $X$  is rank-deficient, i.e. that its determinant is equal to zero. By developing the determinant, one obtains the same  $4 \times 4$  tensor  $\mathcal{L}$  as in §4.1 (if we set  $T$  to the identity).

## 7.2 Segmentation of Points Moving in the Same Motion Plane

After partial alignment (see §4.2), the segmentation of points that move in the same plane, is straightforward and can in principle be done in a single 3D view. This might be done by checking, in 3D, if points are on the same motion plane. An alternative would be to compute plane homographies between the 2D views inside a stereo system, for individual motion planes, and check if corresponding projections of 3D points in the 2D views, are consistent with the plane homographies.

## 7.3 Self-Calibration

We briefly describe a self-calibration algorithm for the scenario of two projective 3D views, under the assumption that the true motion planes are parallel to each other, i.e. the true horizon line is a line at infinity. Using the L-tensor, the horizon line can be determined in the 3D views. Since the true line is a line at infinity, it has two intersection points with the absolute conic – the circular points of all motion planes. We may perform partial self-calibration by searching for the circular points, on the reconstructed horizon lines in our 3D views.

Consider one of the 3D views, after partial alignment as described in §4.2. We suppose that this 3D view has been obtained using two perspective cameras, with unknown and possibly different focal lengths, but known other intrinsic parameters. The two focal lengths can in general be recovered from the epipolar geometry [5], but this is nearly always singular in practice, due to optical axes passing close to each other [7]. The knowledge of a line at infinity in the projective reconstruction, however, can be used to overcome the singularity, as described in the following.

Let  $M$  and  $M'$  be the  $3 \times 4$  projection matrices of the two 2D views. We suppose that the known parts of the calibration matrices (containing aspect ratio and principal point) have been undone, i.e. the unknown calibration matrices of  $M$  and  $M'$  are  $K = \text{diag}(f, f, 1)$  and  $K' = \text{diag}(f', f', 1)$ . We parameterize the problem in the circular points on the horizon line, which, in the partially aligned 3D view, have coordinates  $C_{\pm} \sim (a \pm I, b, 0, 0)^T$  for real  $a, b$  and  $b \neq 0$ . Our self-calibration constraints are that the projections of  $C_+$  and  $C_-$  lie on the images of the absolute conic in the respective views, which leads to:

$$\begin{aligned} (am_1 + bm_2 + Im_1)^T K^{-T} K^{-1} (am_1 + bm_2 + Im_1) &= 0 \\ (am_1 + bm_2 - Im_1)^T K^{-T} K^{-1} (am_1 + bm_2 - Im_1) &= 0 \end{aligned}$$

where  $m_i$  is the  $i$ th column of  $M$ , and similar equations for the second view. Separating the real and imaginary parts of the equations leads to two equations, whose resultant with respect to  $f^2$  is quadratic in  $a$  and  $b$ . We get a similar equation for the second view. The resultant of these two equations, with respect to  $a$ , finally, is the product of the term  $b^2$  and a term that is linear in  $b^2$ . Since  $b \neq 0$ , we thus get a single solution for  $b^2$ , which gives us  $b$  up to sign (the sign does not matter). From  $b$ , unique solutions for  $a$  and the squared focal lengths may then be obtained.

We performed simulated experiments with this method. Twenty moving points on each of three planes were simulated. The 3D points were projected in two stereo pairs, and centered Gaussian noise with a standard deviation of 1 pixel was added to the image coordinates. For each stereo pair, the fundamental matrix was computed using the 8-point method [6], projective reconstruction was performed, and the L-tensor between the two resulting point sets estimated. The point sets were then partially aligned. For several stereo configurations (varying vergence angle), 100 simulation runs each were performed. Self-calibration gave focal lengths with an average relative error of about 6% (excepting between 0 and 4 runs where computation failed).

## 8 Experimental Results

We conducted the following experiment using four stereo pairs of a dynamic scene (see figure 3). About 60 points on the moving objects were manually extracted in all eight images. The experiment was performed for the Euclidean case: the calibration grid visible in the images, was used to obtain full stereo calibration, and thus Euclidean 3D reconstruction of the points. Each such 3D view underwent an arbitrary Euclidean transformation, otherwise they would already have been aligned, since stereo calibration was with respect to the static calibration grid.

From this input, the Euclidean L-tensor between the first and second 3D views was estimated and these 3D views were partially aligned (see §4.2). Then, the other two 3D views were aligned with the two first one, using a simpler variant of the method of §4.2 (the horizon line in the first two views is already known), not described here.

Full alignment was done for the first three 3D views, based on the knowledge that individual points moved on linear trajectories (see §6.1), and by estimating joint tensors specialized to Euclidean alignment transformations. Finally, the fourth 3D view was fully aligned with the others, again using a simplification of the method of §6.1. Some recovered point trajectories are shown in figure 4. Qualitatively, the result seems to be correct, although a quantitative evaluation should definitely be carried out.

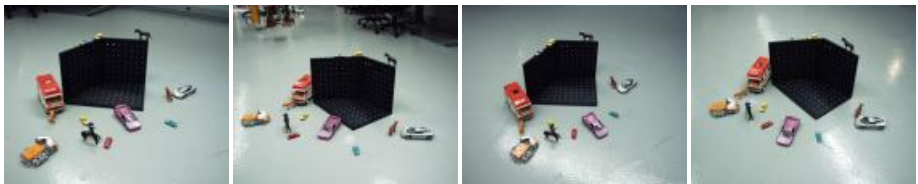
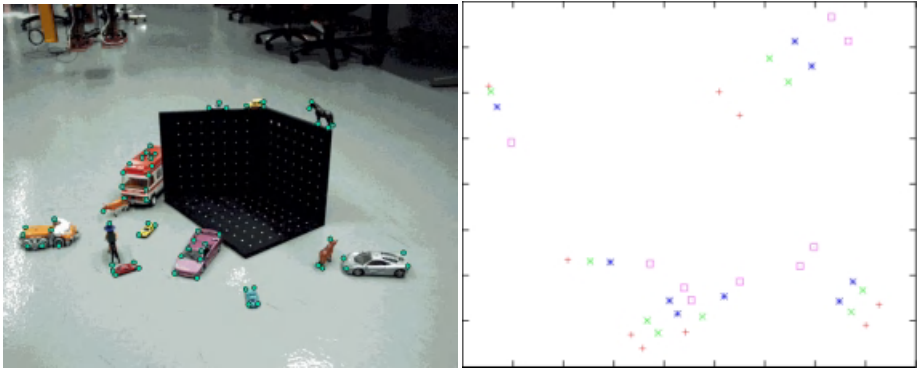


Fig. 3. Two stereo pairs used in the experiments.



**Fig. 4.** Left: the moving points used in the experiment. Right: recovered linear trajectories of several points (4 positions each), orthogonally projected onto the ground plane. The point group on the left corresponds to a point on the horse, the group at the bottom to the caravan (3 moving points shown), the group in between, to one of the cars on the grid. The other two point groups belong to the truck and to the sportscar in front of the grid (2 moving points each).

## 9 Conclusion

We have considered the structure and motion problem for a dynamic scene, consisting of individually moving points, with the restriction that motion happens in a pencil of motion planes. The scene is supposed to be observed by a moving stereo system, resulting in 3D views of the scene, at different time instants. We have derived the matching constraints between two such 3D views, and shown that full alignment of the views is not possible without further information. Information useful to fully recover the motion of the stereo system as well as the motion and structure of the scene, are for example knowledge of static points or linear trajectories. We have especially discussed how to take into account linear trajectories, to achieve full recovery of structure and motion. A preliminary experiment has shown that it may be feasible to solve the problem in practice, at least in the calibrated case.

Among issues for further work on this topic, minimum numbers of correspondences for the mixed case of moving and known/unknown static points, should be established, and a more thorough experimentation is needed.

