

LNCS 3540

Heikki Kalviainen  
Jussi Parkkinen  
Arto Kaarna (Eds.)

# Image Analysis

14th Scandinavian Conference, SCIA 2005  
Joensuu, Finland, June 2005  
Proceedings



**SCIA<sup>2005</sup>**

June 19-22  
Joensuu, Finland



Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Heikki Kalviainen Jussi Parkkinen  
Arto Kaarna (Eds.)

# Image Analysis

14th Scandinavian Conference, SCIA 2005  
Joensuu, Finland, June 19-22, 2005  
Proceedings

**Volume Editors**

**Heikki Kalviainen**

**Arto Kaarna**

Lappeenranta University of Technology, Department of Information Technology

P.O. Box 20, 53851 Lappeenranta, Finland

E-mail: {heikki.kalviainen, arto.kaarna}@lut.fi

**Jussi Parkkinen**

University of Joensuu, Department of Computer Science

P.O. Box 111, 80101 Joensuu, Finland

E-mail: jussi.parkkinen@cs.joensuu.fi

Library of Congress Control Number: 2005927489

CR Subject Classification (1998): I.4, I.5, I.3

ISSN 0302-9743

ISBN-10 3-540-26320-9 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-26320-3 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

[springeronline.com](http://springeronline.com)

© Springer-Verlag Berlin Heidelberg 2005

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 11499145 06/3142 5 4 3 2 1 0

# Preface

This proceedings volume collects the scientific presentations of the Scandinavian Conference on Image Analysis, SCIA 2005, which was held at the University of Joensuu, Finland, June 19–22, 2005. The conference was the fourteenth in the series of biennial conferences started in 1980. The name of the series reflects the fact that the conferences are organized in the Nordic (Scandinavian) countries, following the cycle Sweden, Finland, Denmark, and Norway. The event itself has always been international in its participants and presentations.

Today there are many conferences in the fields related to SCIA. In this situation our goal is to keep up the reputation for the high quality and friendly environment of SCIA. We hope that participants feel that it's worth attending the conference. Therefore, both the scientific and social program were designed to support the best features of a scientific meeting: to get new ideas for research and to have the possibility to exchange thoughts with fellow scientists.

To fulfill the above-mentioned goals, the conference was a single-track event. This meant that a higher percentage of the papers than in earlier SCIAs were presented as posters. We hope that this gave the participants better chances to follow the presentations that they were interested in. SCIA 2005 attracted a record number of submissions: 236 manuscripts. From these, 124 were accepted: 31 oral presentations and 93 poster presentations. This led to an acceptance rate of 53%. The program included also six plenary presentations and three tutorials.

The conference city, Joensuu, is located in the lake region of Finland, where nature is an elementary part of people's lives. In its history Joensuu was a border region between Russia and Sweden. This can be seen in the local culture. These features were taken into account while designing the social program which included, among other events, a sauna by the lake, a conference dinner in the Orthodox Monastery in Valamo, and a postconference tour by ship through Lake Ladoga to St. Petersburg.

As always, there were a number of volunteers involved in the organizing work. We express our sincere thanks to all the invited speakers, the scientists who presented their papers, all participants, and the volunteers for making this conference possible.

We hope that all attendees had a successful and enjoyable conference.

June 2005

Jussi Parkkinen, Heikki Kälviäinen,  
and Markku Hauta-Kasari

# Organization



SCIA 2005 was organized jointly by the Department of Computer Science at the University of Joensuu and the Department of Information Technology at Lappeenranta University of Technology.



<http://cs.joensuu.fi>



<http://www.it.lut.fi>

## Executive Committee

General Chair	Jussi Parkkinen, University of Joensuu (Finland)
Program Chair	Heikki Kälviäinen, Lappeenranta University of Technology (Finland)
Local Chair	Markku Hauta-Kasari, University of Joensuu (Finland)

## Program Committee

- Prof. Heikki Kälviäinen, Chairman, Lappeenranta University of Technology (Finland)  
Prof. Pasi Fränti, University of Joensuu (Finland)  
Prof. Erkki Oja, Helsinki University of Technology (Finland)  
Prof. Matti Pietikäinen, University of Oulu (Finland)  
Prof. Ari Visa, Tampere University of Technology (Finland)  
Prof. Knut Conradi, Technical University of Denmark (Denmark)  
Dr. Ingela Nyström, Uppsala University (Sweden)  
Dr. Arnt-Børre Salberg, Institute of Marine Research, Tromsø (Norway)

## Reviewers

Evgeny Ageenko  
Mats Andersson  
Jaakko Astola  
Ivar Austvoll  
Ewert Bengtsson  
Josef Bigun  
Marten Bjorkman  
Magnus Borga  
Gunilla Borgefors  
Vladimir Botchko  
Sami Brandt  
Anders Brun  
Barbara Caputo  
Henrik Christensen  
Jan-Olof Eklundh  
Olle Eriksson  
Bjarne Ersbll  
Jeppe Frisvad  
Pasi Fränti  
Tapios Grönfors  
Jon Hardeberg  
Markku Hauta-Kasari  
Janne Heikkilä  
Jukka Heikkonen  
Jaakko Hollmen  
Timo Honkela  
Heikki Huttunen  
Jukka Iivarinen  
Peter Johansen  
Martti Juhola  
Timo Jääskeläinen  
Arto Kaarna  
Heikki Kälviäinen  
Joni Kämäräinen  
Sami Kaski  
Hans Knutsson  
Pasi Koikkalainen  
Alexander Kolesnikov  
Markus Koskela  
Timo Kostiainen  
Danica Kragic  
Björn Kruse  
Pauli Kuosmanen  
Ville Kyrki  
Jorma Laaksonen  
Jouko Lampinen  
Vuokko Lantz  
Rasmus Larsen  
Lasse Lensu  
Anssi Lensu  
Reiner Lenz  
Joakim Lindblad  
Birgitta Martinkuppi  
Jarno Mielikäinen  
Thomas B. Moeslund  
Jouni Mykkänen  
Topi Mäenpää  
Allan Nielsen  
Henning Nielsen  
Matti Niskanen  
Bo Nordin  
Ingela Nyström  
Erkki Oja  
Timo Ojala  
Oleg Okun  
Soren Olsen  
Jussi Parkkinen  
Arthur E.C. Pece  
Kim Steenstrup Pedersen  
Markus Peura  
Matti Pietikäinen  
Tapios Repo  
Mauno Ronkko  
Ulla Ruotsalainen  
Juha Röning  
Arnt-Børre Salberg  
Pekka Sangi  
Tapios Seppänen  
Andreas Sigfridsson  
Olli Silven  
Ida-Maria Sintorn  
Olli Simula  
Örjan Smedby  
Jon Sporring  
Mikkel B. Stegmann  
Robin Strand

Stina Svensson  
Toni Tamminen  
Pekka Toivanen  
Aki Vehtari  
Antanas Verikas  
Erik Vidholm

Jan Voracek  
Carolina Wählby  
Felix Wehrmann  
Andreas Wrangsjö  
Kalle Åström  
Tor Øigård

## Sponsoring Organizations

Joensuun Yliopisto



Pattern Recognition Society of Finland

Suomen hahmontunnistustutkimuksen seura ry  
Pattern Recognition Society of Finland



The International Association for Pattern Recognition



# Table of Contents

## Invited Talk

Biometric Recognition: How Do I Know Who You Are?

1

## Image Segmentation and Understanding

Hierarchical Cell Structures for Segmentation of Voxel Images

6

Paving the Way for Image Understanding: A New Kind of Image Decomposition Is Desired

17

Levelset and B-Spline Deformable Model Techniques for Image Segmentation: A Pragmatic Comparative Study

25

Steerable Semi-automatic Segmentation of Textured Images

35

MSCC: Maximally Stable Corner Clusters

45

## Invited Talk

Spectral Imaging Technique for Visualizing the Invisible Information

55

## Color Image Processing

Bayesian Image Segmentation Using MRF's Combined with Hierarchical Prior Models

65

Feature Extraction for Oil Spill Detection Based on SAR Images

75

## XII Table of Contents

Light Field Reconstruction Using a Planar Patch Model	85
Spectral Estimation of Skin Color with Foundation Makeup	95
Color Measurements with a Consumer Digital Camera Using Spectral Estimation Techniques	105
<b>Invited Talk</b>	
Image Analysis with Local Binary Patterns	115
<b>Applications</b>	
Object Evidence Extraction Using Simple Gabor Features and Statistical Ranking	119
Dynamically Visual Learning for People Identification with Sparsely Distributed Cameras	130
3D-Connected Components Analysis for Traffic Monitoring in Image Sequences Acquired from a Helicopter	141
Joint Modeling of Facial Expression and Shape from Video	151
Development of Direct Manipulation Interface for Collaborative VR/MR Worspace	161
Estimating Camera Position and Posture by Using Feature Landmark Database	171

**Invited Talk**

- Geometrical Computer Vision from Chasles to Today 182

**Theory**

- The S-Kernel and a Symmetry Measure Based on Correlation 184

- Training Cellular Automata for Image Processing 195

- Functional 2D Procrustes Shape Analysis 205

- The Descriptive Approach to Image Analysis. Current State and Prospects 214

- A New Method for Affine Registration of Images and Point Sets 224

**Invited Talk**

- Joint Spatial-Temporal Color Demosaicking 235

**Medical Image Processing**

- Shape Based Identification of Proteins in Volume Images 253

- Thickness Estimation of Discrete Tree-Like Tubular Objects: Application to Vessel Quantification 263

- Segmentation of Multimodal MRI of Hippocampus Using 3D Grey-Level Morphology Combined with Artificial Neural Networks 272

- Combined Segmentation and Tracking of Neural Stem-Cells 282

XIV Table of Contents

Morphons: Paint on Priors and Elastic Canvas for Segmentation and Registration

292

**Image Compression**

Efficient 1-Pass Prediction for Volume Compression

302

Lossless Compression of Map Contours by Context Tree Modeling of Chain Codes

312

Optimal Estimation of Homogeneous Vectors

322

Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction

333

**Invited Talk**

A Memory Architecture and Contextual Reasoning Framework for Cognitive Vision

343

**Valamo Special Session in Digitizing of Cultural Heritage**

Synthesizing the Artistic Effects of Ink Painting

359

Application of Spectral Information to Investigate Historical Materials - Detection of Metameric Color Area in Icon Images -

369

An Approach to Digital Archiving of Art Paintings

379

**Poster Presentations 1: Image Analysis, Computer Vision, Machine Vision, and Applications**

Preferential Spectral Image Quality Model	389
Three-Dimensional Measurement System Using a Cylindrical Mirror	399
Mottling Assessment of Solid Printed Areas and Its Correlation to Perceived Uniformity	409
In Situ Detection and Identification of Microorganisms at Single Colony Resolution Using Spectral Imaging Technique	419
Dolphins Who's Who: A Statistical Perspective	429
Local Shape Modelling Using Warplets	439
Learning Based System for Detection and Tracking of Vehicles	449
Texture Analysis for Stroke Classification in Infrared Reflectogramms	459
Problems Related to Automatic Nipple Extraction	470
A Novel Robust Tube Detection Filter for 3D Centerline Extraction	481
Reconstruction of Probability Density Functions from Channel Representations	491
Non-rigid Registration Using Morphons	501

## XVI Table of Contents

Hybridization of the Ant Colony Optimization with the K-Means Algorithm for Clustering	511
Incremental Locally Linear Embedding Algorithm	521
On Aligning Sets of Points Reconstructed from Uncalibrated Affine Cameras	531
A New Class of Learnable Detectors for Categorisation	541
Overlapping Constraint for Variational Surface Reconstruction	551
Integation Methods of Model-Free Features for 3D Tracking	557
Probabilistic Model-Based Background Subtraction	567
A Bayesian Approach for Affine Auto-calibration	577
Shape-Based Co-occurrence Matrices for Defect Classification	588
Complex Correlation Statistic for Dense Stereoscopic Matching	598
Reconstruction from Planar Motion Image Sequences with Applications for Autonomous Vehicles	609
Stereo Tracking Error Analysis by Comparison with an Electromagnetic Tracking Device	619
Building Detection from Mobile Imagery Using Informative SIFT Descriptors	629

Perception-Action Based Object Detection from Local Descriptor Combination and Reinforcement Learning	639
Use of Quadrature Filters for Detection of Stellate Lesions in Mammograms	649
A Study of the Yosemite Sequence Used as a Test Sequence for Estimation of Optical Flow	659
A Versatile Model-Based Visibility Measure for Geometric Primitives	669
Pose Estimation of Randomly Organized Stator Housings	679
3D Reconstruction of Metallic Surfaces by Photopolarimetric Analysis	689
Inferring and Enforcing Geometrical Constraints on a 3D Model for Building Reconstruction	699
Aligning Shapes by Minimising the Description Length	709
Segmentation of Medical Images Using Three-Dimensional Active Shape Models	719
A Novel Algorithm for Fitting 3-D Active Appearance Models: Applications to Cardiac MRI Segmentation	729
Decision Support System for the Diagnosis of Parkinson's Disease	740
Polygon Mesh Generation of Branching Structures	750
Joint Analysis of Multiple Mammographic Views in CAD Systems for Breast Cancer Detection	760

## XVIII Table of Contents

Approximated Classification in Interactive Facial Image Retrieval	770
Eye-Movements as a Biometric	780
Inverse Global Illumination Rendering for Dense Estimation of Surface Reflectance Properties	790
Multimodal Automatic Indexing for Broadcast Soccer Video	802
Evaluation of the Effect of Input Stimuli on the Quality of Orientation Maps Produced Through Self Organization	810
Modeling Inaccurate Perception: Desynchronization Issues of a Pattern Recognition Neural Network	821
A High-Reliability, High-Resolution Method for Land Cover Classification into Forest and Non-forest	831
<b>Poster Presentations 2: Pattern Recognition, Image Processing, and Applications</b>	
Invariance in Kernel Methods by Haar-Integration Kernels	841
Exemplar Based Recognition of Visual Shapes	852
Object Localization with Boosting and Weak Supervision for Generic Object Recognition	862
Clustering Based on Principal Curve	872
Block-Based Methods for Image Retrieval Using Local Binary Patterns	882

Enhanced Fourier Shape Descriptor Using Zero-Padding	892
Color-Based Classification of Natural Rock Images Using Classifier Combinations	901
Fast Guaranteed Polygonal Approximations of Closed Digital Curves	910
Fast Manifold Learning Based on Riemannian Normal Coordinates	920
TIPS: On Finding a Tight Isothetic Polygonal Shape Covering a 2D Object	930
Approximate Steerability of Gabor Filters for Feature Detection	940
Nonlinear Dimensionality Reduction Using Circuit Models	950
Mapping Perceptual Texture Similarity for Image Retrieval	960
Toward Automatic Motor Condition Diagnosis	970
Improving K-Means by Outlier Removal	978
Maximal Digital Straight Segments and Convergence of Discrete Geometric Estimators	988
Improving the Maximum-Likelihood Co-occurrence Classifier: A Study on Classification of Inhomogeneous Rock Images	998
The Tangent Kernel Approach to Illumination-Robust Texture Classification	1009

XX Table of Contents

Tissue Models and Speckle Reduction in Medical Ultrasound Images	1017
A Comparison Among Distances Based on Neighborhood Sequences in Regular Grids	1027
Restoration of Multitemporal Short-Exposure Astronomical Images	1037
A Comparative Study of Angular Extrapolation in Sinogram and Stackgram Domains for Limited Angle Tomography	1047
A Classification of Centres of Maximal Balls in $\mathbb{Z}^3$	1057
3D Object Volume Measurement Using Freehand Ultrasound	1066
Modeling, Evaluation and Control of a Road Image Processing Chain	1076
A Graph Representation of Filter Networks	1086
Optimal Ratio of Lamé Moduli with Application to Motion of Jupiter Storms	1096
Extraction and Removal of Layers from Map Imagery Data	1107
Tensor Processing for Texture and Colour Segmentation	1117
Cerebrovascular Segmentation by Accurate Probabilistic Modeling of TOF-MRA Images	1128
MGRF Controlled Stochastic Deformable Model	1138

Dissolved Organic Matters Impact on Colour Reconstruction in Underwater Images	1148
Denoising of Time-Density Data in Digital Subtraction Angiography	1157
The Use of Image Smoothness Estimates in Speeding Up Fractal Image Compression	1167
DCT Based High Quality Image Compression	1177
Optimal Encoding of Vector Data with Polygonal Approximation and Vertex Quantization	1186
Image Compression Using Adaptive Variable Degree Variable Segment Length Chebyshev Polynomials	1196
Linear Hashtable Method Predicted Hexagonal Search Algorithm with Spatial Related Criterion	1208
Fractal Dimension Analysis and Statistical Processing of Paper Surface Images Towards Surface Roughness Measurement	1218
Estimation of Critical Parameters in Concrete Production Using Multispectral Vision Technology	1228
Automated Multiple View Inspection Based on Uncalibrated Image Sequences	1238
Interactive 3-D Modeling System Using a Hand-Held Video Camera	1248

**XXII Table of Contents**

Automatic Segmentation of the Prostate from Ultrasound Data Using  
Feature-Based Self Organizing Map

1259

**Author Index**

1267

# Biometric Recognition: How Do I Know Who You Are?

Anil K. Jain

Department of Computer Science and Engineering,  
3115 Engineering Building, Michigan State University,  
East Lansing, MI 48824, USA  
[jain@cse.msu.edu](mailto:jain@cse.msu.edu)  
<http://biometrics.cse.msu.edu>

## Extended Abstract

A wide variety of systems require reliable personal recognition schemes to either confirm or determine the identity of an individual requesting their services. The purpose of such schemes is to ensure that the rendered services are accessed only by a legitimate user, and not anyone else. Examples of such applications include secure access to buildings, computer systems, laptops, cellular phones and ATMs. In the absence of robust person recognition schemes, these systems are vulnerable to the wiles of an impostor. Biometric recognition, or simply biometrics, refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics. By using biometrics it is possible to confirm or establish an individual's identity based on who she is, rather than by what she possesses (e.g., an ID card) or what she remembers (e.g., a password). Although biometrics emerged from its extensive use in law enforcement to identify criminals, i.e., forensics, it is being increasingly used today to carry out person recognition in a large number of civilian applications (e.g., national ID card, e-passport and smart cards) [1], [2]. Most of the emerging applications can be attributed to increased security threats as well as fraud associated with various financial transactions (e.g., credit cards).

What biological measurements qualify to be a biometric? Any human physiological and/or behavioral characteristic can be used as a biometric characteristic as long as it satisfies the following requirements:

- Universality: each person should have the characteristic;
- Distinctiveness: any two persons should be sufficiently different in terms of the characteristic;
- Permanence: the characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time;
- Collectability: the characteristic can be measured quantitatively.

However, in a practical biometric system (i.e., a system that employs biometrics for person recognition), there are a number of other issues that should be considered, including:

- Performance, which refers to the achievable recognition accuracy and speed, the resources required to achieve the desired performance, as well as the operational and environmental factors that affect the performance;
- Acceptability, which indicates the extent to which people are willing to accept the use of a particular biometric identifier (characteristic) in their daily lives;
- Circumvention, which reflects how easily the system can be fooled using fraudulent methods.

A practical biometric system should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, be easy to use and be sufficiently robust to various fraudulent methods and attacks on the system. Among the various biometric measurements in use, fingerprint-based systems [3] and face recognition systems [4] are the most popular.

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. Depending on the application context, a biometric system may operate either in a verification mode or an identification mode [5]. A biometric system is designed using the following four main modules: (i) sensor module, (ii) feature extraction module, (iii) matcher module, and (iv) system database module.

Two samples of the same biometric characteristic from the same person (e.g., two impressions of a user's right index finger) are not exactly the same due to imperfect imaging conditions (e.g., sensor noise), changes in the user's physiological or behavioral characteristics (e.g., cuts and bruises on the finger), ambient conditions (e.g., temperature and humidity) and user's interaction with the sensor (e.g., finger placement). In other words, biometric signals have a large..... Therefore, the response of a biometric matching system is a matching score that quantifies the similarity between the input and the database template representation. Higher score indicates that the system is more certain that the two biometric measurements come from the same person. The system decision is regulated by the threshold: pairs of biometric samples generating scores higher than or equal to the threshold are inferred as mate pairs (i.e., belonging to the same person); pairs of biometric samples generating scores lower than the threshold are inferred as non-mate pairs (i.e., belonging to different persons). A biometric verification system makes two types of errors: (i) mistaking biometric measurements from two different persons to be from the same person (called.....), and (ii) mistaking two biometric measurements from the same person to be from two different persons (called.....). These two types of errors are often termed as..... and....., respectively.

Deployment of biometric systems in various civilian applications does not imply that biometric recognition is a fully solved problem. Table 1 presents the state-of-the-art error rates of three popular biometric traits. It is clear that there

**Table 1.** State-of-the-art error rates associated with fingerprint, face and voice biometric systems. Note that the accuracy estimates of biometric systems are dependent on a number of test conditions

	Test	Test Parameter	False Reject Rate	False Accept Rate
Fingerprint	FVC 2004 [6]	Exaggerated skin distortion, rotation, skin conditions	2%	2%
Face	FRVT 2002 [7]	Enrollment and test images were collected in indoor environment and could be on different days	10%	1%
Voice	NIST 2004 [8]	Text independent, multi-lingual	5-10%	2-5%

is a plenty of scope for improvement in the performance of biometric systems. We not only need to address issues related to reducing error rates, but we also need to look at ways to enhance the usability of biometric systems and address the . . . . . issue.

Biometric systems that operate using any single biometric characteristic have the following limitations: (i) noise in sensed data, (ii) intra-class variations, (iii) lack of distinctiveness [9], (iv) non-universality, and (v) spoof attacks. Some of the limitations imposed by unimodal biometric systems can be overcome by using multiple biometric modalities (such as face and fingerprint of a person or multiple fingers of a person). Such systems, known as multimodal biometric systems, are expected to be more reliable due to the presence of multiple, independent pieces of evidence [10]. These systems are also able to meet the stringent performance requirements imposed by various applications [11]. Multimodal biometric systems address the problem of non-universality, since multiple traits ensure sufficient population coverage. Further, multimodal biometric systems provide anti-spoofing measures by making it difficult for an intruder to simultaneously spoof the multiple biometric traits of a legitimate user. By asking the user to present a random subset of biometric traits (e.g., right index finger followed by right middle finger), the system ensures that a live user is indeed present at the point of data acquisition. Thus, a challenge-response type of authentication can be facilitated by using multimodal biometric systems. Of course, multimodal biometric systems involve additional cost and increase the enrollment and verification times.

The utilization of digital techniques in the creation, editing and distribution of multimedia data offers a number of opportunities to a pirate user, such as high fidelity copying. Furthermore, Internet is providing additional channels for a pirate to quickly and easily distribute the copyrighted digital content without the fear of being tracked. As a result, the protection of multimedia content (image, video, audio, etc.) is now receiving a substantial amount of attention. Multimedia content protection that is based on biometric data of the users is

being investigated [12]. Password-only encryption schemes are vulnerable to illegal key exchange problems. By using biometric data along with hardware identifiers such as keys, it is possible to alleviate fraudulent usage of protected content [13].

In summary, reliable personal recognition is critical to many government and business processes. The conventional knowledge-based and token-based methods do not really provide positive person recognition because they rely on surrogate representations of the person's identity (e.g., exclusive knowledge or possession). It is, thus, obvious that any system assuring reliable person recognition must necessarily involve a biometric component. This is not, however, to state that biometrics alone can deliver error-free person recognition. In fact, a sound system design will often entail incorporation of many biometric and non-biometric components (building blocks) to provide reliable person recognition. As biometric technology matures, there will be an increasing interaction among the market, technology, and the applications. This interaction will be influenced by the added value of the technology, user acceptance, and the credibility of the service provider. It is too early to predict where and how biometric technology would evolve and get embedded in which applications. But it is certain that biometric-based recognition will have a profound influence on the way we conduct our daily business.