**Acknowledgement.** I wish to thank Adrien Bartoli for preparing the experimental data of section 8.

## References

1. Costeira, J., Kanade, T.: A Multi-Body Factorization Method for Motion Analysis. ICCV – International Conference on Computer Vision (1995) 1071–1076



2. Fitzgibbon, A.W., Zisserman, A.: Multibody Structure and Motion: 3-D Reconstruction of Independently Moving Objects. ECCV – European Conference on Computer Vision (2000) 891–906
3. Han, M., Kanade, T.: Reconstruction of a Scene with Multiple Linearly Moving Objects. CVPR – International Conference on Computer Vision and Pattern Recognition, Vol. II (2000) 542–549
4. Han, M., Kanade, T.: Multiple Motion Scene Reconstruction from Uncalibrated Views. ICCV – International Conference on Computer Vision, Vol. I (2001) 163–170
5. Hartley, R.I.: Estimation of Relative Camera Positions for Uncalibrated Cameras. ECCV – European Conference on Computer Vision (1992) 579–587
6. Hartley, R.: In Defence of the 8-Point Algorithm. ICCV – International Conference on Computer Vision (1995) 1064–1070
7. Newsam, G.N., Huynh, D.Q., Brooks, M.J., Pan, H.P.: Recovering Unknown Focal Lengths in Self-Calibration: An Essentially Linear Algorithm and Degenerate Configurations. XVIIIth ISPRS Congress, Part B3 (1996) 575–580
8. Semple, J.G., Kneebone, G.T.: Algebraic Projective Geometry. Oxford Science Publications (1952)
9. Shashua, A., Wolf, L.: Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Planar Points. ECCV – European Conference on Computer Vision (2000) 507–521
10. Triggs, B.: Matching Constraints and the Joint Image. ICCV – International Conference on Computer Vision (1995) 338–343
11. Wolf, L., Shashua, A., Wexler, Y.: Join Tensors: On 3D-to-3D Alignment of Dynamic Sets. ICPR – International Conference on Pattern Recognition (2000) 388–391
12. Wolf, L., Shashua, A.: On Projection Matrices  $P^k \rightarrow P^2$ ,  $k = 3, \dots, 6$ , and their Applications in Computer Vision. ICCV – International Conference on Computer Vision, Vol. I (2001) 412–419

# What Does the Scene Look Like from a Scene Point?

M. Irani<sup>1</sup>, T. Hassner<sup>1</sup>, and P. Anandan<sup>2</sup>

<sup>1</sup> Dept. of Computer Science and Applied Mathematics  
The Weizmann Institute of Science

76100 Rehovot, Israel

<sup>2</sup> Microsoft Research

One Microsoft Way

Redmond, WA 98052-6399

**Abstract.** In this paper we examine the problem of synthesizing virtual views from scene points *within* the scene, i.e., from scene points which are imaged by the real cameras. On one hand this provides a simple way of defining the position of the virtual camera in an uncalibrated setting. On the other hand, it implies *extreme* changes in viewpoint between the virtual and real cameras. Such extreme changes in viewpoint are not typical of most New-View-Synthesis (NVS) problems.

In our algorithm the virtual view is obtained by aligning and comparing all the projections of each line-of-sight emerging from the “virtual camera” center in the input views. In contrast to most previous NVS algorithms, our approach does not require prior correspondence estimation nor any explicit 3D reconstruction. It can handle any number of input images while simultaneously using the information from all of them. However, very few images are usually enough to provide reasonable synthesis quality. We show results on real images as well as synthetic images with ground-truth.

**Keywords:** Novel-view synthesis, Synthesis *without* structure or motion.

## 1 Introduction

Consider a sporting event with several cameras taking pictures of the game. What does the scene look like from the point view of one of the players in the field? What does the ball “see”? In this paper we show how from a few images of the same scene we can synthesize a virtual view from a real physical scene point which is imaged by all the input cameras.

This problem can be seen as a special case of *New View Synthesis* (NVS). However, it is unique in two ways: (i) In the problem defined above, the virtual camera is located *within* the scene itself, whereas the real cameras view the scene from “outside”. This implies an *extreme* change in viewpoint between the virtual camera and each of the real cameras. Such extreme changes in viewpoint are not typical of most NVS examples. (ii) On the other hand, the specification of the new view location is simplified here since this location is visible in the input images. This allows to naturally specify a *physically meaningful position*

of the “virtual camera” with respect to the scene in an uncalibrated setting (i.e., without requiring Euclidean calibration of the scene).

In this paper, we describe an algorithm for synthesizing virtual views from scene points within the scene. Our algorithm requires *no prior correspondence estimation nor any 3D reconstruction*. The “epipole” of the virtual camera (the *virtual epipole*) in each of the real views is defined by the image of the physical scene point selected to be the location of the virtual camera. This point is visible in the input images. The color of each pixel in the virtual camera image is determined by aligning and comparing the projections of each line-of-sight (LOS) emerging from the virtual camera center in all the real images. This process does not require knowledge of the 3D structure of the scene and can be done without Euclidean calibration. This leads to a representation which bears some resemblance to the “generalized disparity space” of [16]. Our approach can handle any number of input images while simultaneously using the information available in all of those images.

Many algorithms for NVS have been proposed over the years. To better explain the benefits of our approach and to place it in the appropriate context, we briefly review existing approaches to NVS. These can broadly be classified into three classes of techniques:

The first class of techniques relies on 3D reconstruction of the scene followed by the rendering of the new view (e.g., [9,18,14,7,12]). However, the 3D reconstruction process is error-prone. These errors can lead to significant distortions in the reconstructed image because geometric error criteria often used to optimize 3D reconstruction do not translate gracefully into errors in rendered appearance. Furthermore, the 3D reconstruction is optimized in one coordinate system whereas the virtual view is rendered in another coordinate system. Distortions in synthesis due to inaccuracies in 3D reconstruction are amplified in cases of severe changes in viewing position between the real and virtual cameras. Many of these methods thus require a large number of input images taken from a wide range of viewing directions (e.g., Kanade et. al [11,12] report using 51 cameras placed on an elaborate rig surrounding the scene being reconstructed).

The second class of approaches avoids explicit 3D reconstruction, and instead utilizes dense correspondence estimation between the input views. These dense correspondences are then used to perform a “view transfer” from the input views to the virtual views (e.g., [1,20]). For example, given two images and their dense correspondence field, a third view can be synthesized by using geometric constraints induced by the Trifocal tensor on the location of these points in the three views. While these methods avoid explicit 3D reconstruction, the errors in correspondence result in similar distortions in the new view. Moreover, the synthesis part involves a forward warping step, which leads to “hole-filling” problems at surface discontinuities. These effects are amplified in cases of severe changes in viewing position between the real and virtual cameras.

The approach proposed in this paper performs *direct* synthesis without an intermediate step of correspondence estimation or explicit 3D reconstruction. Because the analysis and synthesis are done directly in the coordinate system of the virtual view, our process involves only backward (“inverse”) warping.

Backward-warping does not generate holes in the synthesized view, and handles more gracefully large changes in viewpoint and in image resolution between the synthesized and real images. Furthermore, our synthesis method optimizes errors directly in the coordinate system of the virtual view, thus avoiding many of the optimization problems associated with the first two classes of NVS methods. Our method can thus synthesize views of the scene from significantly different viewing positions than those of the real cameras (as required in order to synthesize the scene from a point *within* the scene).

A third class of NVS methods exists which, like our method, also avoids the need for 3D reconstruction and correspondence estimation altogether. This family of methods is exemplified by the “lightfield” [8] and the “lumigraph” [4]. However, these methods use very dense sampling of the view-space. They require an extreme number of input camera views to generate 4D representations of the scene by storing the radiance observed at *each point* in the 3D world in *each direction*. Synthesis then proceeds by extracting 2D slices of this dense 4D data volume, corresponding to the light observed in the requested viewing direction from the requested viewpoint. Acquiring all the images for generating this large 4D volume at preprocessing is a practical limitation of these methods. This problem amplifies when dealing with large-scale complex scenes or with dynamic scenes (i.e., scenes which change frequently), as collecting the required amount of data at reasonable space-time costs becomes practically impossible.

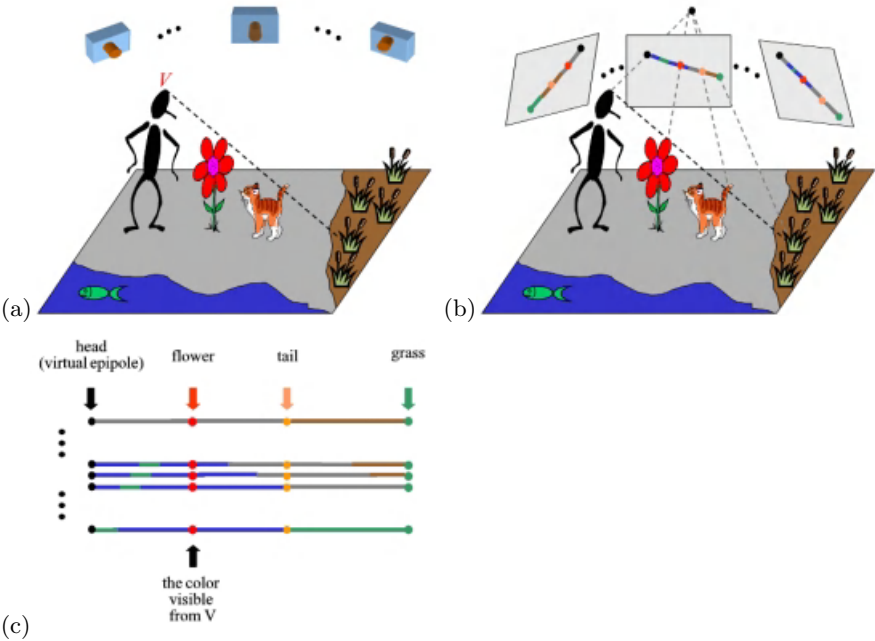
Unlike this family of methods, our approach does not need a large number of input images. In fact, very few images (in some experiments less than ten) are enough to provide reasonable synthesis quality by our method, with image quality degrading gracefully with fewer images.

Our algorithm is embedded in the Plane+Parallax geometric framework [6, 5, 2, 17, 15, 13]. By aligning all the input images with respect to a real planar surface in the scene (e.g., the ground plane), the camera matrices simplify to the camera epipoles between the real views. We further show that for some practical scenarios the explicit estimation of the epipoles (which may be difficult between widely separated views) is not necessary. Thus, in those situations we can also deal with cases where the real input cameras are situated very far apart from each other viewing the scene from significantly different viewing directions. This is on top of the wide base-line between the virtual camera and each of the real input cameras, which is inherently dictated by our NVS problem.

The rest of the paper is organized as follows: In Section 2 we provide an overview of our approach for solving this problem. The formulation of the problem using the Plane+Parallax geometric framework is described in Section 3. Section 4 summarizes the algorithm. In Section 5 we show how the problem further simplifies in several practical scenarios. Results are shown in Section 6.

## 2 Overview of the Approach

Figure 1 illustrates a scenario where  $n$  uncalibrated cameras image a 3D scene. We wish to synthesize a view of the scene from a scene point  $V$  (in this case -



**Fig. 1. Overview of the approach.** (a)  $n$  cameras imaging a scene. The position of the virtual camera  $V$  is selected as the tip of the man’s head. The *line of sight* (LOS) stretching from  $V$  to the patch of grass is shown as a thick dotted line. (b) The 3D LOS is projected via the  $n$  camera centers to 2D lines in the  $n$  input images. (c) The  $n$  projected lines associated with the 3D LOS are geometrically aligned and stacked. The leftmost (black) column corresponds to the virtual camera  $V$  in each of the images (i.e., the virtual epipoles). The color-consistent column closest to the projections of  $V$  is selected as the color visible by  $V$  when looking in the direction of the LOS (i.e. the man sees “red” when he looks in that direction).

the tip of the man’s head). To estimate what “color” (radiance) is visible from  $V$  when looking in any particular direction, one need only analyze the *line of sight* (LOS) stretching from  $V$  in that direction. Naturally, the first physical point (object) in the scene intersecting this LOS will be the point visible from  $V$ . For example, in our illustration, the LOS (shown as a thick dotted line in Figures 1.a and 1.b) stretching from  $V$  to the patch of grass on the ground, intersects both the flower and the cat’s tail. Since the flower is the closest physical object to  $V$  on this LOS, its color (red) is the one visible from  $V$  in this viewing direction.

The 3D LOS is not available to us. Instead, assuming that the 3D LOS is not occluded from the real cameras, what we do have is the 2D projections of the 3D LOS in each of the input images (Figure 1.b). The physical points on the 3D LOS (e.g. head, flower, tail, grass) are thus projected onto the corresponding points on the 2D lines. However, because the 3D LOS is not fully occupied with physical objects these 2D projected lines also contain projections of points

“behind” the 3D LOS which are visible through the “gaps” (such as the blue from the water, the green from the fish, etc.) Given one such 2D projection of the LOS obtained from one input image we cannot tell which of these colors originated from points on the LOS and which colors originated from points not on the LOS. However points on the LOS (i.e., the head, flower, tail and grass) will consistently appear in all projections of the LOS in *all* input images. This is not the case for points not on the LOS. Therefore, if we can identify where all the 2D projections of a single 3D LOS are, geometrically align them and stack them (see Figure 1.c), then physical points on the LOS will be distinguished from other points (seen through the “gaps”) as they will generate uniformly colored columns within the stacked lines (e.g. the black, red, ginger and green columns in Figure 1.c). Other columns (points) will not have consistent colors and can therefore be ruled out (e.g., the brown, green and blue colors of Figure 1.c). The leftmost (black) column corresponds to the virtual epipoles of the virtual camera  $V$  on each of the input images. Of all the color-consistent columns the one closest to the projections of the virtual camera  $V$  is selected as the color visible from  $V$  when looking in the direction of the LOS. In our example it is the red color (the flower’s leaf). Applying this logic to any viewing direction allows us to estimate the complete view of the scene from the virtual camera  $V$ .

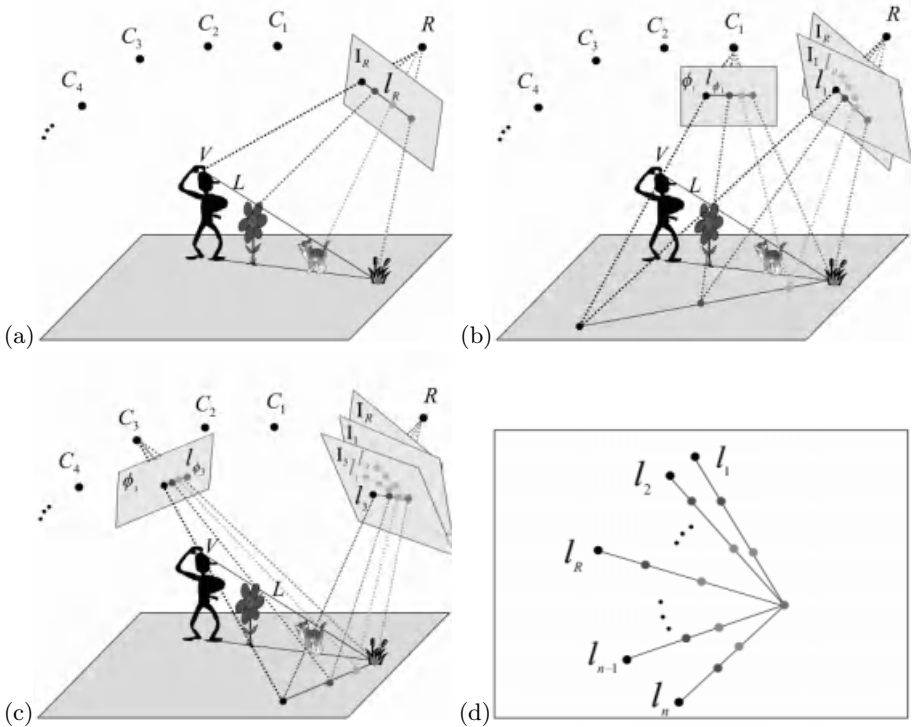
### 3 Formulating the Problem Using “Plane+Parallax”

We next show how the detection and alignment of the lines shown in Figure 1 becomes simple using the “Plane+Parallax” geometric framework (e.g., [6,5,2,17,15,13]). The 2D line projections of the 3D Line-of-Sight (LOS)  $L$  in Figure 1.b are in different coordinate systems of the different camera views. One point correspondence across these lines is known (the images of the virtual camera  $V$ ). We next show that by aligning any planar surface in the scene across all these images, these lines transform to a single coordinate system forming a pencil-of-lines which emerge from a common axis point. In fact, this axis point is the piercing point of the LOS  $L$  with the planar surface. The axis point and the known virtual epipoles uniquely define these lines for any LOS  $L$ .

#### 3.1 Pencil of Lines

Let  $\Phi_0, \Phi_1, \dots, \Phi_n$  be images of a scene taken using cameras with unknown calibration parameters. Let  $\Phi_0$  denote the “reference image”. Let  $\Pi$  be a plane in the scene that is visible in all the images (e.g., could be the ground plane). We can align all the images with respect to the plane  $\Pi$  by estimating the homography  $\mathcal{H}_i$  of  $\Pi$  between the reference image  $\Phi_0$  and each of the other images. Warping the images by those homographies yields a new set of images  $\{\mathcal{I}_i\}_{i=0}^n$ , where for any pixel  $p$  in image  $\mathcal{I}_i$ :  $\mathcal{I}_i(\mathbf{p}) = \Phi_i(\mathcal{H}_i \mathbf{p})$ . Note that the reference image  $\mathcal{I}_0 = \Phi_0$  remains unchanged. We will refer to it as  $\mathcal{I}_R$ , to emphasize that this is the reference image.

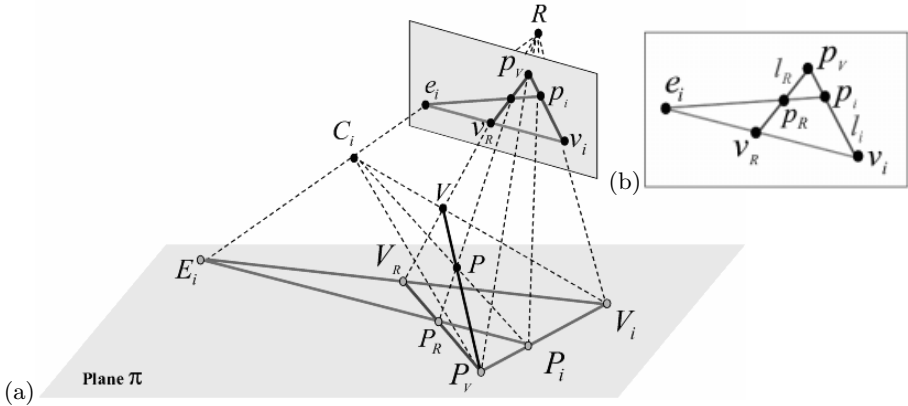
We will next show that all the 2D line projections of a 3D line-of-sight (LOS) form a *pencil of lines* in the plane-aligned images  $\mathcal{I}_R, \mathcal{I}_1, \dots, \mathcal{I}_n$ . In the generation



**Fig. 2. The pencil of lines.** (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))

of image  $\mathcal{I}_i$  from  $\Phi_i$  each pixel in  $\Phi_i$  was warped by the homography  $\mathcal{H}_i$ . The geometric meaning of this operation is displacing each pixel in  $\Phi_i$  as if its corresponding 3D point was on the plane  $\Pi$ , as shown in Figure 2.b. Points that are truly on  $\Pi$  (e.g., the green grass) will thus appear in  $\mathcal{I}_i$  in their correct image position (i.e., will be aligned with their corresponding point in  $\mathcal{I}_R$ ), whereas points not on  $\Pi$  (e.g., the tail, flower, or man's head) will be misaligned with their corresponding points in  $\mathcal{I}_R$ . (The farther a point is from  $\Pi$ , the more misaligned it will be.)