## References

1. Jain, A.K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics* **14** (2004) 4–20
2. Wayman, J.L., Jain, A.K., Maltoni, D., Maio, D.: *Biometric Systems, Technology, Design and Performance Evaluation*. Springer (2005)
3. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer (2003)
4. Li, S., Jain, A.K.: *Handbook of Face Recognition*. Springer (2005)
5. Jain, A.K., Bolle, R., Pankanti, S.: *Biometrics: Personal Identification in Networked Security*. Kluwer Academic Publishers (1999)
6. Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2004: Third Fingerprint Verification Competition. In: *Proceedings of International Conference on Biometric Authentication, LNCS 3072*, Hong Kong (2004) 1–7
7. Philips, P.J., Grother, P., Micheals, R.J., Blackburn, D.M., Tabassi, E., Bone, J.M.: FRVT2002: Overview and Summary. (Available at <http://www.frvt.org/FRVT2002/documents.htm>)
8. Reynolds, D.A., Campbell, W., Gleason, T., Quillen, C., Sturim, D., Torres-Carrasquillo, P., Adami, A.: The 2004 MIT Lincoln Laboratory Speaker Recognition System. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA (2005)
9. Pankanti, S., Prabhakar, S., Jain, A.K.: On the Individuality of Fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 1010–1025
10. Ross, A., Jain, A.K.: Information Fusion in Biometrics. *Pattern Recognition Letters, Special Issue on Multimodal Biometrics* **24** (2003) 2115–2125

11. Hong, L., Jain, A.K.: Integrating Faces and Fingerprints for Personal Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 1295–1307
12. Uludag, U., Jain, A.K.: Multimedia Content Protection via Biometrics-based Encryption. In: Proceedings of IEEE International Conference on Multimedia and Expo, vol. III, Baltimore, USA (July 2003) 237–240
13. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric Cryptosystems: Issues and Challenges. *Proceedings of IEEE, Special Issue on Multimedia Security for Digital Rights Management* **92** (2004) 948–960

# Hierarchical Cell Structures for Segmentation of Voxel Images\*

Lutz Priese, Patrick Sturm, and Haojun Wang

Institute for Computational Visualistics,  
University Koblenz-Landau, Koblenz, Germany

**Abstract.** We compare three hierarchical structures,  $S_{15}$ ,  $C_{15}$ ,  $C_{19}$ , that are used to steer a segmentation process in 3d voxel images. There is an important topological difference between  $C_{19}$  and both others that we will study. A quantitative evaluation of the quality of the three segmentation techniques based on several hundred experiments is presented.

## 1 Introduction

Segmentation of 3d voxel images is a rather new and important discipline as the request for 3d images is increasing in industry and medicine. Most standard segmentation or edge detecting techniques of 2d have been adapted to 3d, such as morphologic operations (e.g., [1]), watersheds (e.g., [2]), level sets (e.g., [3]), B-spline snakes (e.g., [4]), anisotropic diffusion filters (e.g., [2]), Sobel filters and gradient maps (e.g., [5]), etc.

However, generalizing the fast and robust 2d Color Structure Code (CSC) [6] leads to an interesting geometric challenge. The CSC is a hierarchical, inherently parallel, elaborated region growing technique that works with partially overlapping sub-segments: at level  $n+1$  all neighbored and similar partial segments of level  $n$  inside a so-called island of level  $n+1$  are merged into a new segment of level  $n+1$ , whereas the non similar but overlapping segments are splitted. This is done in parallel for all islands of the image. The island structure steers this homogeneous growing or splitting of segments all over the image. The quality of the segmentation depends heavily on certain topological properties of this underlying island structure.

In an ongoing research project of several research groups this 2d CSC is generalized to 3d voxel images. Thus, the 2d hierarchical overlapping island structure has to be generalized to a 3d hierarchical, overlapping 'cell' structure. Unfortunately, in 3d there is no cell structure with all the nice properties of the 2d island structure, complicating the generalization of the CSC.

We very briefly introduce the 2d island structure,  $\gamma_7$ , and the CSC based on  $\gamma_7$ . Then three generalizations of  $\gamma_7$  to 3d cells,  $\gamma_{15}$ ,  $\gamma_{15}$  and  $\gamma_{19}$ , are presented

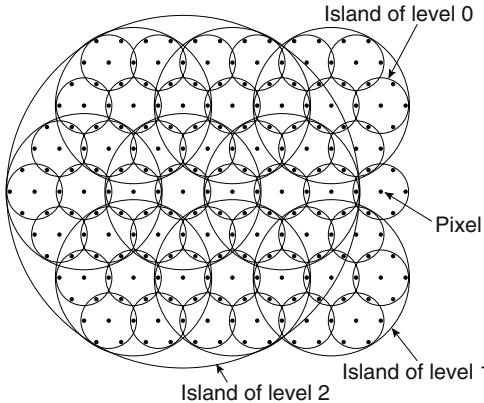
---

\* This work was founded by the BMBF under grant 01/IRC01B (research project 3D-RETISEG).

and some topological properties are mentioned. We discuss the differences between  $_{15}$ ,  $_{15}$  and  $_{19}$ . A measurement of the coverability and error rates in more than thousand segmented 3d images compared with the 'perfect segmentation' gives a quantitative analysis of a CSC segmentation based on  $_{15}$ ,  $_{15}$ , or  $_{19}$  cells.

## 1.1 The Island Structure

The 2d island structure,  $I_7$ , is from [7]. An island of level 0 in a hexagonal pixel structure consists of a center pixel and its six neighbor pixels. An island of level  $n+1$  consists of a center island plus its six neighbor islands, each of level  $n$ . Two islands of level  $n$  are called neighbored if they overlap (in an island of level  $n-1$ ), see Figure 1. Some properties of this structure are:



**Fig. 1.** The hierarchical hexagonal island structure  $I_7$

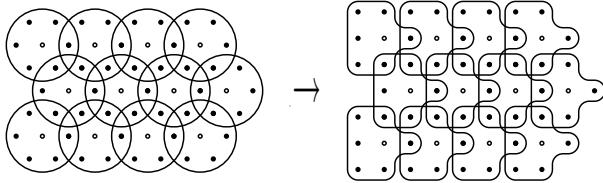
- 6-Neighborhood: each island overlaps with six islands of the same level.
- Plainness: two islands of the same level overlap in at most one island of one level lower.
- Coverability: each island (except the topmost island) is a sub-island of a (parent) island of one level higher.
- Density: two neighbored islands are sub-islands of one common parent island of one level higher.

For a deeper discussion see [8]. This island structure steers a homogeneous region growing (the CSC) as follows:

Do for any island  $I$  of level 0: form a segmentation within the seven pixels of  $I$ .

Do for any island  $I$  of level  $n+1$ :

- Merge neighbored and similar segments of level  $n$  in  $I$  into a new segment of level  $n+1$ , and



**Fig. 2.** Transformation of the hexagonal structure  $I_7$  into the orthogonal grid

- If two neighbored segments  $i_1, i_2$  of level  $n$  within  $I$  are not similar enough then split their common sub-region  $i_{1,2}$  optimally into two sub-regions  $i_{1,2}^1, i_{1,2}^2$  and merge  $i_{1,2}^i$  with  $i_i$ .

Note, a merging of partial segments of level  $n$  is solely done within an island of level  $n+1$ . Thus, if such an island structure would violate 'coverability' all already created partial segments within an island that possesses no parent island would become completely lost for a further region growing. If 'density' is violated already created partial segments of two neighbored islands without a common parent island cannot become merged. Thus, a violation of 'coverability' or 'density' results principally in too many too small segments, an over-segmentation.

Two partial segments are only merged if they already possess a common sub-segment of one level lower. This hierarchical overlapping is the key for the quality of a CSC segmentation.

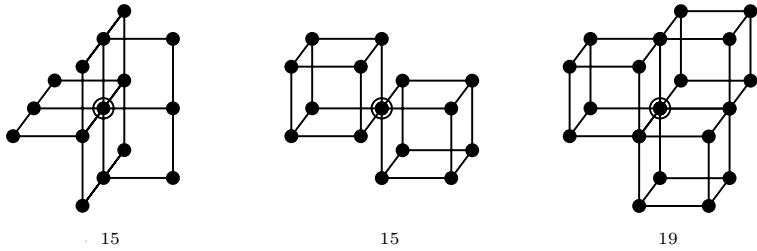
As pixels in real images are given in an orthogonal topology the hexagonal topology of the island structure has to be transformed. I.e., an island of level 0 now becomes simply a window as shown in Figure 2.

## 2 3D Cell Structures

We follow the idea of the most dense sphere packing. The 3d space is organized into 2d layers and the spheres must be placed onto a layer. Each sphere touches 6 spheres of the same layer (in the known hexagonal topology) and three spheres in each layer 'above' and 'below'. Thus,  $\dots_{13}$  consists of a center sphere and 12 neighbored spheres.

Any cell (large enough) can create a hierarchical overlapping cell structure by declaring any second cell of level  $n$  of any second macro row in any second macro layer as a center of a cell of level  $n+1$  (of the same shape as the level 0 cell), compare Figure 1. In this hierarchical topology two cells of level  $n+1$  are neighbored if their center cells of level  $n$  have a (macro) distance of 2 (measured in cells of level  $n$ ). Thus, two neighbored cells of level  $n+1$  possess common sub-cells of level  $n$ ; an overlapping is achieved. Unfortunately, such a  $\dots_{13}$  cell structure even violates coverability: on each level  $1/8$  of all cells don't possess a parent cell.

In a corrected version a  $\dots_{15}$  cell consists of a center cell plus its six neighbors in the same layer plus four cells in each layer above and below.  $\dots_{15}$  can be transformed in several ways in the orthogonal topology. A very compact



**Fig. 3.** The three cell structures

transformation is  $\_15$ , another one, more similar to a cube with holes, is  $\_15$ , see Figure 3. The Manhattan distance of the center to its remotest corner is 2 in  $\_15$  and 3 in  $\_15$ . Both cell structures, created by  $\_15$  and  $\_15$ , possess the wanted properties of 14-neighborhood (each cell overlaps with 14 cells of the same level), plainness (two cells overlap in 0 or 1 sub-cells of one level below), and coverability, but they violate density. Thus, a 3d CSC steered by a  $\_15$  or  $\_15$  cell structure must lead to over-segmentations.

This over-segmentation is avoided by a  $\_19$  cell as presented also in Figure 3. A  $\_19$  cell structure fulfills coverability and density. However, it violates plainness and the neighborhood property:

- Two cells may overlap in 0, 1 or 2 sub-cells of the level below,
- The center cell doesn't overlap with all other 18 sub-cells of a parent cell.

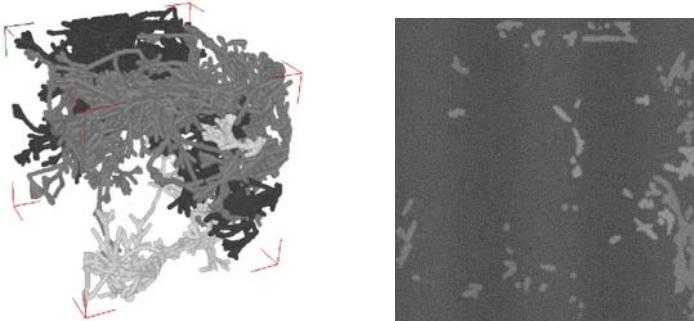
Further, a cell may be a sub-cell of 1, 2 or 3 parent cells. Those three deficiencies heavily complicate an implementation, but they don't lead to an over-segmentation.

### 3 Evaluation

All CSC versions (based on  $\_15$   $\_15$   $\_19$  cells) have been implemented. A  $\_15$ -CSC version will exist soon in hardware. We have applied all versions to real MR and CT (T1 and T2) medical images. A first inspection hinted to slightly better segmentation results for the  $\_15$ -CSC than for the  $\_15$ -CSC (possibly due to the more compact form of the  $\_15$ , although both are orthogonal realizations of the same cell  $\_15$ ). But both are beaten by the  $\_19$ -CSC. A quantitative measurement with real MR and CT data is almost impossible as the theoretical correct segmentations are unknown and a hand segmentation by humans may be worse than by any CSC version. Therefore, we had to create artificial test images where the correct segmentation is known in advance.

#### 3.1 Test Images

We created 16 test images of  $256^3$  voxels of 8 bit gray depth, the 'original images'. 8 images possess a background of gray value 88 and two 3d trees, plexus  $\_1$   $\_2$ , of gray value 122 and 133. Both plexuses  $\_1$  and  $\_2$  don't touch each other but



**Fig. 4.** Three plexuses plus a slice through the blurred image

have a minimal distance of 1 (4 images) or 2 (4 images) voxels. The diameter of each plexus is five voxels and their density varies. Further 8 images are created in the same way but with three plexuses in each of a gray value 122, 133, and 143. Thus, the correct classification, which voxel belongs to plexus or to the background, is known.<sup>1</sup> Now, all images are blurred:

- A sinus curve along the z-axis adds values between -5.3 and +5.3,
  - A 3d binomial filter (a threefold convolution with the vector 1 4(1 2 1) along the x-, y- and z-axis) is applied, and
  - A Gaussian noise with  $\sigma = 2$  (with  $\sigma = 4$  and  $\sigma = 6$ , respectively) is added.

This gives us 48 'blurred images'. Figure 4 present an original image with three plexuses (where each plexus is visualized in an artificial clearly distinguished gray value, and the gray value of the background is set to transparent). A slice through its blurred form with  $\sigma = 6$  in the original gray values is also shown in this Figure.

### 3.2 Statistics

In all CSC versions we declare two neighbored partial segments to be similar enough to become merged if the difference of their mean gray values is below a certain threshold . All 48 test images (with 120 different plexuses) are segmented by all three CSC versions with different thresholds ranging from 6 to 17. This gives us more than 1.000 segmented images with some ten thousand segments. The perfect segmentation for each plexus is known, namely the plexus itself within the corresponding original image. Thus, for any plexus and any set of voxels we can compute in a blurred image the following:

The  $p(\cdot)$  of by , i.e. the number of voxels of that belong to , divided through the number of voxels inside .

The . . . . of (with respect to ), i.e. the number of voxels of that do not belong to , divided by the number of voxels inside .

<sup>1</sup> Those images are free available under 'Download-Section: 3d plexus images' on <http://www.uni-koblenz.de/~lb>

We want to measure the quality of a segmented image. There are many possible ways to do this. The following two methods follow the idea that one wants to conclude from the segmented blurred image to the position of a plexus in the (usually unknown) original image. Thus, the rates  $p(\cdot)$  and  $p(\cdot)$  will be unknown in practice. However, it is reasonable to assume that one can detect the correspondence between a segment  $\cdot$  and a plexus  $\cdot$  if  $p(\cdot)$  is high and  $p(\cdot)$  is low.

**Quality Measure I:** We call a segment  $\cdot$  (with respect to a plexus  $\cdot$ ) if  $p(\cdot) \leq 20\%$ , i.e. more than 80% of the segment  $\cdot$  must belong to  $\cdot$ . For each we regard at most the 25 segments  $i$  with the highest rates  $p(i)$ .

For any segmented blurred image  $'$ , for any plexus  $\cdot$  in the original image of  $'$  do:

- if there exists a fair segment  $s$  with  $p(s) \geq 90\%$  then set output :=  $\cdot$ , otherwise:
- if there exists a union  $\cdot$  of fair segments with  $p(\cdot) \geq 90\%$  then choose such a union  $\cdot$  with minimal error rate and set output :=  $\cdot$ , otherwise:
- if there exists a fair segment  $s$  with  $p(s) \geq 80\%$  then set output :=  $\cdot$ , otherwise:
- if there exists a union  $\cdot$  of fair segments with  $p(\cdot) \geq 80\%$  then choose such a union  $\cdot$  with minimal error rate and set output :=  $\cdot$ , otherwise:
- if there exists a fair segment  $s$  with  $p(s) \geq 70\%$  then set output :=  $\cdot$ , otherwise:
- if there exists a union  $\cdot$  of fair segments with  $p(\cdot) \geq 70\%$  then choose such a union  $\cdot$  with minimal error rate and set output :=  $\cdot$ , otherwise:
- choose a segment  $\cdot$  with maximal  $p(\cdot) - p(\cdot)$ , set output :=  $\cdot$ .

For output  $\cdot$  we measure  $\cdot := p(\cdot)$ ,  $\cdot := p(\cdot)$ ,  $\# := \#_p(\cdot)$ , where  $\#(\cdot)$  is the number of segments used to produce  $\cdot$ .

We group our images into 6 classes  $(\cdot)$ , for  $\in \{1\}$ ,  $\in \{2\}$ ,  $\in \{4\}$ ,  $\in \{6\}$ . Each class  $(\cdot)$  consists of all images with a minimal distance of  $\cdot$  between two plexuses where a Gaussian noise with  $\cdot = \cdot$  was applied. Each class possesses 8 images with 20 plexuses. We present the mean values  $\cdot$ ,  $\cdot$ , and  $\#$  in each class for some thresholds with reasonable results. Using a threshold  $\cdot = 8$  in the  $= 1$  classes, e.g., would result in error rates above 10%.

The line  $\frac{d}{2} \frac{\sigma}{2} \frac{\text{cell}}{S_{15}} \frac{t}{13} \frac{\text{CR}}{92.6} \frac{\text{ER}}{3.2} \frac{\#}{11.6}$  of Table 1 reads as: in average 11.6 segments are required to cover in average 92.6% of a plexus with a mean error rate of 3.2% in the (2,2)-class if we have made a  $S_{15}$ -CSC segmentation with a threshold 13.

The idea of this measure is to find out how few segments are needed to cover a reasonable rate of a plexus. However, the last line of the algorithm, to use a segment with maximal difference of the coverability and error rate,  $p(\cdot) - p(\cdot)$  is maximal, may lead to a segment of the background that touches relatively few voxels of  $\cdot$ . In our next quality measure we will correct

**Table 1.** Statistics for Quality Measure I

d	$\sigma$	cell	t	CR	ER	#	d	$\sigma$	cell	t	CR	ER	#
2	2	$S_{15}$	13	92.6	3.2	11.6	1	2	$S_{15}$	8	77.7	0.6	19.8
2	2	$C_{15}$	13	92.7	3.5	9.8	1	2	$C_{15}$	8	77.8	0.6	19.9
2	2	$C_{19}$	13	98.6	3.1	1.0	1	2	$C_{19}$	6	76.9	0.6	14.2
2	4	$S_{15}$	13	90.8	4.9	13.1	1	4	$S_{15}$	8	73.4	9.2	19.5
2	4	$C_{15}$	13	91.1	5.2	11.8	1	4	$C_{15}$	8	73.6	9.7	21.0
2	4	$C_{19}$	13	96.9	5.0	1.0	1	4	$C_{19}$	8	79.1	16.8	11.7
2	6	$S_{15}$	10	73.9	4.2	17.1	1	6	$S_{15}$	8	64.6	1.6	21.7
2	6	$C_{15}$	10	73.6	4.4	17.1	1	6	$C_{15}$	8	64.1	1.7	22.6
2	6	$C_{19}$	10	83.1	6.6	13.8	1	6	$C_{19}$	6	55.9	2.4	25.0

this final step and will simplify the first steps: we just try to cover as much of as possible, however only with 'reasonable' segments.

**Quality Measure II:** We call a segment  $\dots \dots \dots$  (with respect to ) if is fair and  $p(\ ) \geq 0.1\%$ , i.e. it is large enough.

For any segmented blurred image  $'$ , for any plexus in the original image of ' do:

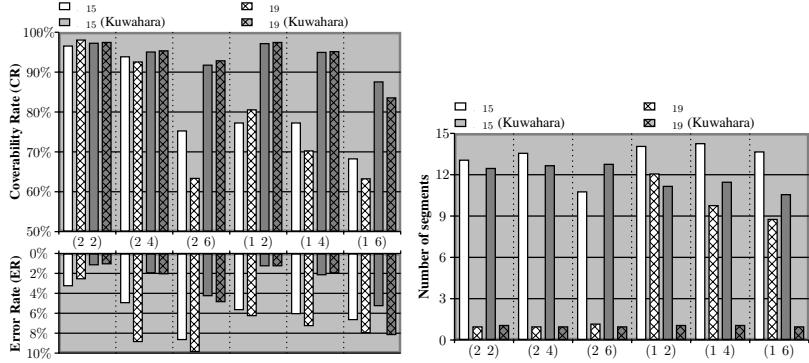
- build the union of all reasonable segments;
- if  $p(\ ) \geq 60\%$  then set output := , otherwise:
- if there exists a segment with  $p(\ ) > 50\%$  choose such a segment with a maximal coverability rate and set output := , otherwise:
- set output :=  $\emptyset$ .

For output  $\neq \emptyset$  we measure , and #, as before. For output =  $\emptyset$  we set := 0, := undefined, and # := undefined.

The line  $\frac{d \ \sigma \ \text{cell} \ t \ \text{CR} \ \text{ER} \ \#}{2 \ 2 \ S_{15} \ 13 \ 96.7 \ 3.3 \ 13.1}$  of Table 2 reads as: a  $_{15}$ -CSC segmentation with a threshold 13 can cover with all reasonable segments in average 96.7% of each plexus in class (2 2) with a mean error rate of 3.3%, where in average 13.1 segments are required. Table 2 is graphically visualized in Graph 1 and 2.

**Table 2.** Statistics for Quality Measure II

d	$\sigma$	cell	t	CR	ER	#	d	$\sigma$	cell	t	CR	ER	#
2	2	$S_{15}$	13	96.7	3.3	13.1	1	2	$S_{15}$	10	77.4	5.7	14.1
2	2	$C_{15}$	12	95.4	2.8	13.5	1	2	$C_{15}$	10	79.8	5.8	14.7
2	2	$C_{19}$	12	98.2	2.6	1.0	1	2	$C_{19}$	8	80.7	6.3	12.1
2	4	$S_{15}$	13	94.0	5.0	13.6	1	4	$S_{15}$	10	77.4	6.1	14.3
2	4	$C_{15}$	14	95.4	6.2	13.8	1	4	$C_{15}$	10	76.2	6.6	15.5
2	4	$C_{19}$	14	92.7	8.9	1.0	1	4	$C_{19}$	8	70.4	7.3	9.8
2	6	$S_{15}$	13	75.4	8.7	10.8	1	6	$S_{15}$	10	68.4	6.7	13.7
2	6	$C_{15}$	13	79.8	10.6	13.1	1	6	$C_{15}$	10	71.1	7.4	16.3
2	6	$C_{19}$	12	63.5	9.9	1.2	1	6	$C_{19}$	8	63.4	8.0	8.8



**Graph 1.** Number of segments in **Graph 2.** Coverability and error each class  $(d, \sigma)$

**Table 3.** Statistics for Quality Measure I for filtered Images

d	$\sigma$	cell	t	CR	ER	#	d	$\sigma$	cell	t	CR	ER	#
2	2	$S_{15}$	8	93.5	1.0	15	1	2	$S_{15}$	8	95.3	1.2	16.8
2	2	$C_{15}$	8	93.7	1.0	13.5	1	2	$C_{15}$	8	95.0	1.2	15.6
2	2	$C_{19}$	8	97.6	1.0	1.0	1	2	$C_{19}$	8	97.5	1.3	1.0
2	4	$S_{15}$	13	92.8	2.2	16.6	1	4	$S_{15}$	8	92.8	1.9	15.3
2	4	$C_{15}$	13	92.8	2.2	15.1	1	4	$C_{15}$	8	93.3	1.9	15.0
2	4	$C_{19}$	13	95.6	2.2	1.0	1	4	$C_{19}$	8	95.2	2.0	1.0
2	6	$S_{15}$	14	90.9	4.6	16.2	1	6	$S_{15}$	8	89.6	2.7	16.1
2	6	$C_{15}$	14	91.0	4.6	15.4	1	6	$C_{15}$	8	89.6	2.7	15.1
2	6	$C_{19}$	14	92.7	4.6	2.7	1	6	$C_{19}$	8	91.4	2.8	2.3

**Table 4.** Statistics for Quality Measure II for filtered Images

d	$\sigma$	cell	t	CR	ER	#	d	$\sigma$	cell	t	CR	ER	#
2	2	$S_{15}$	10	97.4	1.2	12.5	1	2	$S_{15}$	9	97.3	1.3	11.2
2	2	$C_{15}$	10	97.4	1.2	12.2	1	2	$C_{15}$	9	97.3	1.3	12.2
2	2	$C_{19}$	8	97.6	1.1	1.1	1	2	$C_{19}$	8	97.6	1.3	1.1
2	4	$S_{15}$	11	95.2	2.0	12.7	1	4	$S_{15}$	10	95.1	2.2	11.5
2	4	$C_{15}$	11	95.2	2.0	12.4	1	4	$C_{15}$	11	95.3	2.3	12.8
2	4	$C_{19}$	11	95.5	2.1	1.0	1	4	$C_{19}$	8	95.3	2.0	1.1
2	6	$S_{15}$	12	91.9	4.3	12.8	1	6	$S_{15}$	10	87.7	5.3	10.6
2	6	$C_{15}$	14	92.4	4.6	12.7	1	6	$C_{15}$	10	92.2	3.3	12.4
2	6	$C_{19}$	16	93.0	4.9	1.0	1	6	$C_{19}$	9	83.7	8.2	1.0

### 3.3 Preprocessing

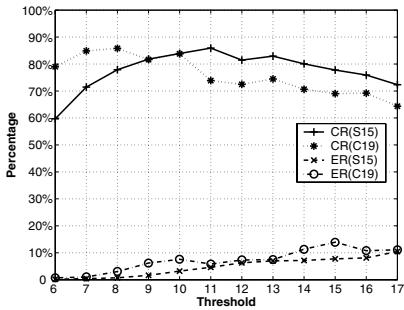
In practice it is important to apply before any segmentation a filter to smoothen the image within homogeneous regions and to enhance the contrast at bor-

ders. Very good such filters are the non-linear Harwood [9] or Kuwahara [10] filters. It is completely obvious how to generalize those known 2d filters to 3d. Tables 3 and 4 present some results for filtered images, where a single iteration of a 3d Kuwahara-Filter of size  $3^3$  is used. This considerably improves the results.