Let  $l_{\Phi_i}$  denote the 2D projection of the 3D LOS  $L$  on image  $\Phi_i$  (Figure 2). As a result of plane alignment,  $l_{\Phi_i}$  is transformed by the homography  $\mathcal{H}_i$  to a line  $l_i$  in image  $\mathcal{I}_i$  (see Figures 2.b and 2.c). We can see from Figure 2.d that all these lines  $\{l_i\}_{i=0}^n$  (one from each image  $\{\mathcal{I}_i\}_{i=0}^n$ ) form a *pencil of lines*. The axis point of this pencil (the green point) corresponds to the image of the “piercing point” of the LOS  $L$  with  $\Pi$  (in this case – the grass). The black points at the other end of the lines  $\{l_i\}_{i=0}^n$  correspond to the “virtual epipoles” of the virtual camera  $V$  in  $\mathcal{I}_R, \mathcal{I}_1, \dots, \mathcal{I}_n$ . These lines ( $\{l_i\}_{i=0}^n$ ) are the lines that need to be “aligned” to obtain the stacked representation of Figure 1.c.



**Fig. 3. The LOS line configuration.** (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))

The virtual epipoles in each image (the black points) are independent of the lines-of-sight  $L$ . Their image coordinates are known, since these points are defined by the user (see Section 4). As to the axis point of the pencil of lines: Every pixel in image  $\mathcal{I}_R$  corresponds to a piercing point of a different LOS  $L$  with  $\Pi$ . Connecting that pixel to the known “virtual epipoles” uniquely defines the pencil of lines corresponding to the LOS  $L$ .

### 3.2 Aligning and Stacking the Lines

We next show how to bring the lines  $\{l_i\}_{i=1}^n$  into alignment with  $l_R$ . This step is necessary so that they can be stacked and processed to find the first color-consistent column (that which is closest to the virtual epipoles), in order to determine which color is visible from  $V$  in the direction of the LOS  $L$  (see Figure 1.c).

We first define some notations. Let  $R, C_1, \dots, C_n$  denote the camera centers of the real cameras.  $V$  denotes the 3D scene-point from which we wish to synthesize the virtual view, i.e., the camera center of the “virtual camera” (whose 2D projection in each of the images is known). Let  $L$  be any line-of-sight, and let  $P$  be a physical scene point on  $L$ . Every pair of the above-mentioned camera-centers or scene-points defines a 3D line. We denote by upper-case letters the piercing points of these lines with the plane  $\Pi$  (see Figure 3.a), and by corresponding lower-case letters the projections of these piercing points in the coordinate system of the reference camera  $R$ . Thus, for example, the piercing-point of the line passing through  $C_i$  and  $P$  is denoted by  $P_i$ , the piercing-point of the line passing through  $R$  and  $P$  is denoted by  $P_R$ , and the piercing-point of the line passing through  $V$  and  $P$  (which is the LOS  $L$ ) is denoted by  $P_V$ .

Figure 3.b shows an enlargement of the projections of the piercing points to the coordinate system of the camera  $R$ . Note that  $\{e_i\}_{i=1}^n$  are the (real) epipoles of cameras  $\{C_i\}_{i=1}^n$  in image  $\mathcal{I}_R$ ,  $v_R$  is the (virtual) epipole of the



“virtual camera”  $V$  in image  $\mathcal{I}_R$  (e.g., the black point in Figure 2.a), and  $\{v_i\}_{i=1}^n$  are the (virtual) epipoles of the virtual camera  $V$  in the (plane-warped) images  $\{\mathcal{I}_i\}_{i=1}^n$  (e.g., the black point in Figure 2.b or Figure 2.c). Further note that  $p_V$  is the projection of the piercing point of  $L$  with  $\Pi$ , which is also the axis point of the pencil-of-lines defined by the LOS  $L$  (e.g., the green point in Figure 2). This point is invariant to the positions of the cameras  $C_1, \dots, C_n$  and is thus common to all the images  $\mathcal{I}_R, \mathcal{I}_1, \dots, \mathcal{I}_n$ .

We now proceed to show how for any given axis point  $p_V$  and for any camera  $C_i$ , we can compute the geometric coordinate transformation between the line  $l_R$  in image  $\mathcal{I}_R$  (marked in blue) and the line  $l_i$  in image  $\mathcal{I}_i$  (marked in red), which will bring the two lines into alignment.

In all the derivations below we assume that points are expressed in homogeneous coordinates. The “virtual epipoles”  $v_i$  and  $v_R$  are known up to an arbitrary scale factor (these are specified by the user as the images of the scene point from which to synthesize the virtual view; see Section 4). The real epipole  $e_i$  is also known (e.g., can be estimated from  $\mathcal{I}_R$  and  $\mathcal{I}_i$ ). For any point  $p_R$  along the blue (reference) line we can find its corresponding point  $p_i$  along the green line:

$$p_i \cong (v_i \times p_V) \times (e_i \times p_R) \quad (1)$$

This equation can be easily verified by noting from Figure 3.b that the point  $p_i$  is on the intersection of the red line with the green line. This formulation is similar to the one derived in [3] for predicting the position of a scene point in a third image given its location in two other images. Eq. (1) can be rewritten in the following matrix form:

$$p_i \cong M_i p_R \quad (2)$$

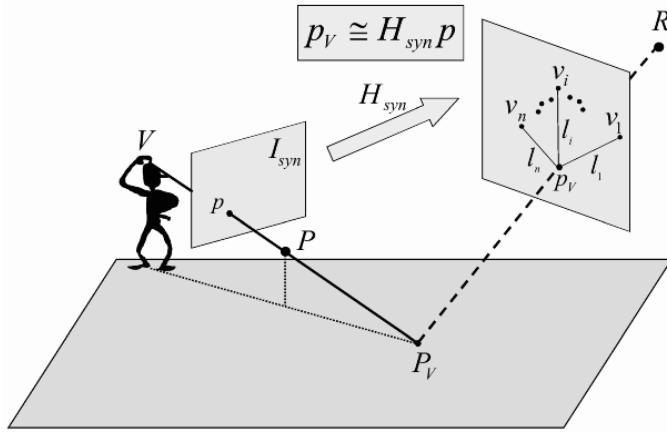
where  $M_i$  is a  $3 \times 3$  matrix whose elements can be expressed in terms of the components of the vectors  $v_i$ ,  $p_V$  and  $e_i$ .

To summarize, given all the real epipoles  $\{e_i\}_{i=1}^n$  (even if only up to *arbitrary* individual scale factors), and the virtual epipoles  $\{v_i\}_{i=1}^n$ , then for any axis point  $p_V$  we can compute the projective transformations  $\{M_i\}_{i=1}^n$  that bring the lines  $\{l_i\}_{i=1}^n$  into alignment with their reference line  $l_R$ .

A real example of alignment and stacking of such lines is shown in Figure 6. Several input images are shown with the projections of a LOS  $L$  highlighted in red (Figures 6.(a-f)). These lines were aligned and stacked (Figure 6.g) and the first color-consistent column is marked (which corresponds to the color of the wooden-gate in front of the can and cubes). The resulting representation in Figure 6.g bears resemblance to 2D slices of the 4D generalized disparity space of [16]. However, our representation is obtained by projectively aligning lines rather than by globally discretizing the disparity space.

## 4 Synthesizing the Virtual View

In this section we provide a summary of our algorithm. Let  $\mathcal{H}_{syn}$  be the homography relating the synthesized view with the reference view  $R$ . Since the position



**Fig. 4. Relating the virtual view to the reference view.** (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))

of the virtual camera is fixed (defined by the virtual epipoles), the only remaining degrees of freedom are the internal parameters of the “virtual camera” (e.g., its zoom) and its orientation (i.e., to which direction we wish the virtual camera to look). The user can select these by specifying  $\mathcal{H}_{syn}$ . For example, if  $\mathcal{H}_{syn}$  is chosen to be the identity matrix, then the synthesized view would be recovered in the coordinate system of the plane-aligned views  $\{\mathcal{I}_i\}_{i=0}^n$  (i.e.,  $\Pi$ -aligned with the reference view  $\mathcal{I}_R$ ), with the same internal calibration parameters as camera  $R$ .

For each pixel  $\mathbf{p} = (x, y, 1)^T$  in the synthesized image  $\mathcal{I}_{syn}$  do:

1. Let  $P$  denote the 3D scene point visible by the virtual camera center  $V$  at pixel  $\mathbf{p}$ . Then :  $p_V \cong \mathcal{H}_{syn}\mathbf{p}$  (See Figure 4).
2. For each of the plane-aligned images  $\mathcal{I}_1, \dots, \mathcal{I}_n$ ,  $l_i$  is the line connecting  $p_V$  with  $v_i$ ,  $1 \leq i \leq n$  (see below). Align all these  $n$  lines with  $l_R$ , the line connecting  $p_V$  with  $p_R$  by using the line-to-line transformation  $M_i$  defined in Eq. (2), and stack them vertically as shown in Figure 1.c.
3. Find the first color-consistent column, i.e., the column which is closest to the column of the synthetic epipoles  $v_R, v_1, \dots, v_n$  (see below for more details). Let “color” denote the color of this column.
4. Assign  $\mathcal{I}_{syn}(\mathbf{p}) := \text{“color”}$ .

Specifying the virtual epipoles  $\{v_i\}_{i=1}^n$  can be done by the user in one of the following ways: (i) The user can “pin-point” the selected scene point in all the input images, thus determining  $v_i$ ,  $1 \leq i \leq n$  explicitly. (ii) The user can “pin-point” the selected scene point in *two* images, and geometrically infer its position in all other images using trifocal constraints (which are also simple epipole-based constraints after plane alignment [5]). (iii) The user can “pin-point” the selected scene point in one image and use correlation-based techniques to find its corresponding point in each of the other images. All three options provide a

way for specifying a physically meaningful position of the virtual camera in an *uncalibrated* setting.

Color consistency within each column (step 3) is determined using color techniques proposed by [10]: Let  $A_{(n+1) \times 3}$  be a column of colors represented in YIQ color space. Denote by  $cov(A)$  the covariance matrix of  $A$ . We use the maximal eigenvalue of  $cov(A)$  (denoted by  $\lambda$ ) to indicate the degree of color consistency within that column. A high value for  $\lambda$  indicates large variance of colors in the column being evaluated, whereas a low value for  $\lambda$  indicates high color consistency (assuming a Lambertian model). A column whose  $\lambda$  is below a certain threshold is considered “color-consistent”. Since  $\lambda$  (the variance) is almost always nonzero, the actual color we select as a representative of a “color-consistent” column is the median color of that column. To robustify the synthesis process we prefer color consistent columns whose position along the LOS is spatially consistent with the chosen column of neighboring pixels. This can be thought of as a local “smoothing” constraint.

Large homogeneous regions in the scene may pose a problem for our algorithm. Projections of different physical points in a uniform region may be interpreted as projections of the same point because they share the same color. To reduce the effect of uniform regions we use a flagging scheme similar to that used in [14] for scene reconstruction. We automatically flag pixels in images already used in the synthesis to prevent them from being used again: If a column contains pixels flagged as previously used, it is not selected even if it is color consistent. The pixel scanning order is defined in  $\mathcal{I}_{syn}$  so that pixels corresponding to physical points closer to  $\Pi$  are evaluated before points farther away.

Since in many natural scenes the reference plane may contain large uniform regions (e.g. floor, grass etc.) we further add a preprocessing step which detects the ground plane after alignment. Thus, color consistent columns containing information from the ground plane will only be regarded as projections of a physical point on the LOS if they appear in the last column (i.e., the piercing point of LOS  $L$  with plane  $\Pi$ ).

## 5 A Practical Scenario (Avoiding Epipole Estimation)

So far we assumed the real epipoles of the cameras  $\{e_i\}_{i=1}^n$  are known. However, these may be difficult to estimate, especially when the cameras are widely separated and are looking at the scene from significantly different viewing positions. We next show how in some practical scenarios the need to estimate the inter-camera epipoles can be alleviated and replaced by simple image-based manipulations.

Such is the case when all the cameras are at the same height from the plane  $\Pi$ . In many real-world scenarios this may not be a very restrictive assumption. For example, consider a sports event. If we choose the ground plane to be the plane  $\Pi$ , then if all cameras are placed at the same height in the stadium (e.g., the same bench level) this assumption would be satisfied. Similarly, If a camera is attached to a mobile rig (e.g., a cart), and is moved around on the flat ground while taking pictures of the scene, then all these pictures would be taken from the same height

(even if the camera changes its internal parameters and its orientation as the rig moves). The same assumption holds if the camera is mounted on an airplane flying at fixed altitude from the ground or if the camera is attached to a crane which is parallel to the plane  $\Pi$  ( $\Pi$  can also be a wall or any other plane in the scene).

In all these cases the “cameras” are all coplanar. Referring back to Figure 3.b, we note that the epipole  $e_i$  lies along the line connecting the two (known) virtual epipoles  $v_i$  and  $v_R$ . Thus, there is only one degree of freedom in the epipole  $e_i$  (namely, where it is along that line). When all the real cameras are coplanar, then all the epipoles  $\{e_i\}_{i=1}^n$  lie on a single line in  $\mathcal{I}_R$ . This line is defined by the intersection of two planes: the plane defined by the real (coplanar) camera centers and the image-plane of the reference camera  $R$ . If we further *rectify* the image  $\mathcal{I}_R$  of  $R$ , then this line of real epipoles would go to infinity, and all the real epipoles  $\{e_i\}_{i=1}^n$  would now be uniquely defined by the virtual epipoles. For example, if  $e_i$  is known to be infinite (i.e., its third component  $w_{e_i} = 0$ ), then:

$$e_i \cong w_{v_R} v_i - w_{v_i} v_R \quad (3)$$

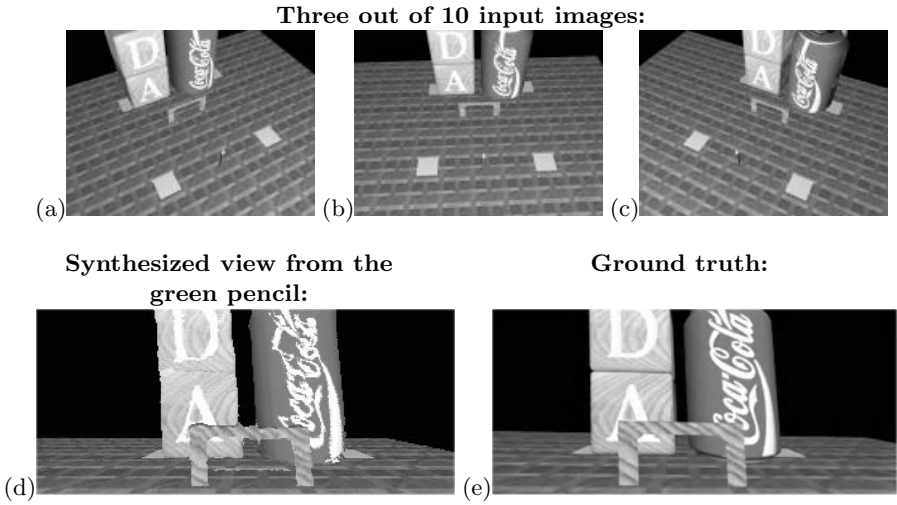
where  $v_i = [x_{v_i} y_{v_i} w_{v_i}]^T$ ,  $v_R = [x_{v_R} y_{v_R} w_{v_R}]^T$ , and  $e_i = [x_{e_i} y_{e_i} w_{e_i}]^T$ .

Substituting  $e_i$  in Eq. (1) with the expression of Eq. (3), we get a new matrix  $M_i$  whose 3rd row equals  $(0, 0, 1)$ . In other words  $M_i$  is now an affine transformation. This implies that the line-to-line transformations  $\{M_i\}_{i=1}^n$  reduce to *simple linear stretching* of the lines  $\{l_i\}_{i=1}^n$  relative to  $l_R$ .

The above result was obtained by rectifying the reference image  $\mathcal{I}_R$ . There are different possible ways of rectifying images. However, since in this particular case we assumed the cameras were of the same height from the plane  $\Pi$ , then the rectified image of  $\mathcal{I}_R$  should be a “nadir view” of the plane  $\Pi$ . Such rectification can be achieved based on the visual information in the image  $\mathcal{I}_R$  alone, without referring to any of the other images. The rectification can be obtained by manually imposing linear constraints that force parallel lines on the plane  $\Pi$  to become parallel after rectification of  $\mathcal{I}_R$ . This rectification step can be thought of as a type of weak scene-based calibration [19].