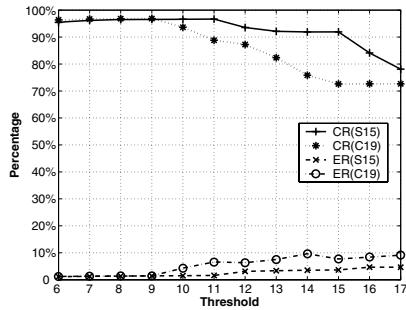
Table 4 is also visualized in Graph 1 and 2. The coverability rates (error rates) for the  $_{15}$  and  $_{19}$  segmentations with and without a Kuwahara-Filter are presented as upper (lower, resp.) bars in Graph 1 and the required number of segments is shown in Graph 2.

### 3.4 Statistics of all Thresholds

In tables 1 to 4 we have presented the coverability and error rates for the different CSC-segmentation variants  $_{15}$ ,  $_{15}$ ,  $_{19}$  for some distinguished thresholds. Here we present the statistics of all used thresholds from 6 to 17 for all images in the classes (2 2), (2 4) and (1 2). Graph 3 presents the coverability and error rates for the  $_{15}$ - and  $_{19}$ -segmentation variant (as  $_{15}$  has very similar results to  $_{15}$  we dropped it) without a filter and Graph 4 does the same with a Kuwahara-Filter.



**Graph 3.** Statistics for Quality Measure II (without filter)



**Graph 4.** Statistics for Quality Measure II for filtered Images (Kuwahara-Filter)

## 4 Discussion

We have compared CSC segmentations based on the rather elegant and compact  $_{15}$  cell structure, on the  $_{15}$  cell structure, and on the much more complicated  $_{19}$  cell structure. There is no significant difference between a  $_{15}$ - or  $_{15}$ -CSC segmentation. A  $_{19}$ -CSC segmentation gives the best results and requires the fewest segments for the unfiltered images with a reasonable small Gaussian noise ( $\sigma = 6$ ). However, for the highly blurred images ( $\sigma = 6$ ) all methods have problems but the  $_{19}$ -CSC segmentation becomes worst. This is probably due

to the larger size of a  $_{19}$  cell. The (  $_6$ )-classes are certainly unfair: the mean difference in the gray values of the plexuses in the unblurred images is just 10, and two plexuses may be rather close. Adding a Gaussian noise with  $\sigma = 6$  means that we change the gray value of 32% of all voxels by a value  $\leq -6$  or  $\geq +6$ , and even 4.5% are changed by a value  $\leq -12$  or  $\geq +12$ . A high deviation in a plexus asks for a large threshold - that, on the other hand, may also melt blurred parts of two different close plexuses.

Applying a single Kuwahara-Filter significantly improves all results. In most cases a single segment of a  $_{19}$ -CSC segmentation suffices to describe a searched structure in the original image. It may be argued that it is unfair to 'unblur' those artificially blurred images. But also in all our tests with real MR or CT images an application of a Kuwahara-Filter improves the segmentation.

A  $_{15}$ - or  $_{15}$ -CSC segmentation however requires always a further aggregation of several segments to describe a searched structure, due to the inherent over-segmentation in both methods. Therefore, a classification problem remains to be solved for the  $_{15}$ - and  $_{15}$ -CSC: which segments have to be aggregated to get the unknown original structure? A solution will probably depend on the kind of images and structures one wants to segment, and requires a further research. The  $_{19}$ -CSC seems to avoid this classification problem by computing only very few reasonable segments. Also, dynamic thresholds have to be added to the CSC. Similarity of segments should depend on properties of local histograms of the images.

## References

- [1] P. Dokladal, R. Urtasun, I. Bloch, and L. Garnero. Segmentation of 3d head mr images using morphological reconstruction under constraints and automatic selection of markers. In *IEEE International Conference on Image Processing 2001*, pages 1075–1078, 2001. ICIP-2001, Thessalonique, Greece, Vol. III.
- [2] J. Sijbers, P. Scheunders, M. Verhoye, A. Van der Linden, D. Van Dyck, and E. Raman. Watershed-based segmentation of 3d mr data for volume quantization. *Magnetic Resonance Imaging*, 15(6):679–688, 1997.
- [3] C. Baillard, P. Hellier, and C. Barillot. Segmentation of brain 3d mr images using level sets and dense registration. *Med. Image Anal.*, 5(3):185–194, 2001.
- [4] Tobias Stammberger, S. Rudert, Markus Michaelis, Maximilian Reiser, and Karl-Hans Englmeier. Segmentation of mr images with b-spline snakes. a multi-resolution approach using the distance transformation for model forces. In *Bildverarbeitung für die Medizin*. Springer, Heidelberg, 1998.
- [5] Christian D. Werner, Frank B. Sachse, Karsten Mühlmann, and Olaf Dössel. Modellbasierte segmentation klinischer mr-aufnahmen. In *Bildverarbeitung für die Medizin*. Springer, Berlin, 1998.
- [6] L. Priese, V. Rehrmann. A fast hybrid color segmentation method. In S.J. Pöpl, H. Handels, editor, *Proc. DAGM Symposium Mustererkennung, Informatik Fachberichte, Springer 1993*, pages 297–304, 1993.
- [7] G. Hartmann. Recognition of hierarchically encoded images by technical and biological systems. *Biological Cybernetics*, 57:73–84, 1987.

- [8] P.Sturm, L.Priese. 3d-color-structure-code. A hierarchical region growing method for segmentation of 3d-images. In J. Bigun, T. Gustavson, editor, *Proc. SCIA 2003, LNCS 2749*, pages 603–608, 2003.
- [9] D. Harwood, M. Subbararaao, H. Hakalahti, L. Davis. A new class of edge preserving smoothing filters. *Pattern Recognition Letters*, 2:155–162, 1987.
- [10] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Processing of r-angiographic images. In K. Preston and M. Onoe, editors, *Digital Processing of Biomedical Images*, pages 187–202, 1976.

# Paving the Way for Image Understanding: A New Kind of Image Decomposition Is Desired

Emanuel Diamant

VIDIA-mant, P.O. Box 933, 55100 Kiriat Ono, Israel  
emanl@012.net.il

**Abstract.** In this paper we present an unconventional image segmentation approach which is devised to meet the requirements of image understanding and pattern recognition tasks. Generally image understanding assumes interplay of two sub-processes: image information content discovery and image information content interpretation. Despite of its widespread use, the notion of “image information content” is still ill defined, intuitive, and ambiguous. Most often, it is used in the Shannon’s sense, which means information content assessment averaged over the whole signal ensemble. Humans, however, rarely resort to such estimates. They are very effective in decomposing images into their meaningful constituents and focusing attention to the perceptually relevant image parts. We posit that following the latest findings in human attention vision studies and the concepts of Kolmogorov’s complexity theory an unorthodox segmentation approach can be proposed that provides effective image decomposition to information preserving image fragments well suited for subsequent image interpretation. We provide some illustrative examples, demonstrating effectiveness of this approach.

## 1 Introduction

Meaningful image segmentation is an issue of paramount importance for image analysis and processing tasks. Natural and effortless for human beings, it is still an unattainable challenge for computer vision designers. Usually, it is approached as an interaction of two inversely directed subtasks. One is an unsupervised, bottom-up evolving process of initial image information discovery and localization. The other is a supervised, top-down propagating process, which conveys the rules and the knowledge that guide the linking and grouping of the preliminary information features into more large aggregations and sets. It is generally believed that at some higher level of the processing hierarchy this interplay culminates with the required scene decomposition (segmentation) into its meaningful constituents (objects), which then can be used for further scene analysis and interpretation (recognition) purposes.

It is also generally believed that this way of processing mimics biological vision peculiarities, especially the characteristics of the Human Visual System (HVS). Treisman’s Feature Integrating Theory [1], Biederman’s Recognition-by-components theory [2], and Marr’s theory of early visual information processing [3] are well known

milestones of biological vision studies that for years influenced and shaped computer vision development. Although biological studies since then have seriously improved and purified their understanding of HVS properties [4], these novelties still have not find their way to modern computer vision developments.

## 2 Inherited Initial Misconceptions

The input front-end of a visual system has always been acknowledged as the most critical system's part. That is true for the biological systems and for the artificial systems as well. However, to cope with input information inundation the systems have developed very different strategies. Biological systems, in course of their natural evolution, have embraced the mechanism of Selective Attention Vision, which allows sequential part-by-part scene information gathering. Constantly moving the gaze from one scene location to another, the brain drives the eye's fovea (the eye's high-resolution sensory part) to capture the necessary information. As a result, a clear and explicit scene representation is built up and is kept in the observer's mind. (Every one, relying on his personal experience, will readily confirm this self-evident truth.)

Human-made visual systems, unfortunately, have never had such ability. Attempts (in robotic vision) to design sensors with log-polar placing of sensory elements (imitating fovea) that are attached to a steerable camera-head (imitating attentional focusing) have permanently failed. The only reasonable solution, which has survived and became the mainstream standard, was to place the photosensitive elements uniformly over the sensor's surface, covering the largest possible field of view of an imaging device. Although initially overlooked, the consequences of this move were dramatic: The bottom-up principle of input information gathering, which prescribes that every pixel in the input image must be visited and processed (normally referencing its nearest neighbors) at the very beginning, imposes an enormous system computational burden. To cope with it, unique image-processing-dedicated Digital Signal Processors (DSPs) were designed and put in duty. The latest advertised prodigy – the Analog Devices TigerSHARC – is able to provide 3,6 GFLOPS of computing power. However, despite of that, to meet real-life requirements, the use of a PCI Mezzanine Card (a BitWare product) featuring four TigerSHARCs on a single board is urgently advised. As well, applications where up to four such cards, performing simultaneously, are envisioned and afforded (delivering approximately 57 GFLOPS per cluster.)

It is worth to be mentioned here that the necessity of the DSPs usage was perfectly clear even when the “standard” image size has not exceeded 262K pixels (512x512 sensor-array). Today, as 3 – 5 Megapixel arrays have became the de facto commercial standard, 16 Megapixel arrays are mastered by professionals, and 30 (up to 80) Megapixel arrays are common in military, space, medical and other quality-demanding applications, what DSP clusters arrangement would meet their bottom-up processing requirements?

The search for an answer always returns to biological vision mysteries. Indeed, the attentional vision studies have never been so widespread and extensive as in the last 5-10 years. The dynamics of eye saccadic movement is quite well understood now. As well, the rules of attention focus guidance. At the same time, various types of perceptual blindness have been unveiled and investigated. The latest research reports convincingly evidence: The hypothesis that our brain image is entire, explicit and clear – (the principal justification for the bottom-up processing) – is simply not true. It is just an illusion [5].

It will be interesting to note that despite of these impressive findings, contemporary computational models of attentional vision (and their computer vision counterparts) keep on to follow the bottom-up information gathering principle [6]. Once upon a time, as well as we are considered, someone had tried to warn about the trap of such an approach [7], but who was ready to hear? Today, a revision of the established canon is inevitable.

### 3 The Revised Segmentation Approach

Considering the results of the latest selective attention vision studies and juxtaposing them with the insights of Kolmogorov Complexity theory, which we adopt to explain the empirical biological findings, we have recently proposed a new paradigm of introductory image processing [8]. For the clarity of our discussion, we will briefly repeat some of its key points.

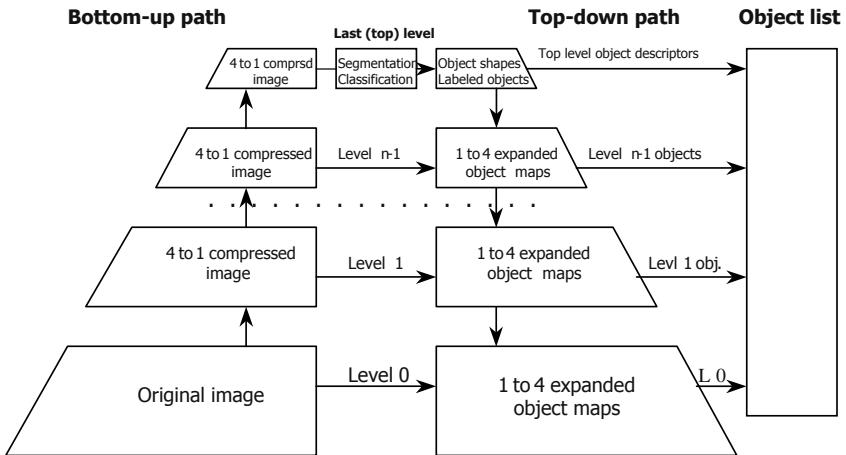
Taking into account the definitions of Kolmogorov's Complexity, we formulate the problem of image information content discovery and extraction as follows:

- Image information content is a set of descriptions of the observable image data structures.
- These descriptions are executable, that is, following them the meaningful part of image content can be faithfully reconstructed.
- These descriptions are hierarchical and recursive, that is, starting with a generalized and simplified description of image structure they proceed in a top-down fashion to more and more fine information details resolved at the lower description levels.
- Although the lower bound of description details is unattainable, that does not pose a problem because information content comprehension is generally fine details devoid.

An image processing strategy that can be drawn from these rules is depicted in Fig.1. As one can see, the proposed schema is comprised of three main processing paths: the bottom-up processing path, the top-down processing path and a stack where the discovered information content (the generated descriptions of it) are actually accumulated.

As it follows from the schema, the input image is initially squeezed to a small size of approximately 100 pixels. The rules of this shrinking operation are very simple and fast: four non-overlapping neighbour pixels in an image at level  $L$  are averaged and the result is assigned to a pixel in a higher ( $L+1$ )-level image. This is known as “four

children to one parent relationship". Then, at the top of the shrinking pyramid, the image is segmented, and each segmented region is labeled. Since the image size at the top is significantly reduced and since in the course of the bottom-up image squeezing a severe data averaging is attained, the image segmentation/classification procedure does not demand special computational resources. Any well-known segmentation methodology will suffice. We use our own proprietary technique that is based on a low-level (local) information content evaluation, but this is not obligatory.



**Fig. 1.** The Schema of the proposed approach

From this point on, the top-down processing path is commenced. At each level, the two previously defined maps (average region intensity map and the associated label map) are expanded to the size of an image at the nearest lower level. Since the regions at different hierarchical levels do not exhibit significant changes in their characteristic intensity, the majority of newly assigned pixels are determined in a sufficiently correct manner. Only pixels at region borders and seeds of newly emerging regions may significantly deviate from the assigned values. Taking the corresponding current-level image as a reference (the left-side unsegmented image), these pixels can be easily detected and subjected to a refinement cycle. In such a manner, the process is subsequently repeated at all descending levels until the segmentation/classification of the original input image is successfully accomplished.

At every processing level, every image object-region (just recovered or an inherited one) is registered in the objects' appearance list, which is the third constituting part of the proposed scheme. The registered object parameters are the available simplified object's attributes, such as size, center-of-mass position, average object intensity and hierarchical and topological relationship within and between the objects ("sub-part of...", "at the left of...", etc.). They are sparse, general, and yet specific enough to capture the object's characteristic features in a variety of descriptive forms.

## 4 Illustrative Example

To illustrate the qualities of the proposed approach we have chosen a scene from the Photo-Gallery of the Natural Resources Conservation Service, USA Department of Agriculture, [9].



**Fig. 2.** Original image, size 1052x750 pixels



**Fig. 3.** Level 5 segmnt., 14 object-regions



**Fig. 4.** Level 4 segmnt., 25 object-regions



**Fig. 5.** Level 3 segmnt., 44 object-regions



**Fig. 6.** Level 2 segmnt., 132 object-regions



**Fig. 7.** Level 1 segmnt., 234 object-regions

Figure 2 represents the original image, Figures 3 – 7 illustrate segmentation results at various levels of the processing hierarchy. Level 5 (Fig. 3) is the topmost nearest level (For this size of the image the algorithm creates a 6-level hierarchy). Level 1 (Fig. 7) is the lower-end closest level. For space saving, we do not provide all exemplars of the segmentation succession, but for readers' convenience all presented examples are expanded to the size of the original image.

Extracted from the object list, numbers of distinguished (segmented) at each corresponding level regions (objects) are given in each figure capture.

Because real object decomposition is not known in advance, only the generalized intensity maps are presented here. But it is clear that even such simplified representations are sufficient to grasp the image concept. It is easy (for the user) now to define what region combination depicts the target object most faithfully.

## 5 Paving the Way for Image Understanding

It is clear that the proposed segmentation scheme does not produce a meaningful human-like segmentation. But it does produce the most reasonable decomposition of visually distinguishable image components, which now can be used as building blocks for an appropriate component grouping and binding procedure. Actually, image understanding arises from the correct arrangement and the right mutual coupling of the elementary information pieces gathered in the initial processing phase. The rules and the knowledge needed to execute this procedure are definitely not a part of an image. They are not an image property. They are always external, and they exist only in the head of the user, in the head of a human observer. Therefore, widespread attempts to learn them from the image stuff (automatically, unsupervised, or by supervised training) is simply a dominant misunderstanding. Nevertheless, numerous learning techniques have been devised and put in duty, including the most sophisticated biology-inspired Neural Networks. However, the trap of low-level information gathering had once again defeated the people's genuine attempts. By definition, neural network tenets assume unsupervised statistical learning, while human learning is predominantly supervised and declarative, that means, essentially natural language based and natural language supported.

What is, then, the right way to introduce to system's disposal the necessary human's knowledge and human's reasoning rules? Such a question immediately involves a subsequent challenge: how this knowledge can be or must be expressed and represented? We think that the answer is only one: as well as human's understanding relies on his world ontology, a task-specific and task-constrained ontology must be provided to system's disposal to facilitate meaningful image processing [10]. It must be human-created and domain-constrained. That means manually created by a human expert and bearing only task-specific and task-relevant knowledge about image parts concepts, their relations and interactions.

It must be specifically mentioned that these vexed questions are not only the fortune of those who are interested in image understanding issues. A huge research and development enterprise is going on now in the domain of the Semantic Web

development [11]. And the unfolding of our own ideas is directly inspired by what is going on in the Semantic Web race.

Our proposed segmentation technique pretty well delineates visually discernable image parts. Essentially, the output hierarchy of segment descriptions by itself can be perceived as a form of a particular ontology, implemented in a specific description language. Therefore, to successfully accomplish the goal of knowledge incorporation, the system designer must also provide the mapping rules between these two ontologies (the mapping also has to be manually created). Because we do not intend to solve the general problem of knowledge transfer to a thinking machine, because we are always aimed on a specific and definite task, it seems that the burden of manual ontology design and its subsequent mapping can be easily carried out. If it is needed, a set of multiple ontologies can be created and cross-mapped, reflecting real life multiplicity of world to a task interaction. At least, such we hope, the things would evolve, when we shall turn to a practical realization of this idea.

## 6 Conclusions

In this paper, we have presented a new technique for unsupervised top-down-directed image segmentation, which is suitable for image understanding and content recognition applications. Contrary to traditional approaches, which rely on a bottom-up (resource exhaustive) processing and on a top-down mediating (which requires early external knowledge incorporation), our approach exploits a top-down-only processing strategy (via a hierarchy of simplified image representations). That means, considerable computational load shrinking can be attained. Especially important is its indifference to any user or task-related assumptions, its unsupervised fashion. The level of segmentation details is determined only by structures discernable in the original image data (the information content of an image, nothing other).

It must be mentioned explicitly: information content description standards like MPEG-4 and MPEG-7, which are fully relying on the concept of a recovered object, left the topic of object segmentation without the scope of the standards (for the reason of irresolvable problem's complexity). As far as we are concerned, that is the first time when a technique is proposed that autonomously yields a reasonable image decomposition (to its constituent objects), accompanied by concise object descriptions that are sufficient for reverse object reconstruction with different levels of details. Moreover, at the final image interpretation stage the system can handle entire objects, and not (as usually) pixels, from which they (obviously) are composed.

## References

1. A. Treisman and G. Gelade, "A feature-integration theory of attention", *Cognitive Psychology*, **12**, pp. 97-136, Jan. 1980.
2. I. Biederman, "Recognition-by-components: A theory of human image understanding", *Psychological Review*, vol. 94, No. 2, pp. 115-147, 1987.

3. D. Marr, "Vision: A Computational Investigation into the Human Representation and Processing of Visual Information", Freeman, San Francisco, 1982.
4. J. M. Henderson, "Human gaze control during real-world scene perception", *Trends in Cognitive Science*, vol. 7, No. 11, pp. 498-504, November 2003.
5. A. Clark, "Is Seeing All It Seems? Action, Reason and the Grand Illusion", *Journal of Consciousness Studies*, vol. 9, No. 5/6, pp. 181-218, May – June 2002.
6. L. Itti, "Models of Bottom-Up Attention and Saliency", In: *Neurobiology of Attention*, (L. Itti, G. Rees, J. Tsotsos, Eds.), pp. 576-582, San Diego, CA: Elsevier, 2005.
7. D. Navon, "Forest Before Trees: The Precedence of Global Features in Visual Perception", *Cognitive Psychology*, 9, 353-383, 1977.
8. E. Diamant, "Searching for image information content, its discovery, extraction, and representation", *Journal of Electronic Imaging*, vol. 14, issue 1, January-March 2005.
9. NRCS image collection. Available: <http://photogallery.nrcs.usda.gov/> (Maryland collection).
10. M. Uschold and M. Gruninger, "ONTOLOGIES: Principles, Methods and Applications", *Knowledge Engineering Review*, vol. 11, No. 2, pp. 93-155, 1996.
11. G. Antoniou, F. van Harmelen, "Semantic Web Primer", MIT Press, 2004.

# Levelset and B-Spline Deformable Model Techniques for Image Segmentation: A Pragmatic Comparative Study

Diane Lingrand and Johan Montagnat

Rainbow Team, I3S Laboratory UMR 6070 UNSA/CNRS,

930, route des Colles – B.P. 145,

F06903 Sophia Antipolis Cedex- France

[Diane.Lingrand@unice.fr](mailto:Diane.Lingrand@unice.fr)

<http://www.i3s.unice.fr/~lingrand>

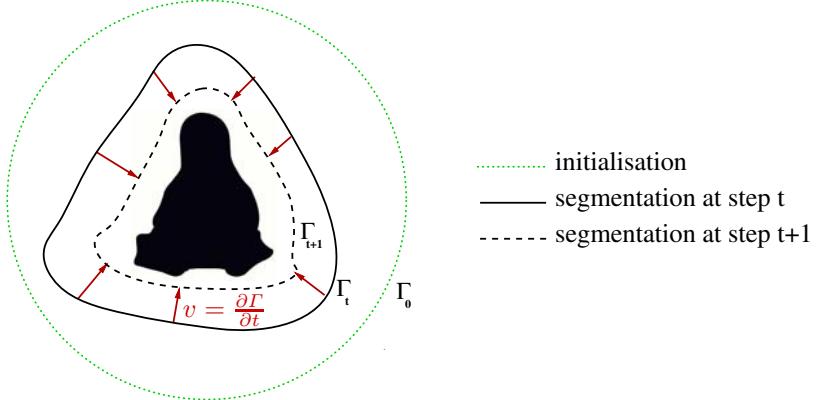
**Abstract.** Deformable contours are now widely used in image segmentation, using different models, criteria and numerical schemes. Some theoretical comparisons between some deformable model methods have already been published [1]. Yet, very few experimental comparative studies on real data have been reported. In this paper, we compare a levelset with a B-spline based deformable model approach in order to understand the mechanisms involved in these widely used methods and to compare both evolution and results on various kinds of image segmentation problems. In general, both methods yield similar results. However, specific differences appear when considering particular problems.

## 1 Motivations

To model objects and segment images, both explicit [7, 18] and implicit [13, 2] deformable models have been proposed in literature [12]. Among these methods, some focus on detecting edges characterized by a high variation of features [10, 3], others on detecting regions characterized by homogeneity of spatially localized properties [17, 11]. Some others focus on both approaches.

Implicit deformable models are very commonly represented by levelsets. Levelsets are widely used for 2D image segmentation [14, 3, 4] and 2D or 3D medical image segmentation [10, 15, 5, 6, 9, 8] among other areas. There are many explicit model representations among which parametric models are the most widely used. In this paper, we consider B-spline parametric models [16] as explicit representation.