## 6 Results

In our experiments we avoided epipole estimation by maintaining constant camera height above the ground plane (see Section 5). We first tested our method on a synthetic scene with ground truth data. Figure 5.(a–c) shows three of the ten images rendered from a 3D graphic model (each of size  $800 \times 600$ ). These images were used to synthesize a view of the scene from the tip of the green pencil standing on the floor, looking in the direction of the wooden gate. The synthesized view is shown in Figure 5.d. Note the extreme change in viewpoint between the positions of the real cameras and the “virtual camera” (the green pencil). In all input images only the ground is visible through the wooden gate, while in the reconstructed image (Figure 5.d) parts of the Coke-can and the toy cube are visible beneath the gate. For comparison Figure 5.e shows the ground

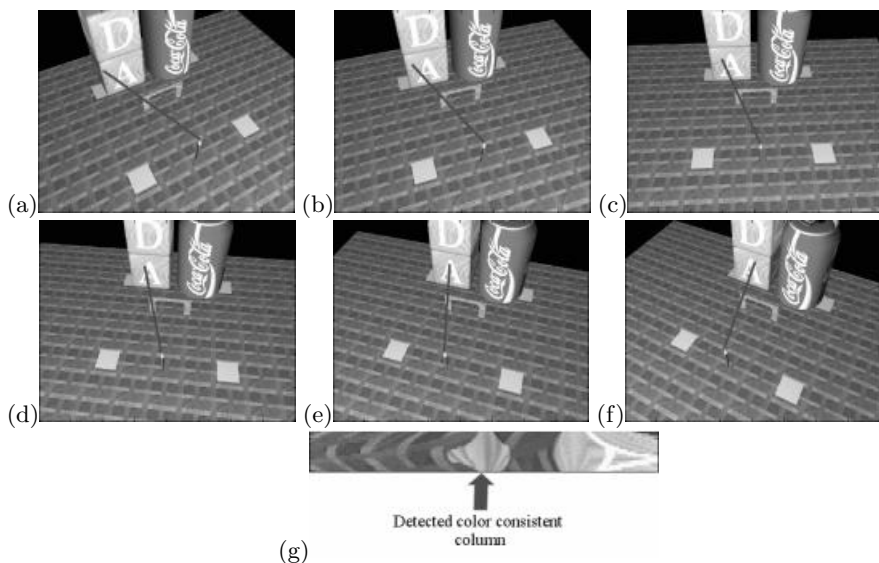


**Fig. 5.** Synthesis results for the Coke-Can sequence (see text). (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))

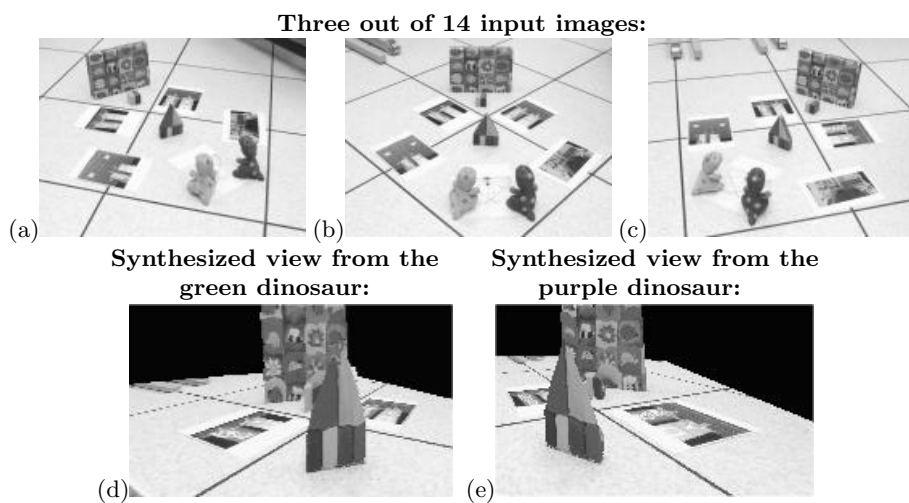
truth image rendered directly from the 3D model from the same viewpoint. This shows the geometric correctness of our synthesis. The differences in image quality are due to the fact that the image our algorithm synthesized (Figure 5.d) was generated from low-resolution input images (e.g., Figures 5.(a–c)) whereas the ground truth image was rendered directly from a perfect 3D model. See figure 6 for an illustration of our synthesis process.

Figure 7 shows the result of applying our algorithm to images of a real scene captured by an off the shelf digital camera. The camera was mounted on a tripod to guarantee constant height above the ground in all images. Three of the 14 input images (each of dimension  $640 \times 480$ ) are shown in Figure 7.(a–c). The algorithm was used to synthesize virtual views of the scene from two different scene points: Figure 7.d shows a reconstructed view of the scene from the tip of the green dinosaur’s nose using all 14 input images. Figure 7.e shows the reconstructed view of the scene from the tip of the purple dinosaur’s nose created using only 11 of the input images. Although both the folder and the green-and-yellow cube on the floor are fully visible in all input images, they are partially occluded in the synthesized views: The cube appears over the left shoulder of the triangle in the green dinosaur’s view, and over the right shoulder of the triangle in the purple dinosaur’s view as can be expected. Both the triangle and the cube occlude appropriate parts of the folder located behind them with respect to the viewpoint of the two dinosaurs.

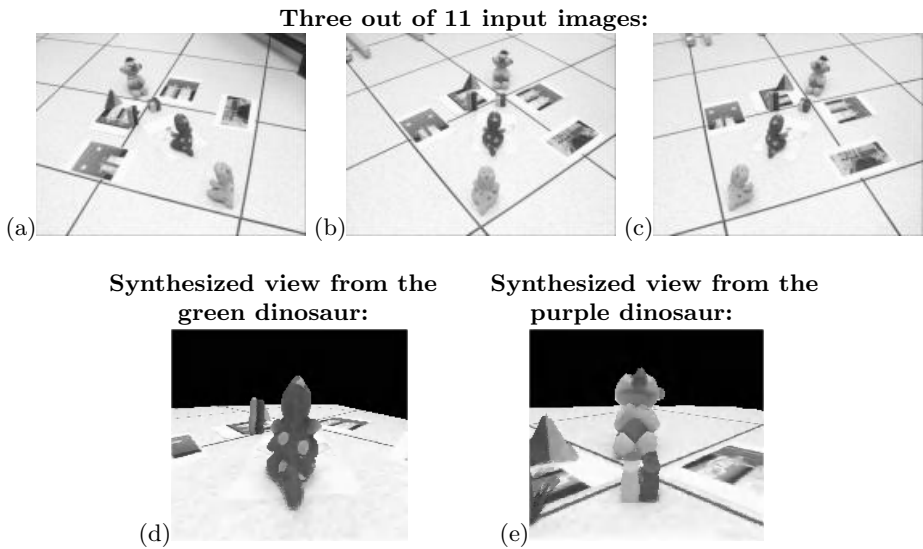
Figure 8 shows another real example only this time one dinosaur is placed in front of the other. Eleven images ( $560 \times 420$  pixels each) were used to synthesize a view from the green dinosaur’s nose (Figure 8.d), and only nine were enough to synthesize a view from the purple dinosaur’s nose Figure (8.e). It can be seen that the purple dinosaur completely occludes both the cube and the puppet from the



**Fig. 6. LOS color analysis (the Coke-Can sequence).** (a-f) Show the projection of a line of sight emerging from the virtual camera (at the tip of the green pencil) in 6 input images of the Coke-Can experiment (See Figure 5). The projected LOS is highlighted in red. (g) All ten lines were aligned and stacked. The first color-consistent column is indicated by a red arrow. Its color is that of the wooden gate, which is the first physical object along this LOS from the pencil. (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))



**Fig. 7. Synthesis results for the Folder sequence** (see text). (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))



**Fig. 8.** Synthesis results for the Puppet sequence (see text). (For color version see paper on attached CD or [www.cs.weizmann.ac.il/~vision/NVS.html](http://www.cs.weizmann.ac.il/~vision/NVS.html))

viewpoint of the green dinosaur (Figure 8.d). Also, the green and orange columns on the left almost totally occlude the red-yellow triangle (only the tip of its yellow side is visible). In contrast the purple dinosaur sees both clearly (Figure 8.e). The green smear on the floor at the bottom left side of the synthesized view in Figure 8.e is due to the fact that this floor-region was never visible in any of the input images. The puppet (Ernie) appears leaning backwards from the purple dinosaur's view because it is indeed leaning back as can be seen in Figures 8.(a,c).

## References

1. S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. In *IEEE Transactions on Visualization and Computer Graphics*, 1998.
2. A. Criminisi, I. Reid, and A. Zisserman. Duality, rigidity and planar parallax. In *ECCV*, Freiburg, 1998.
3. O. Faugeras and L. Robert. What can two images tell us about a third one? In *ECCV*, pages 485–492, 1994.
4. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, 1996.
5. M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *ECCV*, Freiburg, June 1998.
6. R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: a parallax based approach. In *Proc 12th ICPR*, pages 685–688, 1994.
7. K. N. Kutulakos. Approximate n-view stereo. In *ECCV*, 2000.
8. M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

9. P. J. Narayanan, P. W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *ICCV*, 1998.
10. M. Orchard and C. Bouman. Color quantization of images. In *IEEE Transactions on Signal Processing*, volume 39, 1991.
11. P. Rander, P.J. Narayanan, and T. Kanade. Virtualized reality: constructing time-varying virtual worlds from real events. In *Proc. IEEE Visualization*, pages 277–283, October 1997.
12. H. Saito and T. Kanade. Shape reconstruction in projective grid space from a large number of images. In *CVPR*, 1999.
13. H. Sawhney. 3D geometry from planar parallax. In *CVPR*, 1994.
14. S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR*, 1997.
15. A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *CVPR*, pages 483–489, 1994.
16. R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–524, January 1998.
17. W. Triggs. Plane + parallax, tensors, and factorization. In *ECCV*, pages 522–538, June 2000.
18. S. Vedula, P. Rander, H. Saito, and T. Kanade. Modeling, combining, and rendering dynamic real-world events from image sequences. In *Proceedings of the International Conference on Virtual Systems and Multimedia*, 1998.
19. D. Weinshall, P. Anandan, and M. Irani. From ordinal to euclidean reconstruction with partial scene calibration. In *Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, Freiburg, June 1998.
20. T. Werner, R.D. Hersch, and V. Hlaváč. Rendering real-world objects using view interpolation. In *ICCV*, pages 957–962, June 1995.



## Author Index

- Adams, N.J. IV-82  
Agarwal, S. IV-113  
Ahuja, N. IV-685  
Alvarez, L. I-721  
Anandan, P. II-883  
Aner, A. IV-388  
Ansar, A. IV-282  
Araujo, H. IV-237  
Attias, H. I-736  
Aubert, G. III-365  
Auf der Maur, D. III-180  
August, J. III-604  
Avidan, S. III-747
- Bajaj, C. III-517  
Barla, A. IV-20  
Barlaud, M. III-365  
Barnard, K. IV-97  
Barreto, J.P. IV-237  
Bartoli, A. II-340  
Bazin, P.-L. II-262  
Beal, M.J. I-736  
Belhumeur, P.N. III-869  
Bell, J.W. IV-358  
Belongie, S. III-21, III-531  
Bhat, K.S. I-551  
Bhotika, R. III-112  
Bischof, H. IV-761  
Black, M.J. I-476, I-692, I-784, IV-653  
Blake, A. I-645, IV-67  
Blanz, V. IV-3  
Bø, K. III-133  
Borenstein, E. II-109  
Botello, S. IV-560  
Boult, T. IV-590  
Boyer, E. IV-221  
Brand, M. I-707  
Bregler, C. I-801  
Bretzner, L. III-759  
Breuel, T.M. III-837  
Brodsky, T. I-614  
Buehler, C. III-885  
Buhmann, J.M. III-577
- Caenen, G. III-180  
Calderon, F. IV-560  
Carcassoni, M. I-266  
Carlsson, S. I-68, I-629, II-309  
Carneiro, G. I-282, I-297  
Caspi, Y. I-753  
Castellani, U. II-805  
Chang, J.T. II-31  
Chantler, M. III-289  
Charbonnier, P. I-492  
Chaumette, F. IV-312  
Chazelle, B. II-642  
Cheeseman, P. II-247  
Chefd'hotel, C. I-251  
Chellappa, R. II-277, III-681  
Chen, H. I-236  
Chen, Y. IV-546  
Chung, F. III-531  
Cipolla, R. II-155, II-852  
Clerc, M. II-495  
Cobzas, D. II-415  
Cohen, L.D. III-807, IV-531  
Cohen, M.F. III-885  
Collins, R. II-657  
Comaniciu, D. I-173, III-561  
Cootes, T.F. III-3, IV-621  
Cornelis, K. II-186  
Corpetti, T. I-676  
Costeira, J. II-232  
Coughlan, J.M. III-453  
Cremers, D. II-93  
Criminisi, A. I-508  
Crouzil, A. IV-252  
Cuff, X. III-408
- Daniilidis, K. II-140, IV-282  
Darrell, T. III-592, III-851  
David, P. III-698  
Davies, R.H. III-3  
Davis, L.S. I-18  
DeCarlo, D. IV-327  
Del Bue, A. III-561  
DeMenthon, D. III-698  
Deriche, R. I-251, I-721  
Deschamps, T. III-807  
Deutscher, J. IV-175

- Dick, A.R. II-852  
 Dickinson, S. III-759  
 Dobkin, D. II-642  
 Donato, G. III-21  
 Doorn, A.J. van I-158  
 Dornaika, F. IV-606  
 Drbohlav, O. II-46  
 Drew, M.S. IV-823  
 Duci, A. III-48  
 Duraiswami, R. III-698  
 Duygulu, P. IV-97  
 Dyer, C. IV-131  
  
 Eklundh, J.-O. III-469  
 Elder, J. IV-606  
 Ernst, F. II-217  
 Everingham, M. IV-34  
  
 Fablet, R. I-476  
 Faugeras, O. I-251, II-790  
 Favaro, P. II-18, II-735  
 Feldman, D. I-614  
 Felsberg, M. I-369  
 Ferrari, V. III-180  
 Ferreira, S.J. III-453  
 Ferrie, F.P. IV-267  
 Finkelstein, A. II-642  
 Finlayson, G.D. IV-823  
 Fisher, J.W. III-592, III-851  
 Fisher, R.B. IV-146  
 Fitzgibbon, A. III-304, III-487  
 Flandin, G. IV-312  
 Fleet, D.J. I-692, III-112  
 Florack, L. I-143, I-190  
 Forsyth, D.A. III-225, IV-97  
 Fossum, R. II-201  
 Fowlkes, C. III-531  
 Freitas, J.F.G. de IV-97  
 Freixenet, J. III-408  
 Frey, B.J. IV-715  
 Fua, P. II-325, II-704, III-163  
 Fukui, K. III-195  
 Funkhouser, T. II-642  
 Fusiello, A. II-805  
  
 Gangnet, M. I-645, I-661  
 Gao, X. IV-590  
 Gee, J. III-621  
 Georgescu, B. II-294  
 Gérard, O. III-807  
  
 Geusebroek, J.-M. I-99  
 Geyer, C. II-140  
 Giannopoulos, P. III-715  
 Giblin, P. II-718  
 Gilboa, G. I-399  
 Goldberger, J. IV-461  
 Goldenberg, R. I-461  
 Gomes, J. II-3  
 Gong, S. IV-670  
 Gool, L. Van II-170, II-186, II-572, II-837, III-180, IV-448  
 Gortler, S.J. III-885  
 Graham, J. IV-517  
 Granger, S. IV-418  
 Greenspan, H. IV-461  
 Greff, M. III-807  
 Grenander, U. I-37  
 Grossberg, M.D. I-220, IV-189  
 Guo, C.-e. III-240, IV-793  
 Gurdjos, P. IV-252  
  
 Hadjidemetriou, E. I-220  
 Han, F. III-502  
 Hancock, E.R. I-266, II-63, II-626, III-822  
 Hartley, R.I. II-433, II-447  
 Harville, M. III-543  
 Hassner, T. II-883  
 Hayman, E. III-469  
 Hebert, M. III-651, III-776  
 Heidrich, W. II-672  
 Hermes, L. III-577  
 Hertz, T. IV-776  
 Hillenbrand, U. III-791  
 Hirzinger, G. III-791  
 Hordley, S.D. IV-823  
 Huang, K. II-201  
 Hue, C. I-661  
 Huggins, P.S. I-384  
  
 Ieng, S.-S. I-492  
 Ilic, S. II-704  
 Irani, M. I-753, II-883  
 Isard, M. IV-175  
  
 Jagersand, M. II-415  
 Javed, O. IV-343  
 Jehan-Besson, S. III-365  
 Jelinek, D. II-463  
 Jepson, A.D. I-282, I-692  
 Jiang, G. I-537

- Jin, H. II-18  
 Jönsson, C. III-759  
 Jojic, N. I-736, IV-715  
  
 Kahl, F. II-447  
 Kam, A.H. IV-297  
 Kamberov, G. II-598  
 Kamberova, G. II-598  
 Kambhamettu, C. II-556, IV-206  
 Kaminski, J.Y. II-823  
 Kanatani, K. III-335  
 Kang, S.B. I-508, III-210  
 Kaucic, R. II-433  
 Kazhdan, M. II-642  
 Kender, J.R. IV-388, IV-403  
 Khosla, P.K. I-551  
 Kim, J. III-321  
 Kimia, B.B. II-718, III-731  
 Kimmel, R. I-461  
 Klein, P.N. III-731  
 Koenderink, J.J. I-158, IV-808  
 Kohlberger, T. II-93  
 Kolmogorov, V. III-65, III-82  
 Košecká, J. IV-476  
 Kozera, R. II-613  
 Kriegman, D.J. III-651, III-869  
 Krüger, V. IV-732  
 Kuehnel, F.O. II-247  
 Kuijper, A. I-143, I-190  
 Kutulakos, K.N. III-112  
  