Several studies comparing different methods at a theoretical level have been published [1] but without concrete confrontation with real data. Our objective is to propose a comparative study of implicit and explicit deformable model based methods using concrete examples which illustrate the differences between the two approaches. These methods are focused on the determination of a close contour of one or several objects. The initialization consists of a circle or another



**Fig. 1.** Principle of image segmentation using deformable curves

closed curve. This curve is iteratively modified according to the law of evolution (see figure 1):

$$\overrightarrow{\gamma} = \mathbf{n}$$

where  $\gamma$  represents the curve,  $\mathbf{n}$  represents its normal vector, and  $v$  is computed from the image features and the intrinsic curve properties. In practice, we need a representation of the curve, an expression of the force  $v$ , and a numerical scheme to solve this equation.

## 2 Representation of a Curve

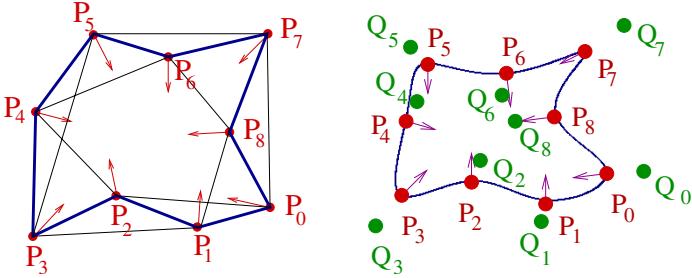
We distinguish three main methods of curve representation: polygonal, parametric and implicit. The polygonal representation is the simplest (see figure 2) but representing a smooth curve implies a model with a large number of points. A parametric model is defined as the set of points  $\gamma(t) = (x(t), y(t))$ , where  $t$  is a real parameter. A lot of papers have been interested by different varieties of splines because of their regularity properties (see figure 2). Finally, among implicit representations, there is a consensus on the levelset representation which handles model topology changes in a simple and elegant manner.

### 2.1 B-Splines

A spline of degree  $n$  is defined as a piecewise polynomial of degree  $n$ . It has an order  $(n - 1)$  continuity. In order to simplify the parameterization of these splines, uniform B-Splines are considered.

A uniform B-Spline is defined by  $(n \geq 3)$  control points  $\gamma_i$  and passes through points  $\gamma_i$  defined by (see figure 2):

$$\gamma_i = (\gamma_{i-1} + 4\gamma_i + \gamma_{i+1})/4 \quad (1)$$



**Fig. 2.** On the left, the simplest representation of a curve: approximation by a polygon. On the right, uniform B-spline defined by the control points  $Q_i$  and passing through the points  $P_i$ . Arrows represent normal vectors

Between two points  $P_i$  and  $P_{i+1}$ , a curve point  $C_i(\lambda)$  is defined by the parameter  $\lambda \in [0, 1]$ :

$$C_i(\lambda) = [P_0 \quad P_1 \quad \dots \quad P_{i-1} \quad P_i \quad P_{i+1} \quad P_{i+2}]^T = \left( -\frac{1}{6} \lambda^{i-1} + \frac{1}{2} \lambda^i - \frac{1}{2} \lambda^{i+1} + \frac{1}{6} \lambda^{i+2} \right)^3 + \left( \frac{1}{2} \lambda^{i-1} - \lambda^i + \frac{1}{2} \lambda^{i+1} \right)^2 + \left( -\frac{1}{2} \lambda^{i-1} + \frac{1}{2} \lambda^{i+1} \right)^3 + \frac{1}{6} \lambda^{i-1} + \frac{2}{3} \lambda^i + \frac{1}{6} \lambda^{i+1}$$

We can rewrite this as:

$$\begin{aligned} C_i(\lambda) &= P_0 + P_1 \lambda + P_2 \lambda^2 + P_3 \lambda^3 \\ C_{i+1}(\lambda) &= P_0 + P_1 \lambda + P_2 \lambda^2 + P_3 \lambda^3 \end{aligned}$$

with:

$$\begin{aligned} [P_0 \quad P_1]^T &= (-i-1 + 4i + i-1) / 6; [P_2 \quad P_3]^T = (-i+1 - 2i + i-1) / 2 \\ [P_1 \quad P_2]^T &= (i+1 - i-1) / 2; [P_3 \quad P_4]^T = (i+2 - 3i+1 + 3i - i-1) / 6 \end{aligned}$$

It is then easy to compute the normal vectors and curvature using the first derivative of  $C_i(\lambda)$  and  $C_{i+1}(\lambda)$ :

$$\mathbf{n} = \frac{1}{\sqrt{\frac{2}{1} + \frac{2}{1}}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = 2(P_1 - P_0)(\frac{2}{1} + \frac{2}{1})^{\frac{3}{2}}$$

This representation is very light: only a limited number of points  $P_i$  are needed.

During model evolution, the force  $F_i$  is applied to points  $P_i$ . At each evolution step, the corresponding control points  $Q_i$  are recomputed by inverting equation (1), in order to determine the B-spline curve (inside and outside regions) and to estimate parameters such as normals and curvature. The computation of normals and curvatures is easy using the above equation. However, during evolution, several parts of the curve may self intersect. It is then necessary to test if this occurs and to split the curve into two parts recursively as needed. Some other tests should be carried out in order to verify the orientation of the curve after splitting operation. If two different curves converge toward the same region (one must be eliminated). However, when two curves intersect, there are different ways of dealing with this depending on the problem to be solved:

- When a shape prior is introduced, and two objects overlap, one may want to segment both objects and preserve their intersection.
- One may want to obtain distinct but non intersecting regions with a common border.
- One may want to fuse the regions into a single one.

Lastly, the property of uniformity may also be altered by evolution. It is then necessary to rearrange the points along the curve.

## 2.2 Levelsets

A 2D curve is defined by a 2D distance map which is an image where each pixel has a value corresponding to the distance between this pixel and the curve. This distance is signed assuming, for example, that points inside (resp. outside) the curve have negative (resp. positive) values. The curve is implicitly defined by the isolevel of value 0.

The normal and curvature are easy to compute on the curve:

$$= \operatorname{div} \left( \frac{\nabla}{|\nabla|} \right) \text{ and } \mathbf{n} = \frac{\nabla}{|\nabla|}$$

where  $\nabla$  represents the distance map. Normal and curvature can also be computed on a point outside the curve using the isolevel going through this point.

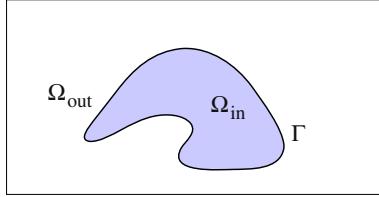
Contrarily to B-splines, the levelset representation implicitly handles problems of topology changes, intersection, superposition or orientation. However, some evolution criterion do not preserve the properties of the distance map: it must be reinitialized regularly.

The levelset representation is simple and easy to implement, but it requires a lot of memory and computations: the whole distance map storage and update is needed for this representation. However, some implementation tricks such as the narrow band may help to reduce the computation time.

## 3 Curve Evolution

A deformable model evolution is steered by an energy minimization process: the solution of the segmentation is defined as the curve minimizing an energy which depends on the image content (data term) and the curve intrinsic regularity properties (internal term). The data term expresses the characterization of what is searched in an image. By segmentation, we mean the extraction of some objects boundaries from a background. Different characterizations of an object exist: using the boundaries or the region inside the object. Boundaries are characterized by areas of large change of intensity or color. Regions are characterized by constant intensity of color over a uniformed background, range of colors, variance, texture, histogram...

In the following, we will consider the segmentation of a region  $_{in}$  from a background  $_{out}$  separated by a curve, (see figure 3).



**Fig. 3.** Notations for the segmentation problem

Assuming that we want to segment a uniform region  $\Omega_{in}$  from a uniform background  $\Omega_{out}$ , the energy is:

$$= \int \int_{\Omega_{in}} (\Omega - \Omega_{in})^2 + \int \int_{\Omega_{out}} (\Omega - \Omega_{out})^2$$

where  $\Omega_{in}$  (resp.  $\Omega_{out}$ ) is the mean intensity of the region  $\Omega_{in}$  (resp.  $\Omega_{out}$ ).

In order to minimize this energy, we derive this expression with respect to the convergence step and obtain the force :

$$= \cdot_1 (\Omega - \Omega_{in})^2 - \cdot_2 (\Omega - \Omega_{out})^2$$

Many authors have observed that several curves may satisfy the evolution criterion because of the presence of noise in the images and the approximation made in order to model the curves. Among the possible solutions, we can decide to take the curve that minimizes the curve length:

$$= \cdot_1 \int \int_{\Omega_{in}} (\Omega - \Omega_{in})^2 + \cdot_2 \int \int_{\Omega_{out}} (\Omega - \Omega_{out})^2 + \cdot_3 \int_{\Gamma}$$

Discretized:

$$= \cdot_1 \sum \sum_{\Omega_{in}} (\Omega - \Omega_{in})^2 + \cdot_2 \sum \sum_{\Omega_{out}} (\Omega - \Omega_{out})^2 + \cdot_3 \int_{\Gamma}$$

The force is then related to the curvature :

$$= \cdot_1 (\Omega - \Omega_{in})^2 - \cdot_2 (\Omega - \Omega_{out})^2 + \cdot_3 \quad (2)$$

## 4 Experiments

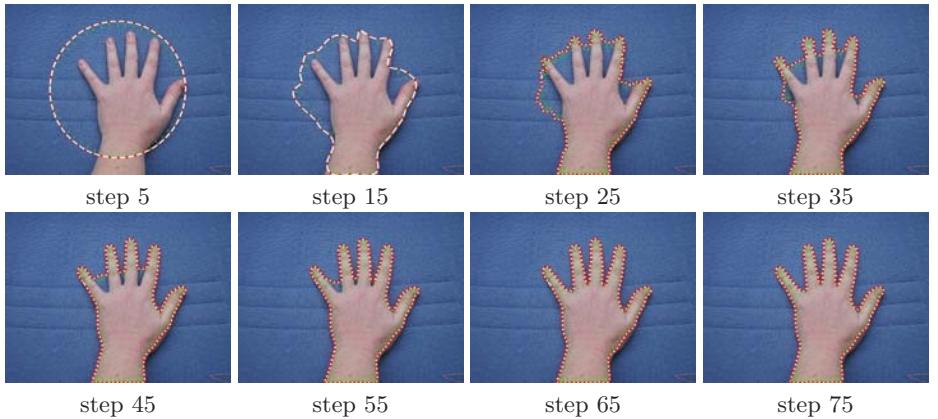
Experiments in this section are done on real images taken with a digital camera. We have restricted the study to objects of uniform value on uniform background. Depending on the images, the value is the gray level, or a color component, or some function of a color component. The numeric scheme used in this paper is the simplest: constant iteration. This is not optimal but it is not the subject studied here.

Experiments have been done using a software written in Java by the author. It is available under the GPL License at author web page.

#### 4.1 Segmenting Objects of High Curvature

In this experiment, we want to segment a hand from a uniform background. The image value used is the ratio of the red component from the sum of blue and green component. We use the evolution criterion (2) with both B-splines and levelset representation, each initialized by a circle centered on the image.

Using the B-splines representation, the curve converges easily to one shape but encounter difficulties to segment the fingers (see figure 4). We have helped the segmentation of the fingers by adding points to the model. It is not easy to automatically determine the number of points needed as it depends on the desired precision and curve smoothness.

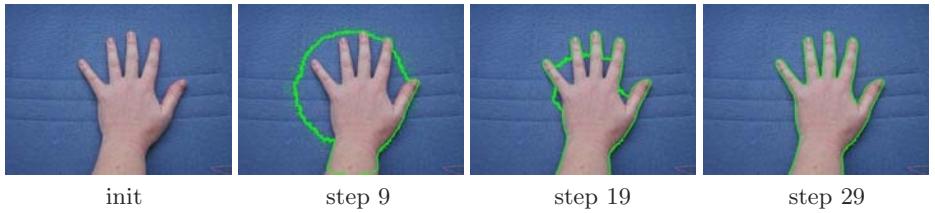


**Fig. 4.** B-spline representation. The convergence is reached after 75 iterations. The parameters used are  $\lambda_1 = \lambda_2 = 0.001$  and  $\lambda_3 = 1$  for iterations 1 to 15 and  $\lambda_1 = \lambda_2 = 0.0005$  and  $\lambda_3 = 1$  for iterations 16 to 60) and finally  $\lambda_1 = \lambda_2 = 0.0001$  and  $\lambda_3 = 1$ . Points have been added in order to be able to represent the curve at iteration 15 (distance between points  $P_i$  limited from 30 to 50 then from 5 to 15 pixels). We observe that the convergence is difficult between the fingers

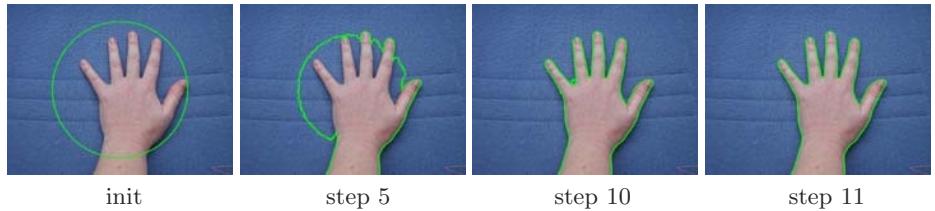
Using the levelset method on the same image, we avoid the difficulties of area of high curvature. However, it is necessary to filter the image before the segmentation with a Gaussian filter (3x3) in order to lower the noise level. Figure 5 shows some artifacts that disappear using the filtering (see figure 6).

#### 4.2 Segmenting Several Regions with Different Colors

In this experiment, we want to segment the different parts of figure 7 using the mean evolution criterion (2). We use the same initialization as previously: a circle centered on the image. Two problems arise: splitting the curve in several parts and having different regions of different intensities.



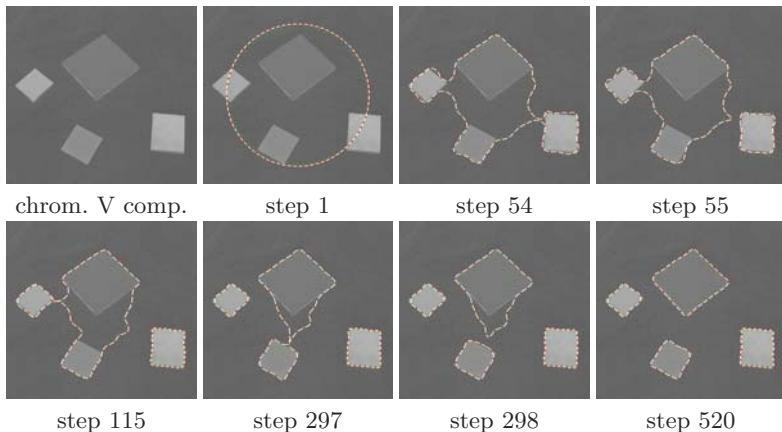
**Fig. 5.** Levelset representation. Convergence using 29 iterations with  $\lambda_1 = 0.001$ ,  $\lambda_2 = 0.001$  and  $\lambda_3 = 1$



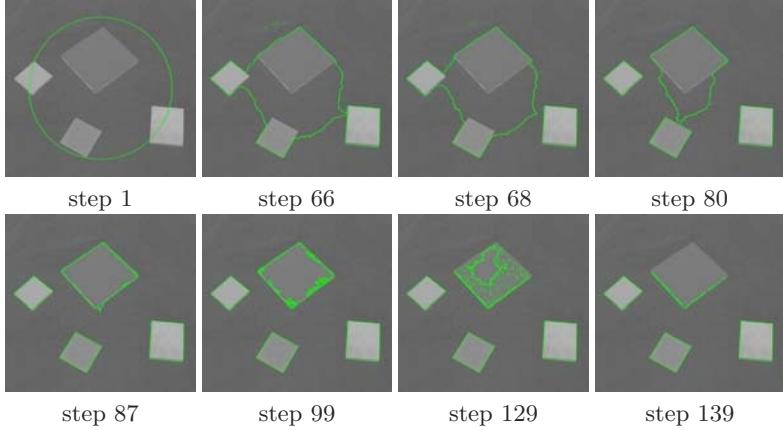
**Fig. 6.** Levelset representation with Gaussian filter (8-connectivity). Convergence using 11 iterations with  $\lambda_1 = 0.001$ ,  $\lambda_2 = 0.001$  and  $\lambda_3 = 1$

As seen before, the levelset representation intrinsically handles the splitting. However, with the B-spline representation, it is necessary to test when the curve intersect itself and to split it into two curves.

Implementing the evolution criterion (2) using the levelset representation,  $_{in}$  is computed from points of negative distance map value while  $_{out}$  is computed



**Fig. 7.** Chrominance component v . Using weights  $\lambda_1 = 0.01$ ,  $\lambda_2 = 0.01$ ,  $\lambda_3 = 1$  and, after iteration number 50:  $\lambda_1 = 0.001$ ,  $\lambda_2 = 0.001$ ,  $\lambda_3 = 1$ . Distance between points  $P_i$  is limited from 30 to 50 pixels



**Fig. 8.** Using Gaussian filter (3x3) and weights  $\lambda_1 = 0.01$ ,  $\lambda_2 = 0.01$ ,  $\lambda_3 = 1$

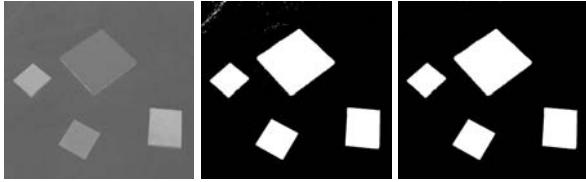
from points of positive values. It is impossible to distinguish points inside one region from points inside another region:  $_{in}$  is common to all regions. Using the B-spline representation, we need to compute a mask of connected points for  $_{in}$  and  $_{out}$  computation: there is no difficulty to compute a new value  $_{in}$  for each region. Figures 7 and 8 show how the curve is splitted. At the beginning, both seem to converge well. But figure 8 shows that the levelset representation cannot deal with the darker region: the estimated mean is higher and this region has a mean intensity closer to the background. The algorithm does not segment this darker region.

The B-spline representation is more adapted to the segmentation of regions with different intensities. However, in figure 7, the squares are approximately segmented because of the small number of points. It would be necessary to add points to better represent corners. Another solution is to divide the image using the segmentation from the B-spline representation and to refine the result, locally, using levelsets.

## 5 Discussion and Conclusion

Assuming that we want to segment objects of uniform intensity (or value) from a uniform background, both levelset and B-spline representation may fit except that (i) B-spline has difficulties to segment objects with high curvature and (ii) levelset are unable to distinguish one region from the others. However, depending on the class of segmentation problem: high precision on one object, several objects to segment, same or different mean intensity, high curvature, necessity of a light representation, automatic or supervised method, we can make use of both representations.

Considering the problem of several different objects with high precision, we propose to first begin with the B-spline representation. The objects are well separated. Assuming that there is no occlusion in the scene, the image is separated



**Fig. 9.** Original image is first filtered using a Gaussian 3x3 filter and then thresholded (threshold value = 110). At last, small regions are deleted

in several parts and levelsets are introduced, using the result of B-spline segmentation for initialization. Is it the simplest and fastest method ? As seen in figure 9, there exists an older method that gives a satisfying result: very simple to implement, very fast. After a Gaussian filtering (3x3), the image is thresholded. Objects are well segmented with some small artifacts that are removed considering only regions of large area. This simplicity hide the problem of the choice of the threshold. Considering an application where this threshold can be calibrated (repeated segmentation in the same room with same luminosity ...), this is the method to be chosen. The PDE based methods do not need a threshold determination but are not fully automatic: the weights are important for the convergence.

As a conclusion, we can say that both methods are useful, separately or one helping another. The PDE based methods are more complex but avoid the problem of threshold determination. B-spline can handle different regions but the number of points must be chosen according to the precision needed. Levelset does not need to determine a number of points but cannot manage different regions if their mean intensity are different. Moreover, PDE methods take their importance for more complex problems using more complex regions properties.

## References

1. G. Aubert and L. Blanc-Féraud. Some remarks on the equivalence between 2D and 3D classical snakes and geodesic active contours. *IJCV*, 34(1):5–17, Sept. 1999.
2. V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. In *Numerische Mathematik*, volume 66, pages 1–33, 1993.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
4. L.D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *Int. J. of Computer Vision*, 24(1):57–78, 1997.
5. M. Droske, B. Meyer, M. Rumpf, and C. Schaller. An adaptive level set method for medical image segmentation. In *Proc. of the Annual Symposium on Information Processing in Medical Imaging*. Springer, LNCS, 2001.
6. J. Gomes and O. Faugeras. Segmentation of the inner and outer surfaces of the cortex in man and monkey: an approach based on Partial Differential Equations. In *Proc. of the 5th Int. Conf. on Functional Mapping of the Human Brain*, 1999.
7. M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active contour models. *International Journal of Computer Vision*, 1:321–332, January 1988.

8. D. Lingrand, A. Charnoz, P.M. Koulibaly, J. Darcourt, and J. Montagnat. Toward accurate segmentation of the LV myocardium and chamber for volumes estimation in gated SPECT sequences. In *8th ECCV*, LNCS 3024, pages 267–278. May 2004.
9. R. Malladi and J.A. Sethian. A Real-Time Algorithm for Medical Shape Recovery. In *(ICCV)*, pages 304–310, Bombay, India, January 1998.
10. R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, 17(2):158–175, February 1995.
11. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):73–91, 1996.
12. J. Montagnat and H. Delingette. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Comput.*, 19(14):1023–1040, Dec. 2001.
13. S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
14. N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. PAMI*, 22(3):266–280, 2000.
15. N. Paragios, M. Rousson, and V. Ramesh. Knowledge-based registration and segmentation of the left ventricle: A level set approach. In *IEEE Workshop on Applications in Computer Vision, Orlando, Florida*, December 2002.
16. F. Precioso and M. Barlaud. B-spline active contour with handling of topological changes for fast video segmentation. *EURASIP*, vol.2002 (6), pages 555–560, 2002.
17. R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.
18. D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and non rigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

# Steerable Semi-automatic Segmentation of Textured Images\*

Branislav Mičušík and Allan Hanbury

Pattern Recognition and Image Processing Group,  
Institute of Computer Aided Automation,  
Vienna University of Technology,  
Favoritenstraße 9/1832, A-1040 Vienna, Austria  
{micusik, hanbury}@prid.tuwien.ac.at

**Abstract.** This paper generalizes the interactive method for region segmentation of grayscale images based on graph cuts by Boykov & Jolly (ICCV 2001) to colour and textured images. The main contribution lies in incorporating new functions handling colour and texture information into the graph representing an image, since the previous method works for grayscale images only. The suggested method is semi-automatic since the user provides additional constraints, i.e. s/he establishes some seeds for foreground and background pixels. The method is steerable by a user since the change in the segmentation due to adding or removing seeds requires little computational effort and hence the evolution of the segmentation can easily be controlled by the user. The foreground and background regions may consist of several isolated parts. The results are presented on some images from the Berkeley database.

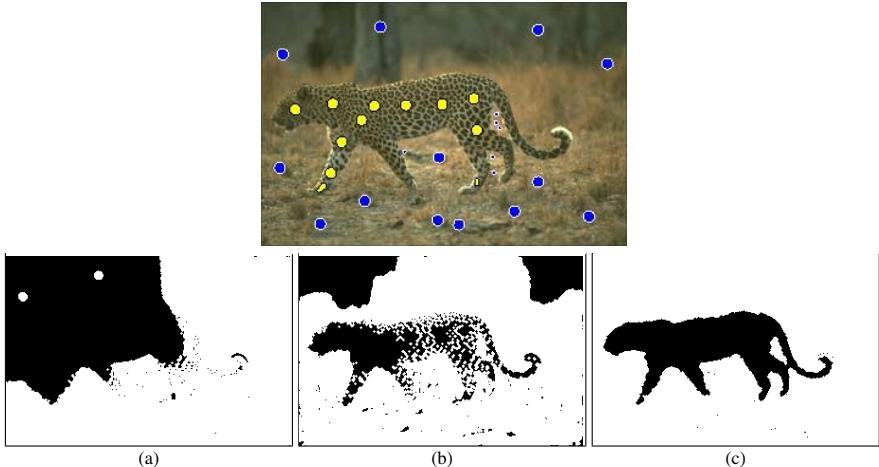
## 1 Introduction

Fully automatic image segmentation is still an open problem in computer vision. An ideal algorithm would take a single image as an input and give the image segmented into semantically meaningful, non-overlapping regions as the output. However, single image segmentation is ill-posed problem and the usual result is either over- or under-segmentation. Moreover, measuring the goodness of segmentations in general is an unsolved problem. Obtaining absolute ground truth is difficult since different people produce different manual segmentations of the same scene [8].