 Lachaud, J.-O. III-438  
 Lafferty, J. III-776  
 Lasenby, J. I-524  
 Lazebnik, S. III-651  
 Leclerc, Y. III-163  
 Lee, A.B. I-328  
 Lee, M.-S. I-614  
 Lee, S.W. II-587  
 Leonardis, A. IV-761  
 Levin, A. II-399, III-635  
 Lhuillier, M. II-125  
 Li, S.Z. IV-67  
 Li, Y. III-210  
 Lin, S. III-210  
 Lindeberg, T. I-52, III-759  
 Liu, C. II-687  
 Liu, T. IV-403  
 Liu, X. I-37  
 Liu, Y. II-657  
  
 Lo, B.P.L. III-381  
 Loy, G. I-358  
 Lu, W. IV-297  
 Luong, Q.-T. III-163  
  
 Ma, Y. II-201  
 McCormick, J. IV-175  
 Maciel, J. II-232  
 Mahamud, S. III-776  
 Maki, A. III-195  
 Malik, J. I-312, III-531, III-666  
 Malis, E. IV-433  
 Malladi, R. I-343  
 Malsburg, C. von der IV-747  
 Maluf, D.A. II-247  
 Manning, R. IV-131  
 Markham, R. IV-502  
 Marroquin, J.L. I-113, IV-560  
 Martinec, D. II-355  
 Martí, J. III-408  
 Mathiassen, J.R. III-133  
 Maybank, S. IV-373  
 Mayer, A. IV-461  
 McGunnigle, G. III-289  
 McMillan, L. III-885  
 Medioni, G. III-423  
 Meer, P. I-236, II-294  
 Mémin, É. I-676  
 Mikołajczyk, K. I-128  
 Mindru, F. IV-448  
 Mirmehdi, M. IV-502  
 Mitran, M. IV-267  
 Mittal, A. I-18  
 Mitter, S. III-48  
 Mojsilovic, A. II-3  
 Moons, T. IV-448  
 Mori, G. III-666  
 Morris, R.D. II-247  
 Muller, H. IV-34  
 Muñoz, X. III-408  
 Murino, V. II-805  
 Murray, D.W. I-82  
  
 Nakashima, A. III-195  
 Narasimhan, S.G. III-148, IV-636  
 Nayar, S.K. I-220, I-508, III-148, IV-189, IV-636  
 Ng, J. IV-670  
 Ng, T.K. I-205

- Nicolescu, M. III-423  
 Noakes, L. II-613  
  
 Odone, F. IV-20  
 Oliensis, J. II-383  
 Osareh, A. IV-502  
 Overveld, K. van II-217  
  
 Pajdla, T. II-355  
 Pal, C. IV-715  
 Papadopoulos, T. I-721  
 Paragios, N. II-78, II-775  
 Pavel, M. IV-776  
 Payrissat, R. IV-252  
 Pece, A.E.C. I-3  
 Pedersen, K.S. I-328  
 Pennec, X. IV-418  
 Pérez, P. I-645, I-661, I-676  
 Perona, P. II-759  
 Petrou, M. III-289  
 Plaenkers, R. II-325  
 Pollefeys, M. II-186, II-837  
 Ponce, J. III-651  
 Pont, S.C. IV-808  
 Popović, J. I-551  
 Prados, E. II-790  
 Prince, J.L. IV-575  
  
 Qian, G. II-277  
 Quan, L. I-537, II-125  
  
 Raba, D. III-408  
 Ragheb, H. II-626  
 Ramesh, V. I-173, II-775, III-561, IV-590  
 Ravve, I. I-343  
 Ren, X. I-312  
 Richard, F. IV-531  
 Ringer, M. I-524  
 Rivera, M. I-113, III-621  
 Rivlin, E. I-461  
 Robertson, D.P. II-155  
 Robles-Kelly, A. II-63  
 Rogers, M. IV-517  
 Romdhani, S. IV-3  
 Ronchetti, L. II-805  
 Ronfard, R. IV-700  
 Rosin, P.L. II-746  
 Roth, D. IV-113, IV-685  
 Rother, C. II-309  
 Rousson, M. II-78, II-775  
  
 Rouy, E. II-790  
 Rudzsky, M. I-461  
  
 Samaras, D. III-272  
 Samet, H. III-698  
 Sánchez, J. I-721  
 Šára, R. II-46, III-900  
 Savarese, S. II-759  
 Sawhney, H.S. I-599  
 Schaffalitzky, F. I-414  
 Scharstein, D. II-525  
 Schiele, B. IV-49  
 Schmid, C. I-128, III-651, IV-700  
 Schmidt, M. III-289  
 Schnörr, C. II-93  
 Schwartz, S. I-173  
 Schweitzer, H. IV-358, IV-491  
 Sebastian, T.B. III-731  
 Seitz, S.M. I-551  
 Sénégas, J. III-97  
 Sethi, A. III-651  
 Shah, M. IV-343  
 Shakhnarovich, G. III-851  
 Sharp, G.C. II-587  
 Shashua, A. II-399, III-635  
 Shechtman, E. I-753  
 Shental, N. IV-776  
 Shokoufandeh, A. III-759  
 Shum, H.-Y. II-510, II-687, III-210, IV-67  
 Sidenbladh, H. I-784  
 Siebel, N.T. IV-373  
 Sigal, L. I-784  
 Skavhaug, A. III-133  
 Skočaj, D. IV-761  
 Smelyansky, V.N. II-247  
 Smeulders, A.W.M. I-99  
 Sminchisescu, C. I-566, I-769  
 Soatto, S. II-735, III-32, III-48  
 Sochen, N.A. I-399  
 Sommer, G. I-369  
 Speyer, G. I-432  
 Srivastava, A. I-37  
 Strecha, C. II-170  
 Sturm, P. II-867, IV-221  
 Sullivan, J. I-629  
 Sun, J. II-510  
 Sung, K.K. I-205  
 Swaminathan, R. I-508  
 Szeliski, R. I-508, II-525

- Tam, R. II-672  
 Tarel, J.-P. I-492  
 Taton, B. III-438  
 Taylor, C.J. II-463, III-3, IV-621  
 Teicher, M. II-823  
 Tell, D. I-68  
 Terzopoulos, D. I-447  
 Thacker, N. IV-621  
 Thomas, B. IV-34, IV-502  
 Tohka, J. III-350  
 Tong, X. III-210  
 Tordoff, B. I-82  
 Torr, P.H.S. II-852  
 Torre, F. De la IV-653  
 Torresani, L. I-801  
 Torsello, A. III-822  
 Trajkovic, M. I-614  
 Triggs, B. I-566, I-769, IV-700  
 Tschumperlé, D. I-251  
 Tsin, Y. II-657  
 Tsui, H.-t. I-537  
 Tu, Z. III-393, III-502  
 Twining, C.J. III-3  
  
 Ullman, S. II-109  
  
 Varma, M. III-255  
 Vasconcelos, N. I-297  
 Vasilescu, M.A.O. I-447  
 Veltkamp, R.C. III-715  
 Vemuri, B.C. IV-546, IV-560  
 Verbiest, F. II-837  
 Vermaak, J. I-645, I-661  
 Verri, A. IV-20  
 Vetter, T. IV-3  
 Vézien, J.-M. II-262  
 Vidal, R. II-383  
 Vogel, J. IV-49  
  
 Wang, B. I-205  
 Wang, C. III-148  
 Wang, S.-C. I-583  
 Wang, Y. I-583, III-272, IV-793  
 Wang, Z. IV-546  
 Waterton, J.C. III-3  
 Wehe, D.K. II-587  
 Weijer, J. van de I-99  
 Weinshall, D. I-614, IV-776  
  
 Werman, M. I-432  
 Werner, T. II-541  
 Wexler, Y. III-487  
 Wieghardt, J. IV-747  
 Wilczkowiak, M. IV-221  
 Wilinski, P. II-217  
 Williams, C.K.I. IV-82  
 Wolf, L. II-370  
 Worrall, A.D. I-3  
 Wu, F. IV-358  
 Wu, Y.N. III-240, IV-793  
 Würtz, R.P. IV-747  
 Wyngaerd, J. Vanden II-572  
  
 Yang, G.-Z. III-381  
 Yang, M.-H. IV-685  
 Yang, R. II-479  
 Yau, W.-Y. IV-297  
 Yezzi, A.J. III-32, III-48, IV-575  
 Yu, Y. II-31  
 Yu, Z. III-517  
  
 Zabih, R. III-65, III-82, III-321  
 Zalesny, A. III-180  
 Zeevi, Y.Y. I-399  
 Zelinsky, A. I-358  
 Zhang, C. II-687  
 Zhang, H. IV-67  
 Zhang, W. IV-476  
 Zhang, Y. II-556  
 Zhang, Z. II-479, IV-67, IV-161  
 Zhao, W.Y. I-599  
 Zheng, N.-N. II-510  
 Zhou, S. III-681, IV-732  
 Zhou, W. IV-206  
 Zhu, L. IV-67  
 Zhu, S.-C. I-583, III-240, III-393, III-502, IV-793  
 Zhu, Y. I-173  
 Zhu, Y. I-583  
 Zickler, T. III-869  
 Zisserman, A. I-414, I-537, II-541, III-255, III-304, III-487  
 Zöllner, T. III-577  
 Zomet, A. II-370  
 Zucker, S.W. I-384  
 Žunić, J. II-746