There are many papers dealing with automatic segmentation. We mention only the state-of-the-art work based on normalized cuts [13] for segmenting the image into many non-overlapping regions. This method uses graph cuts as we do, but a modification is introduced, i.e. normalized graph cuts together with an approximate closed-form solution. However, the boundaries of detected regions often do not follow the true boundaries of the objects. The work [14] is a follow-up to [13] where the segmentation is done at various scales.

---

\* This work was supported by the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04), and the European Union Network of Excellence MUSCLE (FP6-507752).



**Fig. 1.** Image segmentation of the leopard image at the top with user-specified foreground (bright/yellow) and background seeds (dark/blue). For the segmentation (a) grayscale [3], (b) colour, and (c) colour+texture information were used respectively

One possibility to partially avoid the ill-posed problem of image segmentation is to use additional constraints. Such constraints can be *i*) motion in the image caused either by camera motion or by motion of objects in the scene [12, 15, 1], or *ii*) specifying the foreground object properties [3, 11, 2].

The motion assumption is used in video matting [12, 15, 1]. The main focus is finding the opacity/transparency of boundary pixels of the foreground object. In our case we perform binary segmentation, i.e. the pixel can belong either to the foreground or the background. No fuzzy memberships are allowed.

In this paper we focus on single image segmentation. We follow the idea given in [3] of interactive segmentation where the user has to specify some pixels belonging to the foreground and to the background. Such labeled pixels give a strong constraint for further segmentation based on min-cut/max-flow algorithm given in [4]. However, the method [3] was designed for grayscale images and thus most of the information is thrown away. In Fig. 1a, one sees the poor result obtained for a textured image segmented using only the grayscale information. By reducing the colour image to a grayscale one, the different colour regions can transform to the same grayscale intensity. Fig. 1b shows how adding colour information helps to achieve a better result. However, many regions contain texture (most natural objects). Taking texture into account helps to improve the final segmentation even more, see Fig. 1c.

The paper [2] uses the segmentation technique [3] as we do. They suggest a method for learning parameters of colour and contrast models left for the user in the method in [3]. They use the Gaussian mixture Markov random field framework. However, contrary to this paper, they do not handle texture information.

In [16] the spatial coherence of the pixels together with standard local measurements (intensity, colour) is handled. They propose an energy function that operates simultaneously in feature space and in image space. Some forms of such an energy function are

studied in [6]. In our work we follow a similar strategy. However, we define the neighborhood relation through brightness, colour and texture gradients introduced in [7, 9].

In [11] the boundary of a textured foreground object is found by minimization (through the evolution of the region contour) of energies inside and outside the region in the context of the Geodetic Active Region framework. However, the texture information for the foreground has to be specified by the user. In [10] the user interaction is omitted by finding representative colours by fitting a mixture of Gaussian elements to the image histogram. However, such technique cannot be used for textured images.

The main contribution of this paper lies in incorporating brightness, colour and texture cues based on the work [7, 9] into the segmentation method [3] based on the maximal flow algorithm. Second, we introduce a new penalty function derived through the Bayesian rule to measure likelihood of a pixel being foreground or background. The proposed method allows one to segment textured images controlled by user interaction.

The structure of the paper is as follows. In Sec. 2, brightness, colour and texture gradients are briefly described. In Sec. 3, a segmentation based on the graph cut algorithm is outlined together with the new energy functions. Finally, the results and summary conclude the work.

## 2 Boundary Detection

Boundary detection is a difficult task, as it should work for a wide range of images, i.e. for images of human-made environments and for natural images. Our main emphasis is put on boundaries at the changes of different textured regions and not local changes inside one texture. This is complicated since there are usually large responses of edge detectors inside the texture. To detect boundaries in images correctly, the colour changes and texturedness of the regions have to be taken into account.

In this work we use as a cue the brightness, colour, and texture gradients introduced in [7, 9]. We shortly outline the basic paradigm.

### 2.1 Brightness and Colour Gradient

First, the RGB space is converted into the CIELAB \*\*\* space. Each of the three channels is treated separately and finally merged together with the texture gradient (will be explained).

Second, at each pixel location ( ) in the image, a circle of radius is created, and divided along the diameter at orientation . The gradient function ( ) compares the contents (histograms) of the two resulting disc halves. A large difference between the disc halves indicates a discontinuity in the image along the disc diameter. In our experiments we used 8 orientations, every  $45^\circ$ , the radius for the channel  $L = 100$ , for the channel  $a = 200$  and for the channel  $b = 200$ . is the length of the diagonal of the image in pixels. We adopt the values used in [9]. The half-disc regions are described by histograms  $i, i$  which are compared using the  $\chi^2$  histogram difference operator

$$\chi^2(\ ) = \frac{1}{2} \sum_{i=1}^{N_b} \frac{(i - i)^2}{i + i} \quad (1)$$



**Fig. 2.** Top: Filter bank for one scale. Bottom: Universal textons sorted by their norms

where  $b$  is the number of bins in the histograms, here for brightness and colour gradient  $b = 32$ . For one pixel there are as many numbers as orientations of (in our case 8). The gradient at each pixel is the maximal number chosen over all orientations. To obtain more robust results suppression of the non-maxima and the use of parabolic interpolation is advisable. The reader is referred to [9] for more details.

After this step a gradient for every channel, i.e.  $L(\cdot)$ ,  $a(\cdot)$ ,  $b(\cdot)$  is obtained.

## 2.2 Texture Gradient

By the texture gradient we mean the gradient computed on the image in the texton domain to capture the variation in intensities in some local neighborhood. The gradient is not related to surface orientation as sometimes used in literature.

To evaluate the texture gradient, we make use of the oriented filter bank, depicted at the top of Fig. 2. The filters are based on rotated copies of a Gaussian derivative and its Hilbert transform. More precisely, even- and odd-symmetric filters, respectively, are written as follows

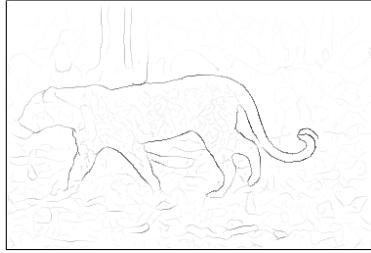
$$\begin{aligned} 1(\cdot) &= \partial_{0,\sigma_1}^2(\cdot) \quad 0,\sigma_2(\cdot) \\ 2(\cdot) &= \dots \quad (1(\cdot)) \end{aligned} \quad (2)$$

where  $\partial_{0,\sigma_1}$  is Gaussian with zero mean and variance  $\sigma_1$ .  $\partial_{0,\sigma_1}^2$  stands for the second derivative. The ratio  $\sigma_2 : \sigma_1$  is a measure of the elongation of the filter. We used the ratio  $\sigma_2 / \sigma_1 = 2$ . The image is convolved with such a bank of linear filters. After that each pixel contains a feature vector with responses to all filters in the filter bank. In our case, we used 24 filters (odd and even symmetric filters with 6 orientations and 2 scales, see the top of Fig. 2). Center-surround filters as in [7] could be added to the filterbank.

The pixels are then clustered, e.g. by  $k$ -means, in filter response feature space. The dominant  $k$  clusters are called *textons*. Alternatively, as we did, the universal textons (obtained from many representative images in Berkeley's database) can be used, see the bottom image of Fig. 2. Then each pixel is assigned to the closest "universal" texton. By this step the image range, usually  $0 - 255$ , is transformed to the range  $1 - k$ , where  $k$  is the number of textons (in our case 64).

The same strategy based on half-discs with 6 orientations and comparing the histograms, as was explained for the brightness and colour gradients in the previous subsection, is applied to the texton image. The number of histogram bins in Eq. (1) is now the number of textons.

After this step a texture gradient  $T(\cdot)$  is obtained.



**Fig. 3.** Combined boundary probability using colour + texture gradient of the leopard image. Black points stand for high, white for low boundary probability

### 2.3 Combined Boundary Probability

The final step for the boundary detection in textured images is to merge the above gradients to obtain a single value for each pixel.

We begin with a vector composed of the brightness, colour, and texture gradients,

$$\mathbf{x}(\cdot) = [1 \quad L(\cdot) \quad a(\cdot) \quad b(\cdot) \quad T(\cdot)]^\top \quad (3)$$

To define the final probability for a pixel at position  $(\cdot)$  to be a boundary, a sigmoid is used [7]

$$b(\cdot) = \frac{1}{1 + e^{-\mathbf{x}^\top \mathbf{b}}} \quad (4)$$

where the constant vector  $\mathbf{b}$  consists of weights for each partial gradient. If there is no boundary change in a pixel at position  $(\cdot)$ , the vector  $\mathbf{x} = (1 \ 0 \ 0 \ 0 \ 0)^\top$  and  $\mathbf{x}^\top \mathbf{b} = 1$ . The “1” is at the beginning of the vector  $\mathbf{x}$  hence allows one to control the weight in the “no boundary” case through the  $1$  in the vector  $\mathbf{b}$ .

The method for obtaining the weights in  $\mathbf{b}$ , i.e. combining the information from all gradients in an optimal way, is suggested in [9]. They used human labeled images from the Berkeley database as ground truth [8]. In our implementation we used the  $\mathbf{b}$  provided with the source code on the web by the authors [9].

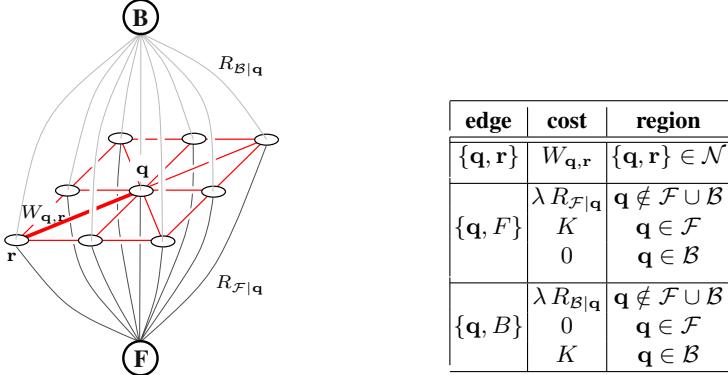
The “0” value for  $b$  in Eq. (4) indicates no boundary, the “1” value indicates a boundary, i.e. a change of colour or texture, in the image with maximal confidence. See Fig. 3 for the combined boundary probability of the image in Fig. 1.

## 3 Segmentation

We used a segmentation technique based on the interactive graph cuts method first introduced in [3]. There exists a very efficient algorithm for finding min-cut/max-flow in a graph [4]. We introduced new penalties on edges in the graph based on a RGB colour cue and on a combined boundary cue, respectively.

### 3.1 Building the Graph

The general framework for building the graph is depicted in Fig. 4. The graph is shown for a 9 pixel image and an 8-point neighborhood  $\mathcal{N}$ . For general images, the graph has



**Fig. 4.** Left: Graph representation for 9 pixel image. Right: Table defining the costs of graph edges.  $K$  and  $\lambda$  are constants described in the text

as many nodes as pixels plus two extra nodes labeled  $\textcircled{r}$ ,  $\textcircled{q}$ , and the neighborhood is larger.

Each node in the graph is connected to the two extra nodes  $\textcircled{r}$ ,  $\textcircled{q}$ . It allows the incorporation of the information provided by the user and sets a penalty for each pixel being foreground or background. The user specifies two disjoint sets  $\mathcal{F}$  and  $\mathcal{B}$  containing samples of foreground and background pixels. If, for instance, the image point  $q$  is marked as belonging to the foreground then there is a maximum weight  $K$  on the edge  $\{q, \textcircled{r}\}$  and zero weight on the edge  $\{q, \textcircled{q}\}$ .  $K$  is some large number larger than  $\min = 1 + \max_q \sum_{r: \{q, r\} \in \mathcal{N}} q, r$ .

The regional penalty of a point  $q$  not marked by the user as being foreground  $\mathcal{F}$  or background  $\mathcal{B}$  is defined as follows

$$\begin{aligned} \mathcal{F}|q &= -\ln (\mathcal{B}|c_q) \\ \mathcal{B}|q &= -\ln (\mathcal{F}|c_q) \end{aligned} \quad (5)$$

where  $c_q = (r_g_b)^T$  stands for a vector in  $\mathbb{R}^3$  of RGB values at the pixel  $q$ . To compute the posterior probabilities in Eq. (5) we used the Bayesian rule as follows

$$\frac{(\mathcal{B}|c_q)}{(\mathcal{F}|c_q)} = \frac{(c_q|\mathcal{B}) / (\mathcal{B})}{(c_q|\mathcal{F}) / (\mathcal{F})} = \frac{(c_q|\mathcal{B}) / (\mathcal{B})}{(c_q|\mathcal{B}) / (\mathcal{B}) + (c_q|\mathcal{F}) / (\mathcal{F})} \quad (6)$$

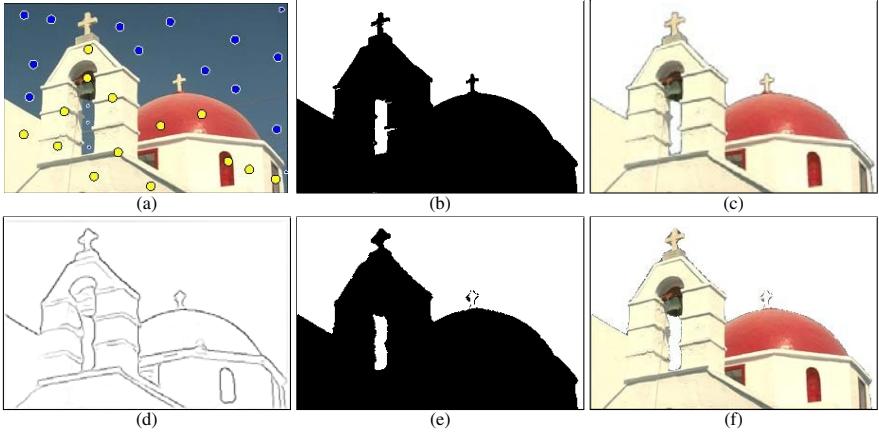
We demonstrate it on  $(\mathcal{B}|c_q)$ , for  $(\mathcal{F}|c_q)$  it is analogical.

We do not know a priori the probabilities  $(\mathcal{F})$  and  $(\mathcal{B})$  of the foreground and background regions, i.e. how large the foreground region is compared to the background one. Thus, we fixed them to  $(\mathcal{F}) = (\mathcal{B}) = 0.5$  and Eq. (6) then reduces to

$$(\mathcal{B}|c_q) = \frac{(c_q|\mathcal{B})}{(c_q|\mathcal{B}) + (c_q|\mathcal{F})} \quad (7)$$

where the prior probabilities are

$$(c_q|\mathcal{F}) = \frac{r}{c_r} \cdot \frac{g}{c_g} \cdot \frac{b}{c_b} \quad \text{and} \quad (c_q|\mathcal{B}) = \frac{r}{c_r} \cdot \frac{g}{c_g} \cdot \frac{b}{c_b}$$



**Fig. 5.** Church image. (a) Input image with specified foreground/background seeds. (b,c) Segmentation using colour cue. (d) Combined boundary probability. (e,f) Segmentation using colour and texture gradient

where  $i^{\{r,g,b\}}$ , resp.  $i^{\{r,g,b\}}$ , represents the foreground, resp. the background histogram of each colour channel separately at the  $i$ th bin learned from seed pixels. Here the histograms are represented in RGB colour space, nevertheless another colour space, e.g. L\*a\*b\*, can be used. We used 64 bins for each colour channel. For better performance we smoothed the histograms using a one-dimensional Gaussian kernel.

In an implementation one should take into account the possibility of a zero value of  $(\mathcal{B}|\mathbf{c}_q)$  in Eq. (5) and thus avoid an overflow. In such a case  $\mathcal{F}|\mathbf{q} = \cdot$ .

The edge weights of neighborhood  $\mathcal{N}$  are encoded in the matrix  $W_{q,r}$ , which is not necessarily symmetric. Setting the values of these weights is discussed in the following subsections. The size and density of the neighborhood are controlled through two parameters. We used a neighborhood window of size  $21 \times 21$  with sample rate 0.3, i.e. only a randomly selected 30% of all pixels in the window are used. Using only a fraction of pixels in the window reduces the computational demand and thus allows the use of larger windows while preserving the spatial relations.

### 3.2 Segmentation Using RGB Colour Cue

The simplest straightforward modification of the weight function in [3] is the augmentation of the penalty function to take colour into account. The weight matrix is chosen as follows

$$W_{q,r} = \frac{-\|\mathbf{c}_q - \mathbf{c}_r\|^2}{\sigma_1} \cdot \frac{1}{\|\mathbf{q} - \mathbf{r}\|} \quad (8)$$

where  $\mathbf{c}_q$  is the RGB vector of a point at the position  $\mathbf{q}$  (as in Eq. (5)).  $\sigma_1$  is a parameter (we used  $\sigma_1 = 0.02$  in all our experiments).

The penalty in Eq. (8) is good only for textureless images, as in Fig. 5. The next section suggests a more general approach using colour and texture.

### 3.3 Segmentation Using Combined Boundary Cue

A more general approach to define graph weights is to incorporate the combined boundary probability from Sec. 2.3. The neighborhood penalty of two pixels is defined as

$$q_{r,r} = \left( -\frac{g(q,r)^2}{\sigma_2^2} \right)^2 \quad (9)$$

where  $\sigma_2$  is a parameter (we used  $\sigma_2 = 0.08$  in all our experiments) and

$$(q, r) = b(q) + \max_{s \in \mathcal{L}_{q,r}} b(s) \quad (10)$$

where  $b(q)$  is the combined boundary probability described in Sec. 2.3 and  $\mathcal{L}_{q,r} = \{x \in \mathbb{R}^2 : x = q + t(r-q), t \in (0, 1)\}$  is a set of points on a line from the point  $q$  (exclusive) to the point  $r$  (inclusive). We used the DDA line algorithm to discretize the line. The penalty in Eq. (10) follows the idea that there is a large weight if the line connecting two points crosses an edge in the combined boundary probability image. The value of the weight corresponds to the strength of the edge. If there is no edge between the points the weight approaches zero.

## 4 Experiments

The segmentation method was implemented in MATLAB. Some of the most time consuming operations (creating the graph edge weights) were implemented in C and interfaced with MATLAB through mex-files. We used with advantage the sparse matrices directly offered by MATLAB. We used the online available C++ implementations of the min-cut algorithm [4] and some MATLAB code for colour and texture gradient computation [7]. The parameters like number of histogram bins (64), neighborhood window size (21x21), sample rate (0.3), and sensitivities ( $\sigma_1 = 0.02$ ,  $\sigma_2 = 0.08$ ) were obtained experimentally for giving the best performance on a large database of images.

The most time consuming part of the segmentation process is creating the weight matrix  $\mathbf{W}$ . However, the matrix  $\mathbf{W}$  is created only once and adding some new seeds (user interaction) changes only the penalties  $\mathcal{F}|_q$ ,  $\mathcal{B}|_q$  which requires a minimum amount of time. Hence the segmentation method can be designed as an interactive method, i.e. the user can interactively add new seed points and thus control the final segmentation. Once the graph is built, finding the min-cut takes 2 – 5 seconds on a  $250 \times 375$  image running on a Pentium 4@2.8 GHz.

The first experiment shows the performance of both suggested weights, Eq. (8) and Eq. (9). The church image in Fig. 5 is a colour image with no significant texture. In this case, using the same seed points, the method based on the RGB colour space gives a comparable result to the method based on the colour + texture gradient. Notice the missing cross in the second method. The user can of course add extra seed points on the cross and the problem would be solved. However, we wanted to show that the penalty based on colour can sometimes give better results than using color + texture gradient and is therefore useful in certain applications, especially if execution time is the main criterion. Segmentation by the first method took 13 seconds, by the second method 97



**Fig. 6.** Results. 1<sup>st</sup> column: original image with foreground (bright/yellow) and background (dark/blue) seeds. 2<sup>nd</sup> column: combined boundary probability. 3<sup>rd</sup> column: binary segmentation. 4<sup>th</sup> column: segmentation with masked original image

seconds. However, the implementation of the texture gradient in C would dramatically speed up the computation time of the second approach.

For textured images, as in Fig. 1, the method taking into account texture information gives the best result. Other results are shown in Fig. 6 on various images from the Berkeley database [5]. On the last “boat” image, it is shown that it depends on the user

to specify what the object of interest in the image will be. It enables one to segment the image into many regions. In our case we first segmented the boat, then the houses at the back.

## 5 Conclusion

We improved the method [3] based on graph cuts by incorporating the brightness, colour, and texture gradient based on [7] into the graph edge weights. We introduced a new penalty function derived through the Bayesian rule to measure the pixel likelihood of being foreground or background. The proposed method is semi-automatic and provides segmentation into foreground and background objects (which may consist of several isolated parts). The method is interactive since adding or removing seeds takes little computational effort and hence the evolution of the segmentation can easily be controlled by the user.

## References

1. N. Apostoloff and A. Fitzgibbon. Bayesian estimation of layers from multiple images. In *Proc. CVPR*, volume 1, pages 407–414, 2004.
2. A. Blake, C. Rother, M. Brown, P. Perez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. ECCV*, volume 1, pages 428–441, 2004.
3. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. ICCV*, pages 105–112, 2001.
4. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.
5. <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>.
6. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.
7. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, 2001.
8. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, pages 416–425, 2001.
9. D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.
10. N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. In *Proc. ECCV*, volume II, pages 224–240, 2000.
11. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *IJCV*, 46(3):223–247, 2002.
12. M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proc. CVPR*, volume 1, pages 18–25, 2000.
13. J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.
14. X. Y. Stella. Segmentation using multiscale cues. In *Proc. CVPR*, pages I:247–254, 2004.
15. Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *Proc. ECCV*, volume 3, pages 487–501, 2002.
16. R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Proc. CVPR*, volume 2, pages 437–444, 2004.

# MSCC: Maximally Stable Corner Clusters\*

Friedrich Fraundorfer, Martin Winter, and Horst Bischof

Institute for Computer Graphics and Vision,

Graz University of Technology,

Inffeldgasse 16/2, A-8010 Graz, Austria

{fraunfri, winter, bischof}@icg.tu-graz.ac.at

<http://www.icg.tu-graz.ac.at>

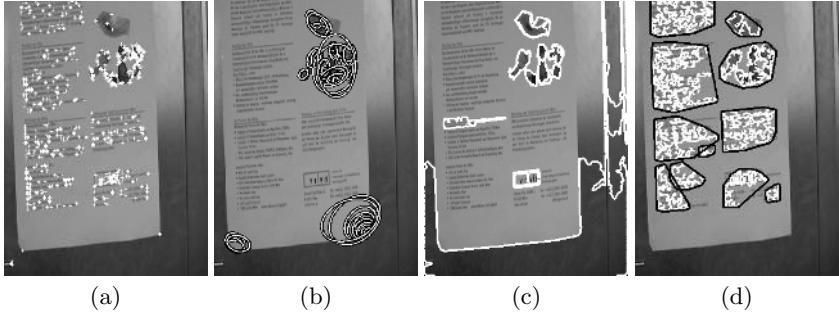
**Abstract.** A novel distinguished region detector, complementary to existing approaches like Harris-corner detectors, Difference of Gaussian detectors (DoG) or Maximally Stable Extremal Regions (MSER) is proposed. The basic idea is to find distinguished regions by clusters of interest points. In order to determine the number of clusters we use the concept of maximal stabilities across scale. Therefore, the detected regions are called: Maximally Stable Corner Clusters (MSCC). In addition to the detector, we propose a novel joint orientation histogram (JOH) descriptor ideally suited for regions detected by the MSCC detector. The descriptor is based on the 2D joint occurrence histograms of orientations. We perform a comparative detector and descriptor analysis based on the recently proposed framework of Mikolajczyk and Schmid, we present evaluation results on additional non-planar scenes and we evaluate the benefits of combining different detectors.

## 1 Introduction

Recently, there has been a considerable interest in using local image region detectors and descriptors for wide base-line stereo [1, 2], video retrieval and indexing [3, 4], object recognition [5], and categorization tasks [6, 7]. There exist two main categories of distinguished region detectors. Corner based detectors like Harris [8], Harris-Laplace [4], Harris-Affine [9] etc. and region based detectors such as Maximally Stable Extremal Regions (MSER) [1], Difference of Gaussian points (DoG) [5] or scale space blobs [10]. In addition Brown et al. proposed to use groups of interest points [11]. Corner based detectors locate points of interest at regions which contain a considerable amount of image structure, but they fail at uniform regions and regions with smooth transitions. Region based detectors deliver blob like structures of uniform regions but highly structured regions are

---

\* This work is funded by the Austrian Joint Research Project Cognitive Vision under sub-projects S9103-N04 and S9104-N04, by a grant from Federal Ministry for Education, Science and Culture of Austria under the CONEX program and partially supported by the European Union Network of Excellence, MUSCLE under contract FP6-507752.



**Fig. 1.** (a) Harris corner detector. (b) Hessian-Affine detector. (c) MSER detector. (d) MSCC detector (one scale only)

not detected. As the two categories act quite complementary it is no surprise that people have started to use detector combinations (e.g. Video Google [3]). The main benefit in combining detectors with complementary properties is the increasing number of detected regions and thus possible matches. A second benefit is that the different detectors will fire in different regions of the image. Thus the image will be more uniformly covered with detected regions, which in turn improves the accuracy of e.g. the wide base-line stereo. In general, it is expected that the robustness of most algorithms will improve by combining different detectors. The accuracy in geometry estimation will be improved if more matches can be used and if the matches are distributed over the whole scene. In object recognition a better coverage of the object increases the robustness against partial occlusions. The available corner and region based detectors cover already a broad variety of image content. However, there are images not sufficiently covered by neither class of detectors. For example, Fig. 1 shows an image from a database we are using for studying visual robot localization tasks. The image shows a door with an attached poster. The poster contains a lot of text. Fig. 1(a) shows as an example detected Harris corners. The text in the image results in a high number of Harris corners with a very similar appearance which will result in a lot of mismatches. Fig. 1(b) shows the results of an Hessian-Affine detector. The detector selects strong corners and constructs local affine frames around the corners on multiple scales. This leads to unfortunate detections as one can see in the lower right part of the image. The MSER detector (see Fig. 1(c)) perfectly detects the homogeneous regions but ignores the parts of the image containing the text. If the resolution of the image would be higher, the detector would detect the individual letters, which in this case would be useless for matching because there are lots of similar letters. This simple example demonstrates that neither of the available detectors delivers satisfactory results.

The dense but locally distinct patterns of detected corners in textured image parts resulting from the Harris detector (Fig. 1(a)) suggests a new distinguished region detector based on characteristic groups of individual detections. In particular we propose to cluster the responses of individual detectors. The thus clustered regions are our distinguished regions. Fig. 1(d) shows some of the ob-

tained regions (only at a single scale). It is clearly seen that we detect also regions where other detectors have problems. The remainder of the paper is organized as follows: In Section 2 we present the MSCC detector. Section 3 is devoted to the novel joint orientation histogram descriptor. The detector and descriptor evaluations are presented in Section 4. A discussion and outlook concludes the paper.

## 2 The MSCC Detector

The MSCC detector aims to benefit from the high repeatability of simple interest point detectors (as shown in [12]). The main idea is to use point constellations instead of single interest points. Point constellations are more robust against viewpoint changes than single interest points because a few missing single points will not affect the detection of the constellation itself. Point constellations are detected by clustering of the interest points for multiple scales. Selecting only those clusters which fulfill a stability criteria leads to robust and highly repeatable detections.

In particular the MSCC algorithm proceeds along the following three steps:

1. Detect simple, single interest points all over the image, e.g. Harris-corners.
2. Cluster the interest points by graph-based point clustering using a minimal spanning tree (MST) for multiple scales.
3. Select clusters which are stable across several scales.

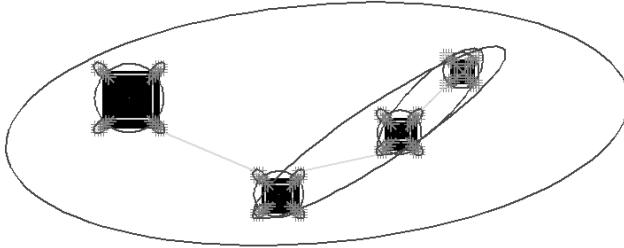
It should be noted, that the steps 2 and 3 of the algorithm can be implemented very efficiently as it is possible to cluster and perform the cluster selection already during the MST construction.

### 2.1 Interest Point Detection and MST Calculation

To detect the interest points acting as cluster primitives we make use of the structure tensor [13]. For every image pixel we evaluate the structure tensor and calculate a Harris-corner strength measure (cornerness) as given by Harris and Stephens [8]. We select a large number of corners (all local maxima above the noise level) as our corner primitives. This ensures that we are not dependent on a cornerness threshold. The interest points itself represent the nodes of an undirected weighted graph in 2D. The MST computation is performed by applying Kruskal's iterative growing algorithm on the Delaunay-triangulation of the nodes [14]. The weight for the edge between two graph nodes is their geometric distance to which we will also refer to as edge length.

### 2.2 Multi Scale Clustering

Since we do not know the number of clusters we have to use a non-parametric clustering method. The method we use is inspired by the MSER detector. Given a threshold  $T$  on the edge length we can get a subdivision of the MST into subtrees



**Fig. 2.** Example of the MSCC detector on a synthetic test image (clustered interest points are indicated by ellipses around them)

by removing all edges with an edge length higher than this threshold. Every subtree corresponds to a cluster and gives rise to an image region. Different values for  $T$  produce different subdivisions of the MST, i.e. different point clusters. To create a multi scale clustering we compute subdivisions of the MST for a certain number of thresholds  $T_1 \dots T_p$  between the minimal and maximal edge length occurring in the MST.

We are now interested in clusters which are stable over several scales, i.e. have the same interest points. Fig. 2 illustrates the method on a synthetic test image. The image shows 4 differently sized squares. The Harris corner detection step produces several responses on the corners of the squares. Connecting the single points with the MST reveals a structure where one can easily see that clustering can be done by removing the larger edges. Clusters of interest points are indicated by ellipses around them. The test image shows the capability of detecting stable clusters at multiple scales, starting from very small clusters at the corners of the squares itself up to the cluster containing all detected interest points.

Unlike many other detectors the MSCC-clusters show arbitrary shapes, an approximative delineation may be obtained by convex hull construction or fitting ellipses. Fig. 4(c) shows examples for the convex hull and the fitted ellipses of detected regions in a 3D plot. One can see, that ellipse fitting is only a poor estimation of region delineation and will also introduce some errors for area and overlap calculation. However we will propose a descriptor without the need for region delineation which uses the clustered points directly.

### 3 The Joint Orientation Histogram Descriptor

Orientation information has already been successfully used by Lowe in his SIFT-descriptor [5]. To make an optimal use of the MSCC detector results we have designed a new descriptor also based on orientation information, the so called joint orientation histograms (JOH).

As an estimate of the local gradient orientation we use Gaussian derivatives. In order to obtain rotational invariance the mean corner orientation within a region is computed. All corner orientations are normalized with respect to this

mean orientation. The basic idea is that for each corner  $c$  within a MSCC region the joint orientation occurrence to its  $n$  local neighbors  $c_t$  weighted by the gradient magnitude is entered in a 2D histogram. That is, the gradient magnitude of  $c$  is added to the bin defined by the orientation of  $c$  and  $c_t$ . The histogram is smoothed in order to avoid boundary effects and normalized to sum to one. Best results are obtained for a typical histogram size of  $8 \times 8$  and  $n = 40$  nearest neighbors resulting in a 64-dimensional descriptor. The histogram takes into account the local structure within the region (e.g. are there many similar orientations or are the orientations uniformly distributed). In contrast to most other descriptors we do not depend on the definition of a center point or a region (ellipse) where the descriptor is calculated from. Therefore, even if the detected clusters differ in shape, this does not affect the descriptor as long as a sufficiently high number of points are re-detected. Matching (see 4.4) is done by nearest neighbor search using the Bhattacharyya distance [15].

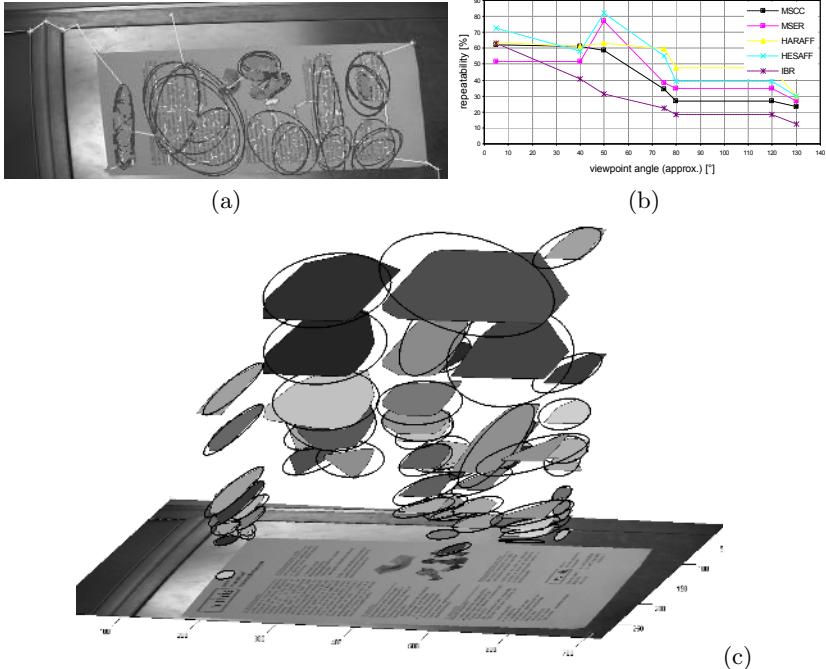
## 4 Experimental Results

### 4.1 Detector Evaluation: Repeatability and Matching Score

To compare the performance of the MSCC detector with other approaches we use the publicly available evaluation framework of Mikolajczyk [16]. The evaluation framework gives two performance measures, a repeatability score and a matching score. The repeatability score is the relative number of repetitive detected interest regions. The matching score is the relative number of correctly matched regions compared to the number of detected regions. The correctness of the matches is verified automatically using a ground truth transformation. A homography is calculated between two (planar) images which allows to transfer the position of an interest region from the first to the second image. For the evaluation we use 10 images from a robot localization experiment. Fig. 3(a) shows an image of the image set and Fig. 4(a) shows an image with detected MSCC regions. To comply with the evaluation framework ellipses are fitted to the MSCC regions, i.e the ellipse parameters are calculated from the covariance



**Fig. 3.** (a) Test scene "doors". (b) Test scene "group". (c) Test scene "room"

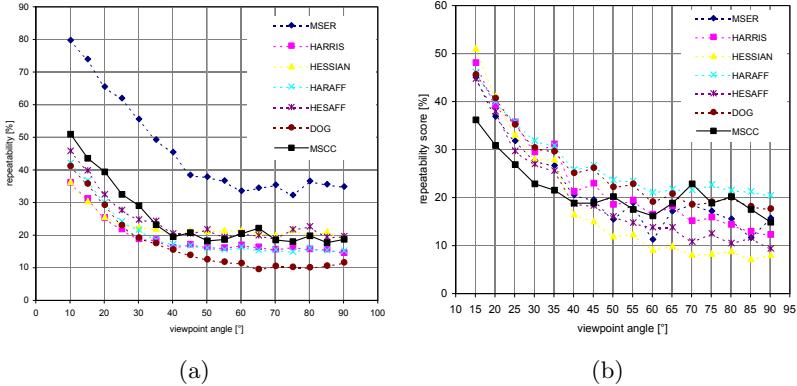


**Fig. 4.** (a) Example for detected MSCC regions. (b) Repeatability score for "doors" scene. (c) Convex hulls and fitted ellipses for detected MSCC regions

matrix of the interest points belonging to the region. We compare the repeatability score and the matching score of our MSCC detector to 4 other detectors on increasing viewpoint change up to  $130^\circ$ . For the matching score the SIFT descriptor is used. Fig. 4(b) shows the results of the MSCC detector compared to the Maximally Stable Extremal Regions (MSER) [1], the Hessian-Affine regions (HESAFF) [9], the Harris-Affine regions (HARAFF) [9] and the intensity based regions (IBR) [2]. The experiment reveals a competitive performance of our novel detector when compared to other approaches. The regions detected by our approach are consistently different from those of other detectors (see also 4.3).

## 4.2 Detector Evaluation on Non-planar Scenes

Non-planar scenes can not be evaluated with the method proposed by Mikolajczyk. We use the method proposed in [17] to evaluate the MSCC detector on non-planar scenes. We compare the results to 6 other detectors on increasing viewpoint angle additionally including the Difference of Gaussian keypoints (DOG) [5] and simple interest point detectors like Harris corners (HARRIS) and Hessian corners (HESSIAN) [8]. The compared value is the repeatability score. We use the publicly available implementation from Mikolajczyk for the other de-

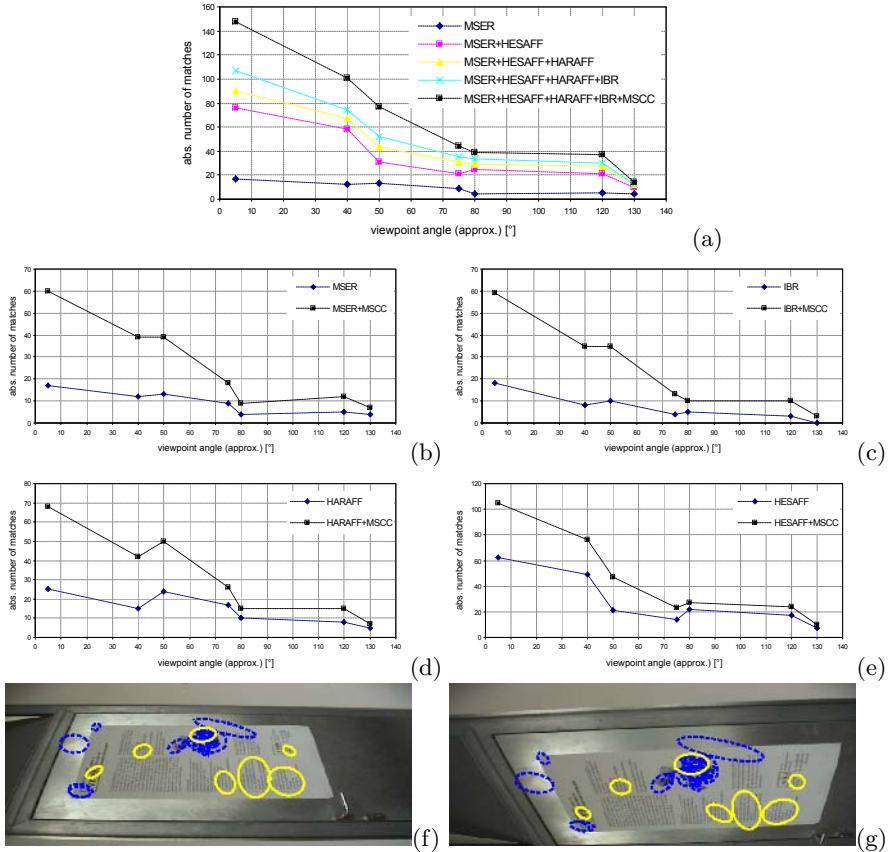


**Fig. 5.** (a) Repeatability score for "group" scene. (b) Repeatability score for "room" scene

tectors. We evaluated the detectors on 2 different complex scenes (see Fig. 3(b) and (c)). The test scene "group" shows two boxes acquired with a turntable. The second test scene "room" shows a part of an office and is of higher complexity than the first one. Both image sequences consist of 19 images and the viewpoint varies from  $0^\circ$  to  $90^\circ$ . Fig. 5(a) shows the repeatability score for the "group" scene. The best performance is obtained by the MSER detector. The proposed MSCC detector comes second and shows a repeatability score noticeable higher than the other detectors. Fig. 5(b) shows the evaluation results for the "room" scene. In this scene the performance of the different detectors is very similar and no one shows a really outstanding performance. The evaluations demonstrate that our MSCC detector is competitive to the other established detectors.

### 4.3 Combining Local Detectors

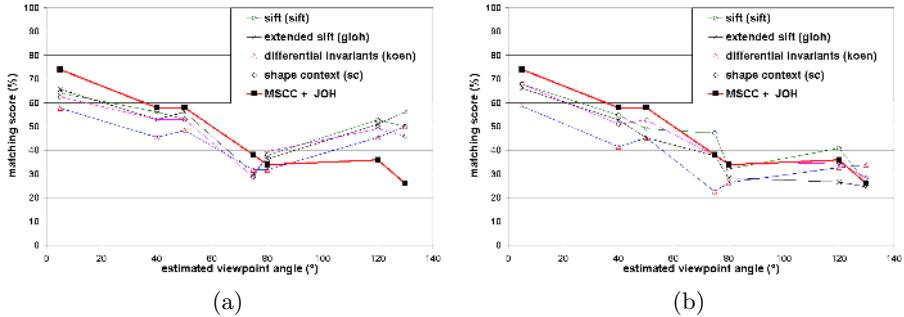
This experiment evaluates the complementarity of the MSCC detector. This is done by counting the non-overlapping correct matching regions from different detectors. Regions from different detectors are counted as non-overlapping if they do not overlap more than 40%. Matching is done using SIFT descriptors and nearest neighbor search (as implemented in Mikolajczyks evaluation framework). Fig. 6(a) shows the absolute number of matched MSER regions, MSER regions combined with HESAFF regions, combination of MSER, HESAFF and HARAFF, combination of MSER, HESAFF, HARAFF and IBR and combination of the previous detectors with the MSCC detector. Fig. 6(b)-(e) show the region numbers for combining the MSCC detector with each of the other detectors. The graphs show that our MSCC detector is able to add a significant amount of new matches to the ones of the other detectors. Fig. 6(f) and (g) show an example for  $120^\circ$  viewpoint change. The dashed dark ellipses mark the matches from the combination of MSER, HESAFF, HARAFF and IBR. The bright ellipses mark the additional matches obtained from the MSCC detector.



**Fig. 6.** Absolute numbers of non-overlapping matched regions. (a) Combining all detectors. (b) Combining MSER and MSCC. (c) Combining IBR and MSCC. (d) Combining HARAFF and MSCC. (e) Combining HESAFF and MSCC. (f),(g) Matches for combination of all detectors at 120° viewpoint change. The bright ellipses mark the additional matches obtained from the MSCC detector

#### 4.4 Descriptor Evaluation

To compare our joint orientation histogram (JOH) descriptor against others we use Mikolajczyk's evaluation framework [16]. We show the results of SIFT-keys, extended SIFT-keys, differential invariants and shape context (see Fig. 7). All the other descriptors of the framework give similar results. For MSCC detector and JOH descriptor we use the convex hull of the MSCC region points instead of ellipse fitting for region overlap calculation. Fig. 7(a) shows the matching scores (on the "doors" scene) of our JOH descriptor on MSCC regions compared to different descriptors on Hessian-Affine regions, as we found them to give best results. In contrast to our approach the descriptors for the Hessian-Affine regions are calculated on affine normalized patches. Fig. 7(b) depicts the results for



**Fig. 7.** Matching scores for images from "doors" dataset on Hessian-Affine regions (a) and MSCC regions (b) for different viewpoints

the same scene but all descriptors are calculated on MSCC regions. For MSCC regions all detectors show similar performance. The same behavior is observed for Hessian-Affine regions.

## 5 Summary and Conclusions

We have presented a novel method for the detection of distinguished regions by clustering feature primitives - the so called Maximally Stable Corner Clusters (MSCC). We have developed a novel local descriptor based on 2D joint orientation (JOH) histograms ideally suited for the properties of the detector. We have evaluated the repeatability of the MSCC under changing viewpoints and compared the performance to other established detectors on planar and non-planar scenes. The results show a competitive performance of the MSCC. Further evaluations on the combination of different detectors have shown, that our detector consistently detects regions different from those of other detectors. Finally, we evaluated the performance of our JOH descriptor against others and obtained comparable results. The results indicate, that the detector successfully enriches the variety and power of the current available set of local detectors.

## References

1. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. 13th British Machine Vision Conference, Cardiff, UK. (2002) 384–393
2. Tuytelaars, T., Gool, L.V.: Matching widely separated views based on affine invariant regions. International Journal of Computer Vision 1 (2004) 61–85
3. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the International Conference on Computer Vision. (2003)

4. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proc. 8th IEEE International Conference on Computer Vision, Vancouver, Canada. (2001) I: 525–531
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60** (2004) 91–110
6. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2003)
7. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. In: Proc. 7th European Conference on Computer Vision, Prague, Czech Republic. (2004) Vol I: 228–241
8. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference. (1988)
9. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Proc. 7th European Conference on Computer Vision, Copenhagen, Denmark. (2002) I: 128 ff.
10. Lindeberg, T.: Feature detection with automatic scale selection. *International Journal of Computer Vision* **30** (1998) 79–116
11. Brown, M., Lowe, D.: Invariant features from interest point groups. In: Proc. 13th British Machine Vision Conference, Cardiff, UK. (2002) Poster Session
12. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *International Journal of Computer Vision* **37** (2000) 151–172
13. Bigün, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: Proceedings of the IEEE First International Conference on Computer Vision, London, Great Britain (1987) 433–438
14. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Cambridge MA (1990)
15. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press Professional (1990)
16. Mikolajczyk, K., Schmid, C.: Comparison of affine-invariant local detectors and descriptors. In: Proc. 12th European Signal Processing Conference, Vienna, Austria. (2004)
17. Fraundorfer, F., Bischof, H.: Evaluation of local detectors on non-planar scenes. In: Proc. 28th Workshop of the Austrian Association for Pattern Recognition, Hagenberg, Austria. (2004) 125–132

# Spectral Imaging Technique for Visualizing the Invisible Information

Shigeki Nakauchi

Department of Information & Computer Sciences,  
Toyohashi University of Technology,  
Toyohashi 441-8580, Japan  
[naka@bpel.ics.tut.ac.jp](mailto:naka@bpel.ics.tut.ac.jp)  
<http://www.bpel.ics.tut.ac.jp>

**Abstract.** Importance of multi-spectral colour information has been remarkably increasing in imaging science. This is because the original spectrum contains much more information about the surface of target objects than perceived colour by human. This article describes our attempts to visualize the invisible information, such as the constituent distribution and internal microstructure of food and plant responses to the environmental stress, by a spectral imaging technique.

## 1 Introduction

More than 300 years ago, Sir Isaac Newton first discovered the spectrum. In his famous experiments, he separated daylight into its spectral components by passing it through a prism. Although we can see several different colours in the spectrum, Newton also claimed that “light itself is not coloured”, meaning that colour results from our visual function to encode the spectral information. Our mechanism to see colours originates in capturing the incident light by three types of colour sensors in the retina which can signal the spectral information by its relative activity. Due to this trichromatic nature, human can see only a portion of the spectral information. Even if the two incoming lights have different physical characteristics, we can not distinguish them when outputs of the three types of photo sensors are the same.

The limitation of our ability to capture the spectral information is owing to the number and the wavelength range of color sensors. Certain kinds of animals can see more colors than human because of more color sensors, e.g. five different colour sensors of Butterflies or a UV sensor of bees [1], [2]. In this sense, if we could develop an artificial eye with higher wavelength resolution and wider wavelength range, it can be superior to our visual system.

Spectral imaging is expected to be a key technology for such an artificial hyper-vision system. This article demonstrates our several attempts to visualize the information which is invisible to human using the spectral imaging technique, such as the constituent distribution and internal microstructure of food, and plant responses to the environmental stress caused by ozone.

## 2 Visualization of the Sugar Content Distribution of Melons by Near-Infrared Imaging Spectroscopy

### 2.1 Background

Recently, automated sweetness sorting machines for vegetables or fruits are getting common, and now in use in more than 1000 packing houses in Japan. However, because of uneven distribution of sugar content, some fruits sorted by this type of machine as sweet ones taste insipid. This is a limitation in point-measurement, and instead imaging-based measurement is required. We aim to develop a technique for visualization of the sugar content distribution of a melon by NIR imaging.

The spectral imaging system we are using consists of an acousto-optic tunable filter (AOTF) and a peltier-cooled monochromatic CCD camera as shown in Fig.1. It can measure the spectral information at any pixel position, ranging from 400 nm to 1,000 nm at about 2 nm intervals, instead of three channel values (R, G and B) as in conventional digital cameras.

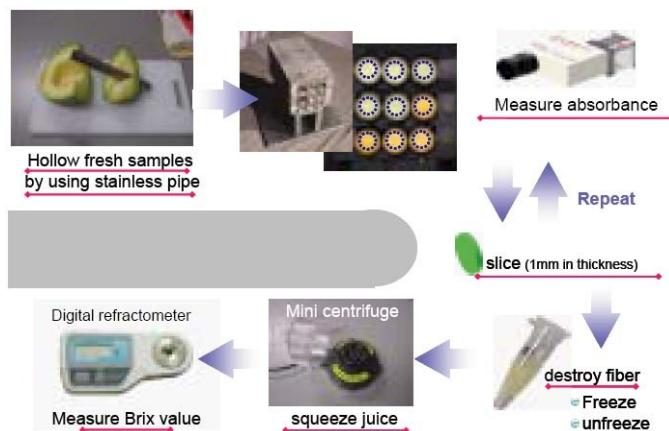


**Fig. 1.** Spectral imaging system consisting of an acousto-optic tunable filter (AOTF, a solid-state electronically tunable spectral bandpass filter) and a peltier-cooled monochromatic CCD camera. This system can measure spectral images ranging from 400 nm to 1,000 nm at about 2nm intervals

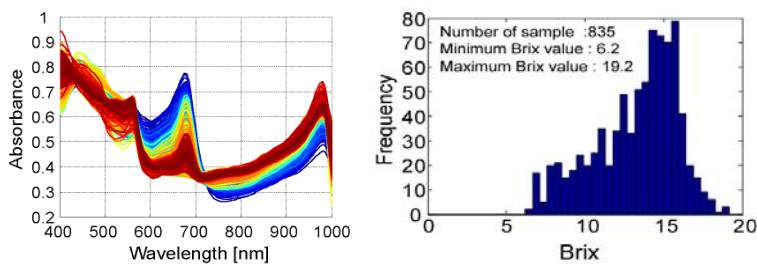
### 2.2 Method and Results

In order to predict the sugar content from measured spectra, we need to make a calibration data set, which is a pair set of absorption spectra and sugar content data (Brix) measured by a refractometer. To do this, we cut and pick up a set of sample slice from melons, measure the absorption spectra and the Brix values of slices as shown in Fig.2. By repeating this process, we obtain a calibration data set shown in Fig.3. Then we describe the relation between the absorbance and the Brix by a multivariate regression analysis. For the data set obtained from 835 samples, we could get a relatively high precision ( $r=0.89$ ).

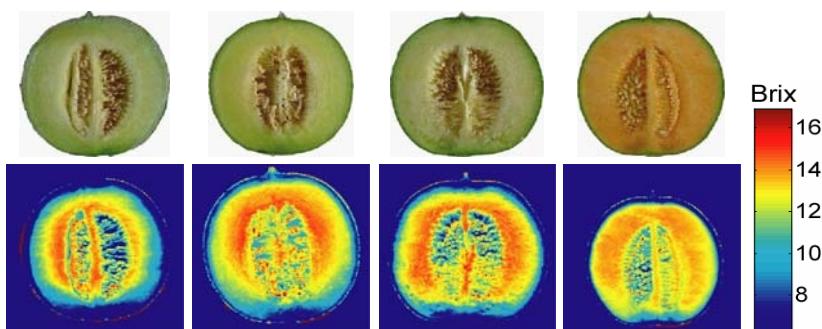
Once we have a calibration curve, the measured absorption spectra at each pixel can be mapped into the sugar content value. Fig.4 shows an example of sugar content map. From the map, we can see the difference of sugar content distribution among two varieties of melons. For example, the green and red type of varieties had different distribution, that is, the red one has more uniform sugar content distribution than the green one.



**Fig. 2.** Process for making calibration data to estimate the Brix value from spectral absorbance measured by the spectral imaging system



**Fig. 3.** Absorbance spectra (left) and Brix values (right shown as histogram) obtained from 835 samples. Colour of a curve represents the Brix values as red for high and blue for low Brix values. A multivariate regression analysis was applied to pairs of these data sets to construct a calibration curve for the sugar content



**Fig. 4.** Visualization of sugar content distributions for cut in half melons using spectral imaging technique. From the sugar content maps shown as bottom panels, the green and red types of varieties had different distributions

### 3 Visualization of the Internal Microstructure of Soybeans by Spectral Fluorescence Imaging

#### 3.1 Background

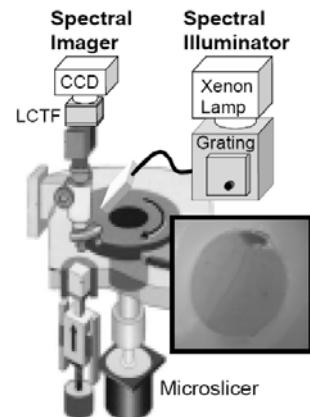
Soybeans (*Glycine soja* L.) are a very fast growing sector of agricultural production in the world market including western world, and many researches have been done on measurement of constituents to understand and improve the soybeans' quality. Here we focus on the excitation-emission matrix (EEM) to visualize the internal microstructure of soybeans.

#### 3.2 Method and Results

An EEM is a contour-like graph composed of an excitation wavelength axis, an emission wavelength axis and a fluorescence intensity axis. Every constituent can be specified by EEM pattern matching because it has a unique EEM pattern reflecting its characteristics. EEM measurements and analyses have been carried out in the archaeological field or the environmental assessment to specify old dyes on the paintings or pollutant in the sea water [3], [4]. Although there have been only a few studies of EEM application to food, it is expected that the internal structure and constituent distribution, which greatly affect food qualities, can be visualized by the measurement and analysis of an EEM at any point of food.

Measurement system composed of a micro-slicer, a spectral illuminator and a spectral imaging system as shown in Fig.5. The spectral illuminator consists of a xenon lamp and a grating spectrometer. It illuminates the sample surface with light at any wavelength from 200 nm to 1,000 nm. The spectral imaging system is made up with a liquid crystal tunable filter and a monochromatic CCD camera. The sample surface image can be captured at any wavelength from 400 nm to 1,100 nm. An EEM at any point of the sample can be measured by combining the spectral illuminator and spectral imaging system.

Process for visualizing the internal microstructure of soybeans is shown in Fig.6. The sample soybean was cut in half by the micro-slicer and fluorescence images of the cut surface were captured for various combinations of excitation and emission wavelengths. The excitation wavelength was changed from 350 nm to 570 nm at 10 nm intervals and the emission wavelength was from 400 nm to 600 nm at 10 nm intervals. 273 images were captured in total as shown in Fig.6(a).

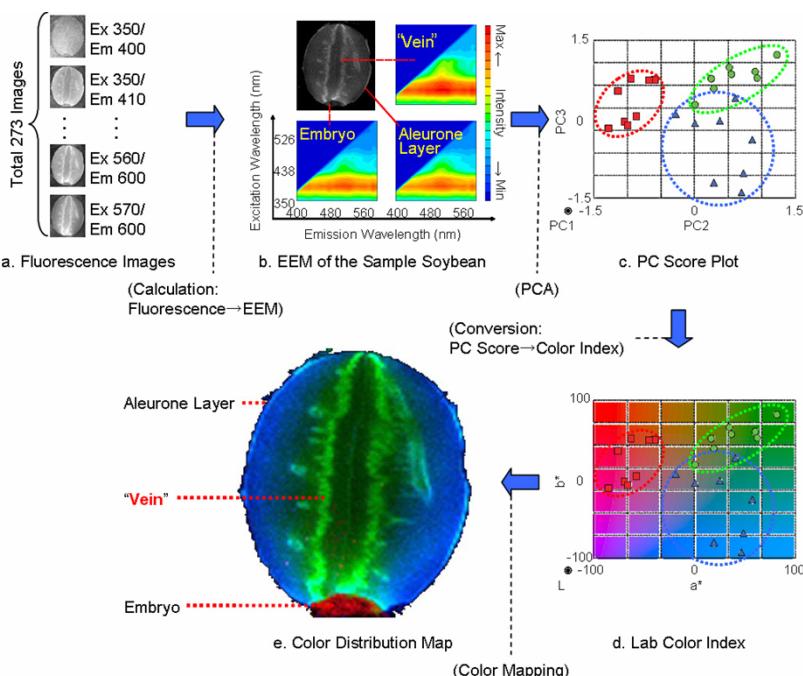


**Fig. 5.** Measurement set up for EEM imaging, consisting of a micro-slicer, a spectral illuminator and a spectral imaging system

The EEM at each pixel of the cut in half soybean image was calculated from measured fluorescence images as shown in Fig.6(b). In order to quantitatively clarify the difference among EEM patterns of different pixels, Principle Component Analysis (PCA) was applied to the EEM of each pixel. PCA is a statistical method to compress multidimensional data into two- or three-dimensional coordinate with the data variance maximized. Using this method, each pixel was plotted on a PC score plot according to its EEM pattern as shown in Fig.6(c).

As shown in Fig.6(d), the PC score plot was converted into a CIELAB color space in order to assign to each pixel. That is, if two pixels had similar EEM patterns, similar colors were assigned to these pixels and if not, they were assigned different colors in proportion to the EEM pattern difference. By applying this color mapping to each pixel, a color distribution map was developed as shown in Fig.6(e).

It is clearly seen in Fig.6(d) that the aleurone layer, the vein-like structure and the embryo were assigned as blue, green and red respectively so that they could be clearly distinguished in the color distribution map as shown in Fig.6(e). Especially, a characteristic branching of the vein-like structure could be clearly observed showing that the measurement and visualization method based on EEM is useful for the visualizing the internal structure in food.



**Fig. 6.** Visualization of the internal microstructure of soybeans by EEM imaging

## 4 Visualization of Early Ozone Stress Response in Leaves

### 4.1 Background

Ozone on the surface of the earth, that is toxic to human beings, animals and plants, is increasing worldwide [5]. Even in case when concentration of ozone is relatively low for human beings, plants receive so-called ozone stress resulting in such as low growth [6] and visible injury [7]. Especially visible foliar injury is a matter of great importance to farmers because agricultural products with visible foliar injury are not marketable and it causes serious economic loss [8]. Generally, this type of injury cannot be mitigated by common measures of plant protection. There is a possibility, however, of reducing the visible injury by fertilizer management [8] or avoiding additional watering of the plants [9]. During shortage of water, plants tend to keep closed their stomata in the leaf surface that allows ozone to enter into the leaf. Less ozone uptake means less ozone damage. Therefore, it is an important subject to detect the influences of ozone stress in its early stages and avoid visible injury by well-timed cultivation measures.

Plant physiological status has been investigated using spectral reflectance at landscape to a single leaf level [10], [11]. With the advantage of technology, more precise investigation of plant physiological status using spectral imaging devices is nowadays possible [12]. To this end, we present a method for detecting and visualizing the plants stress before the occurring of the visible foliar injury using spectral imaging technique.

### 4.2 Materials and Methods

Radish (*Raphanus sativus* L. cv. Akamaru) which is relatively sensitive to ozone [13], [14] was used as a test material in the experiment. The seed was planted in 900 ml flowerpots and was cultivated under ozone-free environment for 38 days. Two potted plants were selected for the ozone exposure and control experiment. Ozone exposure was done in a laboratory environment and sets of spectral images of the leaves were measured. The potted plant was exposed to ozone twice (time 0-16 and 49-64, respectively) and was set in the ambient air after the exposure. Spectral images were measured at every 4 or 5 hours during the exposure and at every 10 hours after the exposure.

Ozone was produced with an ultraviolet (UV) lamp in an ozone generator, and the gas containing ozone was purified by water filter and was led into a glass chamber in which the potted plant was set. The concentration of ozone was controlled by adjusting the voltage applied to the UV lamp. Fluorescent lights (7200 K) were used for illuminating the potted plants so that ozone sensitivity in plants became high. The reason for this is that plants tend to keep opened their stomata in the leaf surface that allow ozone to enter into the leaf when the intensity of illumination is high. The potted plant for control experiment was set just beside the glass chamber during ozone exposure and was set in the ambient air after exposure time.

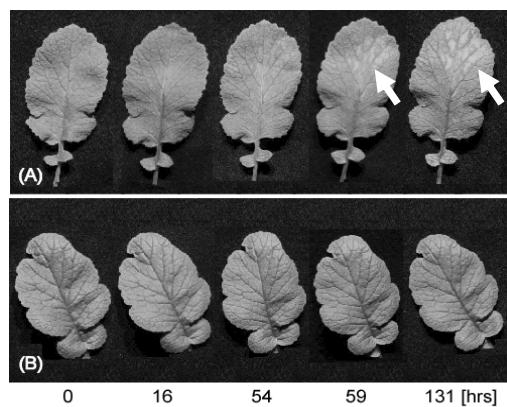
Then the potted plant was taken out of the glass chamber and the leaves were set on an experimental table covered with black cloth. An intensity image of the leaves

illuminated by halogen lamps was taken with a monochrome CCD camera through AOTF with transmitting wavelengths ranging from 400 nm to 1,000 nm. To calculate spectral reflectance images, a spectral image of a white board ( $\text{BaSO}_4$ ) was used as the reference white.

It is well-known that spectral reflectance of green leaves shows typical shape, such as absorption in a long-wavelength region and plateau in near infrared region. We here focus on the averaged reflectance values in the visible and invisible wavelength bands defined as *RED* (670-690 nm) and *NIR* (780-800 nm), in which plant physiological status is expected to be reflected, and the relative changes of those values ( $\Delta\text{RED}$  and  $\Delta\text{NIR}$ ) from the initial status due to the ozone exposure were analyzed.

### 4.3 Results

Fig.7 shows photos of leaves (A) exposed to ozone and (B) without exposure as control data. These photos were taken with a digital camera at time 0 (beginning of the ozone exposure), 16, 54, 59 and 131 hours. The leaves exposed to ozone were gradually changed during the first ozone exposure (time 0-16), however no visible injury occurred. At time 59 during the second ozone exposure (time 49-64), visible injury occurred (shown by a left arrow in Fig.7 (A)) while the controlled leaves did not show any remarkable changes. The visible injury called as necrosis is one of the common symptoms due to ozone exposure. In this case, the colour between veins of green leaves changed to white due to the collapse of leaf tissue. Visible injury developed after the second ozone exposure was terminated, and white spots were clearly found at time 131 (shown by a right arrow in Fig.7 (A)).



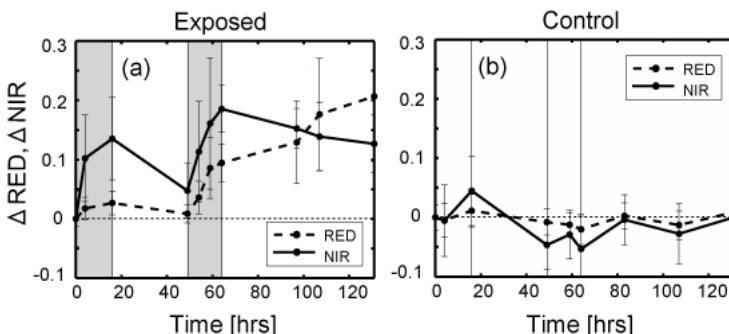
**Fig. 7.** Radish leaves (A) exposed to ozone and (B) without exposure (control). Visible injury first occurred at time 59, while the control leaves did not show any remarkable changes

To analyze the influences of the ozone exposure to leaves, spectral images were measured for the severely visible injured area of the leaf and these were compared

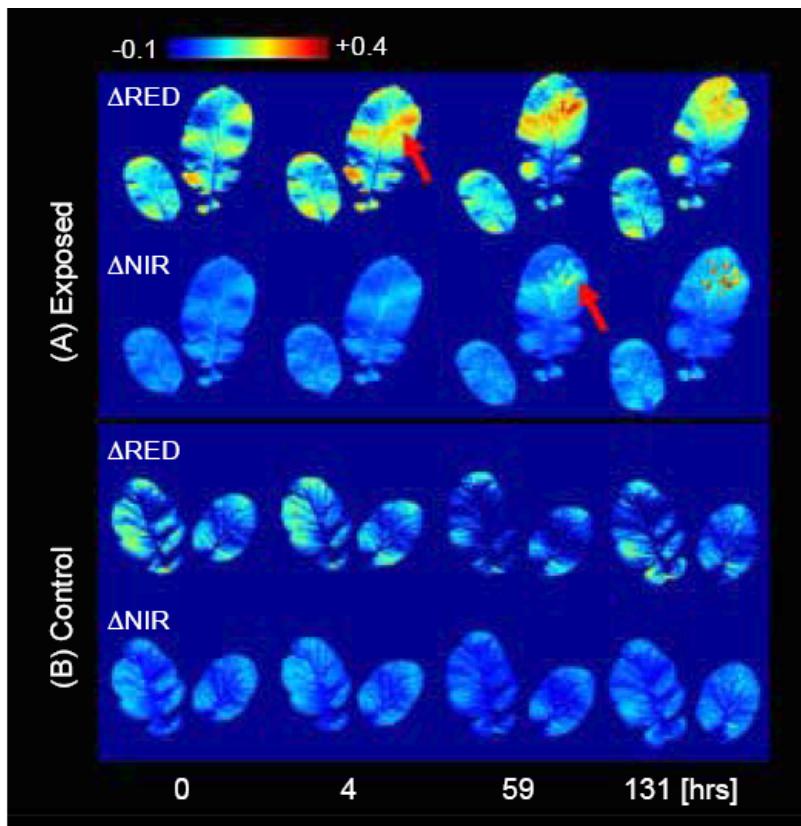
with that of the controlled leaf as follows. First, severely visible injured area was manually decided by referring to the photo taken at time 131. Then, *RED* and *NIR* data at time 0 to 131 were extracted from the decided injured area. The *RED* and *NIR* data were averaged in the area, and  $\Delta\text{RED}$  and  $\Delta\text{NIR}$  were calculated so that the initial *RED* and *NIR* data were brought to zero by subtracting them from data.

Fig.8 (a) and (b) show the relative changes of *RED* and *NIR* of the leaves exposed to ozone and controlled leaves,  $\Delta\text{RED}$  and  $\Delta\text{NIR}$ , respectively. It was found that (1)  $\Delta\text{NIR}$  increased more than  $\Delta\text{RED}$  during the first ozone exposure. (2)  $\Delta\text{NIR}$  approached to zero when the exposure to ozone was terminated before visible injury occurred at time 59. It was also found that  $\Delta\text{RED}$  increased remarkably and visible injury occurred during the second ozone exposure. After visible injury occurred,  $\Delta\text{RED}$  continued to increase and  $\Delta\text{NIR}$  decreased. Controlled leaves did not show any regular changes. Thus,  $\Delta\text{RED}$  is clearly consistent with our observations, while  $\Delta\text{NIR}$  may reflect invisible changes of leaves due to the ozone exposure.

To investigate how  $\Delta\text{RED}$  and  $\Delta\text{NIR}$  actually relates to the visible injury of the leaves, distributions of these values in whole leaf are visualized. Fig.9 shows the visualization results. Note that, in these images, average of *RED* and *NIR* values at time 0 were subtracted from each image to show the relative changes from the initial condition. In  $\Delta\text{RED}$  images, visible foliar injury occurred at time 59 which is consistent with our observations. While, in  $\Delta\text{NIR}$  images, remarkable changes are already found at time 4, at almost the same area where visible injury occurred. The controlled leaves did not show any remarkable changes. We can conclude that (3)  $\Delta\text{NIR}$  showed remarkable change in parts where visible injury occurred earlier than  $\Delta\text{RED}$ . Therefore, the proposed method can be used for detecting the ozone stress in its early stages which allow us to protect the plants from the ozone stress and avoid the visible injury.



**Fig. 8.** Dynamic changes of  $\Delta\text{RED}$  and  $\Delta\text{NIR}$ . (a) Exposed to ozone and (b) controlled. During the first exposure (time 0-16),  $\Delta\text{NIR}$  increased more than  $\Delta\text{RED}$ . When the exposure was terminated (time 16-49),  $\Delta\text{NIR}$  approached zero. During the second exposure (time 49-64),  $\Delta\text{RED}$  increased remarkably and visible injury occurred at time 59. Control did not show any regular changes



**Fig. 9.** Changes of reflectance in visible ( $\Delta$ RED) and invisible ( $\Delta$ NIR) wavelength bands of leaves (A) exposed to ozone and (B) without exposure. At time 59,  $\Delta$ RED changed at the area where visible injury occurred. At time 4, however,  $\Delta$ NIR already showed remarkable change at almost the same area. The controlled leaves did not show any remarkable changes

## 5 Conclusions

In this article, several applications of spectral imaging technique were demonstrated, mainly focusing on the visualization of the invisible information of food and plants. Spectral imaging technique is characterized by its ability to capture enormous amount of information. Due to recent progress of the optical technology, several equipments for the spectral imaging allow us to do measurements in a laboratory scale. It is expected that the spectral imaging technique can be applied to several different areas. At the same time, importance of processing / analyzing / storing / transmittance of the measured spectral images may also increase.

## Acknowledgements

Author thanks T.Suzuki for his efforts on measurements of spectral images of melons; Dr. Sugiyama and Dr. Tsuta for EEM measurements of soybeans; Dr. Miyazawa and H.Iwasaki for measurements of plant responses to ozone. This work was partially supported by Cooperation of Innovative Technology and Advanced Research in Evolutional Area, and the 21st Century COE Program "Intelligent Human Sensing".

## References

1. Kinoshita, M., Shimada, N., Arikawa, K.: Colour vision of the foraging swallowtail butterfly *papilio xuthus*. *J.Exp.Biol.* 202 (1999) 95-102
2. Menzel, R., Shmida, A.: The ecology of flower colors and the natural colour vision of insect pollinators: The Israeli flora as a study case. *Biol. Rev.* 68 (1993) 81-120
3. Shimoyama S., Noda Y., Katsuhara S.: Non-destructive Determination of Colorants Used for Traditional Japanese Ukiyo-e Woodblock Prints by the Three-dimensional Fluorescence Spectrum Using Fibre Optics. *Bunseki Kagaku* 47(2) (1998) 93-100
4. Booksh K. S., Murosaki A. R., Myrick M. L.: Single-Measurement Excitation/ Emission Matrix Spectrofluorometer for Determination of Hydrocarbons in Ocean Water. 2. Calibration and Quantitation of Naphthalene and Styrene. *Analytical Chemistry* 68(20) (1996) 3539-3544
5. Akimoto, H.: Global air quality and pollution, *Science* 302 (2003) 1716-1719
6. Gregg, J.W., Jones, C.G., Dawson, T.E., Urbanization effects on tree growth in the vicinity of New York City. *Nature* 424 (2003) 183-187
7. Novak, K., Skelly, J.M., Schaub, M., Kräuchi, N., Hug, C., Landolt, W., Bleuler, P.: Ozone air pollution and foliar injury development on native plants of Switzerland. *Environ Pollut.* 125(1) (2003) 41-52
8. Nouchi, I.: Agricultural countermeasures for avoiding crop injury from ozone in Japan. *J. Agric. Meteorol.* 59 (2003) 59-67
9. Kostka-Rick, R.: Biomonitoring, <http://www.biomonitoring.com/e/ozon.html>
10. Bowman, W.D.: The relationship between leaf water status, gas exchange, and spectral reflectance in cotton leaves. *Remote Sens. Environ.* 30 (1989) 249-255
11. Peñuelas, J. and Filella, I.: Visible and near-infrared reflectance techniques for diagnosing plant physiological status. *Trends in Plant Science* 3 (1998) 151-156
12. Aario, S., Kauppinen, H., Silvén, O. and Viilo, K.: "Imaging spectrography of greenhouse plants for vitality estimation," in Proceedings of the 3rd International Conference on Computer Vision Systems (ICVS 03), International Workshop on Spectral Imaging, 2002, pp. 49-55.
13. Kostka-Rick, R., Manning, W.J., Radish (*Raphanus sativus* L.): A model for studying plant responses to air pollutants and other environmental stresses. *Environ Pollut.* 82(2) (1993) 107-38
14. Izuta, T., Miyake, H., Totsuka, T.: Evaluation of air-polluted environment based on the growth of radish plants cultivated in small-sized open-top chambers. *Environ Sci.* 2 (1993) 25-37

# Bayesian Image Segmentation Using MRF's Combined with Hierarchical Prior Models

Kohta Aoki<sup>1</sup> and Hiroshi Nagahashi<sup>2</sup>

<sup>1</sup> Interdisciplinary Graduate School of Science and Engineering,  
Tokyo Institute of Technology

<sup>2</sup> Imaging Science and Engineering Laboratory,  
Tokyo Institute of Technology,  
4259 Nagatsuta-cho, Midori-ku, Yokohama, 226-8503 Japan  
[{aoki, longb}@isl.titech.ac.jp](mailto:{aoki, longb}@isl.titech.ac.jp)

**Abstract.** The problem of image segmentation can be formulated in the framework of Bayesian statistics. We use a Markov random field as the prior model of the spacial relationship between image pixels, and approximate an observed image by a Gaussian mixture model. In this paper, we introduce into the statistical model a hierarchical prior structure from which model parameters are regarded as drawn. This would give an efficient Gibbs sampler for exploring the joint posterior distribution of all parameters given an observed image and could make the estimation more robust.

## 1 Introduction

Image segmentation is an important low-level processing which could lead to higher level tasks such as object recognition, tracking and content-based indexing and retrieval. An observed image is segmented into some homogeneous regions. This can be viewed as a labeling process where each pixel in the image is assigned a label indicating the region to which it should belong. In this paper, the problem of image segmentation is formulated in a Bayesian framework.

In recent years, much attention has been attracted to Bayesian approaches that can give promising solutions to various vision problems. Because a Bayesian framework allows for the formulation of statistical models for image characteristics and the integration of prior knowledge about contextual structures, it has been applied to the design of algorithms for image segmentation [9].

We propose a statistical model with a hierarchical prior structure for image segmentation by treating model parameters as random variables and specifying their prior distributions. This allows the Gibbs sampler [5] to effectively explore a posterior distribution of model parameters derived from Bayes' theorem and provides robust estimation. The spacial relationship between image pixels is modeled as a Markov random field (MRF) prior, and the likelihood of possible pixel values is defined as a Gaussian mixture model (GMM). A line process is also used to represent the discontinuity of pixel values at the boundary between two distinct regions.

Our estimation method is performed based on the model posterior distribution rather than using maximum-likelihood (ML) procedures [10, 11]. By choosing conjugate priors of the parameters, we could present a more efficient Gibbs sampler than the sampling methods proposed in [1, 6].

## 2 Stochastic Models for Image Segmentation

Denoting a rectangular lattice of the size  $M \times N$  by  $\mathcal{S} = \{(i, j) | 1 \leq i \leq M, 1 \leq j \leq N\}$ , an observed image is a set of  $d$  dimensional feature vectors defined on  $\mathcal{S}$ , i.e.,  $\mathcal{Y} = \{\mathbf{y}_s | s \in \mathcal{S}\}$ . If  $k$  is the number of regions constituting the observed image, the set of possible labels can be represented by  $\mathcal{K} = \{1, \dots, k\}$ . When a set of random variables  $\mathcal{X} = \{X_s | s \in \mathcal{S}\}$  is defined on the identical lattice, labels assigned to pixel sites are assumed to be a configuration  $\{\mathbf{X} = \mathbf{x}\} = \{X_s = x_s | s \in \mathcal{S}\}$  (or abbreviated simply as  $\mathbf{x}$ ). Image segmentation is viewed as statistical inference for labeling given an observed image.

### 2.1 Mixture Models

A probability distribution of pixel values in an observed image can be approximated by a mixture model. A pixel value  $\mathbf{y}$  is assumed to be the realization of an independent and identically distributed random variable  $\mathbf{Y}$  with the probability density function

$$p(\mathbf{y} | \pi, \boldsymbol{\theta}) = \sum_{c=1}^k \pi_c f(\mathbf{y}; \boldsymbol{\theta}_c), \quad (1)$$

where  $\pi = (\pi_1, \dots, \pi_k)$  are called the mixture weights which satisfy the following two conditions;  $\pi_c > 0, \sum_{c=1}^k \pi_c = 1$ .

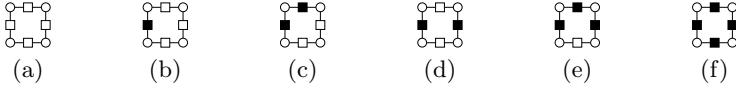
We allow the density functions of a mixture model to follow the multivariate normal distribution  $\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with a mean vector  $\boldsymbol{\mu}$  and a covariance matrix  $\boldsymbol{\Sigma}$ , that is, for  $c = 1, \dots, k$ ,

$$f(\mathbf{y}; \boldsymbol{\theta}_c) = f(\mathbf{y}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_c|}} \exp \left\{ -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{y} - \boldsymbol{\mu}_c) \right\}. \quad (2)$$

A pixel value is drawn from the distribution indicated by the corresponding label. In other words, the distribution of pixel values for each partial region comprising the observed image is approximated by the corresponding normal distribution of a mixture model.

### 2.2 Markov Random Fields

We regard labels as random variables and consider the conditional distribution of a label at each site given those at all the other sites. It is natural to think that if two sites are away from each other, there is little or no relationship between



**Fig. 1.** 6 possible patterns of edges and corresponding potentials (a)  $V_e = -2$ , (b)  $V_e = 2$ , (c)  $V_e = 1$ , (d)  $V_e = -1$ , (e)  $V_e = 1$ , (f)  $V_e = 2$ . (Pixel sites are represented by a circle and line process sites are depicted by a square.)

their labels. If the label at site  $s$  is conditioned on the configuration of a set of sites  $\eta_s$ , then  $\eta_s$  is called the neighborhood of  $s$ .

An MRF defines the Markov property of a spatial process. Although this is characterized by the local conditional distributions involving a neighborhood system, a specific representation of its joint probability distribution is derived from the MRF-Gibbs distribution equivalence [2]. Assuming the 4-neighborhood system, its corresponding clique [5] and homogeneity, the local conditional probability of a label  $x_s$  is defined as

$$p(x_s|\phi, \mathbf{x}_r, r \in \eta_s) = \frac{\exp \left\{ -\alpha_{x_s} - \sum_{r \in \eta_s} \beta V_c(x_s, x_r) \right\}}{\sum_{c \in \mathcal{K}} \exp \left\{ -\alpha_c - \sum_{r \in \eta_s} \beta V_c(c, x_r) \right\}}, \quad (3)$$

where  $\phi = (\alpha_1, \dots, \alpha_k, \beta)$  are the parameters of an MRF, and  $V_c(x_s, x_r) = -1$  if  $x_s$  and  $x_r$  have the same label and 1 otherwise. The energy function of the corresponding Gibbs distribution takes the following form

$$U(\mathcal{X}|\phi) = \sum_{s \in \mathcal{S}} \alpha_{x_s} + \sum_{s \in \mathcal{S}} \sum_{r \in \eta_s} \beta V_c(x_s, x_r). \quad (4)$$

To calculate the Gibbs distribution, the partition function that is the sum over all possible configurations in the label configuration space  $\mathbb{X}$  needs to be evaluated. The evaluation however is generally intractable, so we can approximate it by a pseudo-likelihood [3], which is provided by taking over all pixel sites the product of the conditional probability given by Eq. 3.

### 2.3 Line Processes

A line process [5] comprises a lattice of random variables  $\mathcal{L} = \{L_t | t \in \mathcal{D}\}$  located between vertical or horizontal pair of pixel sites, which can be used to prevent *oversmoothing* that may occur at discontinuities among neighboring pixel sites. The process takes either value of the binary labels  $\{0, 1\}$  indicating that an edge element is absent or present, respectively. The joint prior on both labels and lines is factored as  $p(\mathcal{X}, \mathcal{L}|\phi) = p(\mathcal{X}|\phi, \mathcal{L})p(\mathcal{L})$ . This formulation describes that the label process prior is conditioned on the configuration of the line process.

The corresponding Gibbs energy function is written as

$$\begin{aligned} U(\mathcal{X}, \mathcal{L}|\phi) &= U(\mathcal{X}|\phi, \mathcal{L}) + U(\mathcal{L}) \\ &= \sum_{s \in \mathcal{S}} \alpha_{x_s} + \sum_{s \in \mathcal{S}} \sum_{s' \in \eta_s} (1 - l_{s,s'}) \beta V_c(x_s, x_{s'}) + \sum_{t \in \mathcal{D}} \gamma V_e(l_t, l_{t'} | t' \in \tau_t), \end{aligned} \quad (5)$$

where  $\gamma$  is the constant parameter which manage the balance between a line process and a label process, and  $\tau_t$  is the neighborhood system of a line site  $t$  as in [5].  $V_e(\cdot, \cdot)$  is the potential function involving six types of possible edges (shown in Fig. 1), which may be specified to reflect empirical knowledge about the characteristics found in edges: for example, they tend to lie on a straight line, while isolated elements, endings, and corners are less likely to be seen in due order. The Gibbs distribution for the line process could be approximated by a pseudo-likelihood as with the prior on the label process.

### 3 Hierarchical Prior Models

We introduce a hierarchical structure into the model prior by treating the parameters of the mixture model and two Markov random fields as random variables. The mixture model with a hierarchical prior structure for univariate variables proposed by Richardson & Green [7] can be extended to the multivariate case [8]. Furthermore, combining with an MRF which represents a spatial independence among variables, our model is applied to the segmentation of a color image.

#### 3.1 The Priors on Mixture Model Parameters

The priors on parameters  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\Sigma}_c$  are respectively the normal and Wishart distributions

$$\boldsymbol{\mu}_c \sim \mathcal{N}_d(\boldsymbol{\xi}, \boldsymbol{\kappa}^{-1}) \quad \text{and} \quad \boldsymbol{\Sigma}_c^{-1} | \boldsymbol{\rho} \sim \mathcal{W}_d(a, \boldsymbol{\rho}^{-1}). \quad (6)$$

We choose here three dimensional ( $d = 3$ ) vectors in the  $L^*u^*v^*$  color space as a set of data being observed. Then,  $\boldsymbol{\xi} = [\xi_1 \ \xi_2 \ \xi_3]^T$  are computed so that each element is the midpoint of the observed range of the corresponding data element, and letting  $R_i$  ( $i = 1, 2, 3$ ) be the length of this variable interval,  $\boldsymbol{\kappa}$  is obtained as the diagonal matrix  $\boldsymbol{\kappa} = \text{diag}(R_1^{-2}, R_2^{-2}, R_3^{-2})$ . Although these priors are not “natural conjugate” for  $(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ , where the parameters within each pair are *a priori* dependent [7], their conjugacy are useful to implement the Gibbs sampler.

When priors’ parameters, or hyperparameters, are considered as constant, there could be an inherent uncertainty about their values. Therefore an additional hierarchical structure which could deal with this uncertainty enables us to develop flexible and robust estimation techniques. The hyperparameter  $\boldsymbol{\rho}$  is regarded as a random variable being drawn from the Wishart distribution

$$\boldsymbol{\rho} \sim \mathcal{W}_d(q, \boldsymbol{\nu}^{-1}), \quad (7)$$

where  $\boldsymbol{\nu} = v\boldsymbol{\kappa}$ .

#### 3.2 The Prior on MRF Parameters

The external field parameters  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$  used in an MRF can be correlated with the mixture weights  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$  of a mixture model as

$$\alpha_c = -\log \pi_c, \quad c = 1, \dots, k. \quad (8)$$

Ignoring the term of pairwise clique potentials in Eq. 3 which represents the dependence between two neighboring sites, the probability of a label assigned to site  $s$  is represented as

$$p(x_s) = \frac{e^{-\alpha_{x_s}}}{\sum_{c \in \mathcal{K}} e^{-\alpha_c}} = \frac{\pi_{x_s}}{\sum_{c \in \mathcal{K}} \pi_c} = \pi_{x_s}. \quad (9)$$

Thus the MRF parameters  $\alpha$  turn out to be equivalent to the mixture weights  $\pi$ . We select the following Dirichlet distribution as the conjugate prior of  $\pi$ ;

$$(\pi_1, \dots, \pi_k) \sim \mathcal{D}(u, \dots, u). \quad (10)$$

Due to the approximation of a likelihood by the pseudo-likelihood, the posterior distribution of  $\beta$  will be improper under particular configurations of labels, so  $\beta$  is set *a priori* [1].

### 3.3 The Posterior Distribution

By applying the Bayes' theorem, the joint posterior density for our hierarchical model is expressed as

$$\begin{aligned} p(\mathcal{X}, \mathcal{L}, \boldsymbol{\theta}, \phi, \boldsymbol{\rho} | \mathcal{Y}) &\propto p(\boldsymbol{\rho}) p(\phi) p(\boldsymbol{\theta} | \boldsymbol{\rho}) p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta}) p(\mathcal{X} | \phi, \mathcal{L}) p(\mathcal{L}) \\ &\approx \prod_{s \in \mathcal{S}} |\boldsymbol{\Sigma}_{x_s}|^{-1} \exp \left\{ -\frac{1}{2} (\mathbf{y}_s - \boldsymbol{\mu}_{x_s})^T \boldsymbol{\Sigma}_{x_s}^{-1} (\mathbf{y}_s - \boldsymbol{\mu}_{x_s}) \right\} \\ &\times \prod_{s \in \mathcal{S}} \frac{\exp \left\{ -\alpha_{x_s} - \sum_{s' \in \eta_s} (1 - l_{s,s'}) \beta V_c(x_s, x_{s'}) \right\}}{\sum_{c \in \mathcal{K}} \exp \left\{ -\alpha_c - \sum_{s' \in \eta_s} (1 - l_{s,s'}) \beta V_c(c, x_{s'}) \right\}} \\ &\times \prod_{t \in \mathcal{D}} \frac{\exp \left\{ -\gamma V_e(l_t, \mathbf{l}_{t'}; t' \in \tau_t) \right\}}{\sum_{e \in \{0,1\}} \exp \left\{ -\gamma V_e(e, \mathbf{l}_{t'}; t' \in \tau_t) \right\}} \times p(\boldsymbol{\rho}) \prod_{c \in \mathcal{K}} p(\boldsymbol{\mu}_c) p(\boldsymbol{\Sigma}_c) p(\alpha_c). \quad (11) \end{aligned}$$

This would differ from the model for gray-scale image segmentation proposed by Barker [1] and the extended model for color image presented by Kato [6] in the point that the hierarchical prior structure attached to MRF's could allow for flexible and robust estimation.

## 4 Bayesian Estimation

A posterior distribution can be explored by using Markov chain Monte Carlo (MCMC) methods, especially the Gibbs sampler [5] in this work. Below, all the parameters to be estimated are denoted as  $\boldsymbol{\vartheta} = (\mathcal{X}, \mathcal{L}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \boldsymbol{\rho})$ .

### 4.1 Markov Chain Monte Carlo Methods

Bayesian inference of  $\boldsymbol{\vartheta}$  is based on the joint posterior distribution  $p(\boldsymbol{\vartheta} | \mathcal{Y})$  derived from the priors and the likelihood functions via Bayes' theorem. The posterior marginal distribution of each label  $x_s$  given an observed image is followed by

$$p(x_s = c|\mathcal{Y}) = \sum_{\mathcal{X} \in \mathbb{X}} \delta(x_s, c) p(\mathcal{X}|\mathcal{Y}), \quad (12)$$

where  $\delta(i, j)$  denotes the Kronecker delta. One could find  $p(\mathcal{X}|\mathcal{Y})$  by marginalizing the joint posterior probability  $p(\boldsymbol{\vartheta}|\mathcal{Y}) = p(\mathcal{X}, \mathcal{L}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \rho|\mathcal{Y})$  over all the parameters except for  $\mathcal{X}$ . Because the sum over all possible configurations of both labels and lines needs to be evaluated in Eq. 12, it is not possible to calculate this probability analytically even for images of moderate sizes. Thus, in this paper, we adopt the Gibbs sampler to perform the estimation.

Markov chain Monte Carlo (MCMC) methods can be used to construct a sequence of variables  $\{\boldsymbol{\vartheta}^{(i)} | i \in \mathbb{N}\}$  that follow a target distribution, or  $p(\boldsymbol{\vartheta}|\mathcal{Y})$  in this case, as an invariant distribution. That is, generated samples using an MCMC method can be regarded as asymptotically distributed according to the target distribution. The posterior distribution given by Eq. 12 can therefore be evaluated from such generated samples by

$$\hat{p}(x_s = c|\mathcal{Y}) = \frac{1}{n} \sum_{i=1}^n \delta(x_s^{(i)}, c). \quad (13)$$

When the problem of image segmentation is formulated in a Bayesian framework, the approach to finding the maximum *a posteriori* (MAP) estimate for the site labels is often used (for example, in [1, 6]). Because of the direct implementation of the Gibbs sampler as mentioned below, a labeling that maximizes the posterior marginal (MPM) at each pixel site is considered here as an optimal result, such as

$$x_s^* = \arg \max_{c \in \mathcal{C}} \hat{p}(x_s = c|\mathcal{Y}), \quad s \in \mathcal{S}. \quad (14)$$

In the same way, we can estimate an optimal edge element for each line site  $t$  as

$$l_t^* = \arg \max_{e \in \{0, 1\}} \hat{p}(l_t = e|\mathcal{Y}) = \begin{cases} 1 & \text{if } \#\{i|l_t^{(i)} = 0\} < \#\{i|l_t^{(i)} = 1\} \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

In other words, if the number of generated samples taking the value of 1 is larger than that of samples taking the value of 0, then  $l_t^* = 1$ . Otherwise,  $l_t^* = 0$ .

## 4.2 The Gibbs Sampler

It is not possible to directly draw samples from the model posterior distribution  $p(\boldsymbol{\vartheta}|\mathcal{Y})$ . By partitioning  $\boldsymbol{\vartheta}$  into six parts for which conjugate priors are selected respectively, the full conditional posterior distribution of each parameter where all the other parameters and the observed image are given and fixed can be obtained analytically as in [8]. Thus the Gibbs sampler for our model is composed of the following five procedures:

- (a) updating the MRF parameters  $\alpha$  (or the mixture weights  $\pi$ );
- (b) updating the mixture component parameters  $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ;
- (c) updating the labels  $\mathbf{x}$ ;

- (d) updating the lines  $\boldsymbol{l}$ ;
- (e) updating the hyperparameter  $\rho$ .

Getting started with an initial value  $\boldsymbol{\vartheta}^{(0)}$  drawn from the priors and then iterating the above process, a Markov chain  $\{\boldsymbol{\vartheta}^{(i)} | i \in \mathbb{N}\}$  with the invariant distribution  $p(\boldsymbol{\vartheta}|\mathcal{Y})$  can be constructed.

Each procedure is described below. Notice that the symbol ‘ $\cdots$ ’ being used from here denotes conditioning on the values of all other parameters and on the obtained data.

**Updating the MRF Parameters  $\alpha$ .** In updating the MRF parameters  $\alpha$ , the mixture weights  $\pi$  are first drawn from the conditional distribution as in

$$(\pi_1, \dots, \pi_k) | \cdots \sim \mathcal{D}(u + m_1, \dots, u + m_k), \quad (16)$$

where  $m_c$  is the number of pixel sites with label  $c$ , that is,  $m_c = \#\{s \in \mathcal{S} | x_s = c\}$ . Then  $\alpha$  can be computed from Eq. 8.

**Updating the Mixture Component Parameters  $(\mu, \Sigma)$ .** Updating of the mixture model parameters  $(\mu, \Sigma)$  is performed respectively, following that

$$\boldsymbol{\mu}_c | \cdots \sim \mathcal{N}_d(\boldsymbol{\kappa}_c^{-1} (m_c \boldsymbol{\Sigma}_c^{-1} \bar{\mathbf{y}}_c + \boldsymbol{\kappa} \boldsymbol{\xi}), \boldsymbol{\kappa}_c^{-1}), \quad (17)$$

$$\boldsymbol{\Sigma}_c^{-1} | \cdots \sim \mathcal{W}_d \left( a + m_c, \left[ \boldsymbol{\rho} + \sum_{s: x_s=c} (\mathbf{y}_s - \boldsymbol{\mu}_c)(\mathbf{y}_s - \boldsymbol{\mu}_c)^T \right]^{-1} \right), \quad (18)$$

where  $\bar{\mathbf{y}}_c$  is the cluster mean of the values of pixels with label  $c$ , which is computed as

$$\bar{\mathbf{y}}_c = \frac{1}{n_c} \sum_{s: x_s=c} \mathbf{y}_s, \quad s \in \mathcal{S}, \quad (19)$$

and  $\boldsymbol{\kappa}_c = m_c \boldsymbol{\Sigma}_c^{-1} + \boldsymbol{\kappa}$ .

**Updating the Labels  $x$ .** The label at each site  $s \in \mathcal{S}$  is updated according to its conditional posterior distribution,

$$\begin{aligned} p(x_s = c | \cdots) \propto & |\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{y}_s - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{y}_s - \boldsymbol{\mu}_c) \right. \\ & \left. - \alpha_c - \sum_{s' \in \eta_s} (1 - l_{s,s'}) \beta V_c(c, x_{s'}) \right\}. \end{aligned} \quad (20)$$

Compared with conditional posterior distributions for allocation variables (or the missing data) used in common mixture models [8], this contains the term where both the dependence and the discontinuity between neighboring pixel sites are taken into account.

**Updating the Lines  $\mathbf{l}$ .** The label at each line site  $t$  in the line process  $\mathcal{D}$  is updated by

$$p(l_t = e | \dots) \propto \exp \{ -(1 - e)\beta V_c(x_s, x_{s'}) - \gamma V_e(e, \mathbf{l}_{t'}; t' \in \tau_t) \}, \quad (21)$$

where  $x_s$  and  $x_{s'}$  are the labels at pixel site  $s$  and  $s'$ , respectively, which lie to the sides of line site  $t$ .

**Updating the Hyperparameter  $\rho$ .** The hyperparameter  $\rho$  is drawn from its conditional distribution,

$$\rho | \dots \sim \mathcal{W}_d \left( q + ka, \left[ \boldsymbol{\nu} + \sum_{c=1}^k \boldsymbol{\Sigma}_c^{-1} \right]^{-1} \right). \quad (22)$$

### 4.3 Convergence Diagnostics

An initial sample  $\boldsymbol{\vartheta}^{(0)}$  of the parameters might be drawn from the corresponding priors. To reduce the dependency of an estimate given by Eq. 13 on the initial sample, the first  $m$  iterations of the Gibbs sampler would be discarded. In other words, inference is performed using only subsequent samples after they are thought to achieve convergence to an invariant distribution. Many different approaches have been proposed for determining a suitable value of  $m$ , or for diagnosing whether or not the chains converge; see, for example, the expository and comparative review by Cowles and Carlin [4].

We adopt here one of the simplest methods, where the label changing rate between consecutive samples is computed from obtained label samples  $\mathbf{x}^{(t)}$  by

$$r = 1 - \frac{1}{MN} \sum_{s \in \mathcal{S}} \delta(x_s^{(t-1)}, x_s^{(t)}), \quad (23)$$

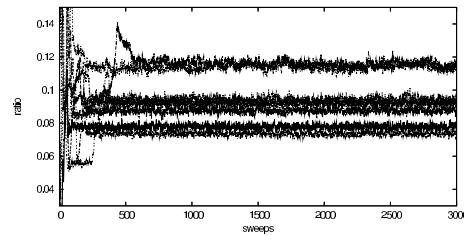
and then convergence diagnostics is done for several sample paths by assessing whether each the changing rate can stably fluctuate without depending on the initial value.

## 5 Experiments

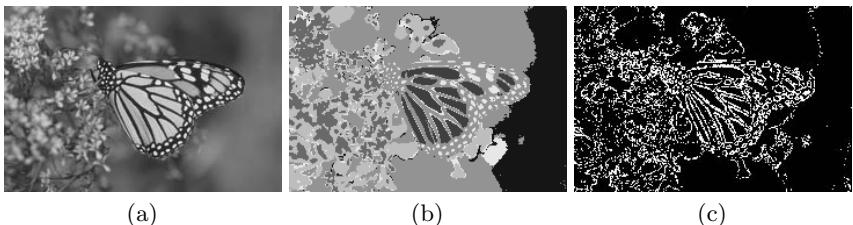
The proposed statistical model has been used to segment several standard color images. The pixel  $RGB$  values in an observed image are converted into coordinates in the  $L^*u^*v^*$  color space which is designed to be perceptually uniform, and then are normalized so that each element has the mean of zero and the variance of one. The number of regions comprising an image is suitably specified: we here assume that  $k = 6$ . The constant parameters are set referring to the literature [7], such as  $a = d + 3$ ,  $q = d + 1$ ,  $v = 100q/a$ ,  $u = 1$ . In our experiments, the pairwise clique coefficient  $\beta$  and the parameter  $\gamma$  are both set to the value of 0.1.

For the image shown in Fig. 3(a), several sample paths were constructed using the Gibbs sampler, where each path consists of 3,000 sample points. The variations of the label changing rates computed from Eq. 23 are presented in Fig. 2. This shows that the first about 1,000 sample points might be dependent on initial values and the subsequent points could be thought to be distributed according to the invariant distribution. The same thing can be said about the image shown in Fig. 4(a). Hence the first 1,000 sample points are discarded and the remaining 2,000 points are used to estimate both optimal labels and lines from Eq. 13, 14 and 15.

Fig. 3(b) and 4(b) show the segmentation results of the images in Fig. 3(a) and 4(a), respectively. The corresponding estimates of edge elements given by



**Fig. 2.** Variations of the label changing rates for sample paths generated by Gibbs sampling



**Fig. 3.** Results for image *monarch* (a) Original (b) Segmentation (c) Edge elements



**Fig. 4.** Results for image *home3* (a) Original (b) Segmentation (c) Edge elements

Eq. 15 are shown in Fig. 3(c) and 4(c). We could use the same parameter settings as these experiments to segment other images. In other words, this method robustly performs image segmentation without laborious parameter tuning.

## 6 Conclusions

In this paper, we have formulated the problem of image segmentation in the framework of Bayesian statistics and have proposed an approach to robustly estimating the posterior distribution of all the model parameters given an observed image. We have developed the hierarchical prior structure by regarding the model parameters as random variables and assuming the suitable distributions. This enables us to effectively implement the Gibbs sampler and to perform robust estimation. Experimental results show that our approach could achieve the favorable image segmentation.

## References

1. S. A. Barker, Image Segmentation using Markov Random Field Models, PhD thesis, University of Cambridge, 1998.
2. J. E. Besag, Spatial Interaction and The Statistical Analysis of Lattice Systems, *Journal of the Royal Statistical Society B*, vol. 36, pp. 192-236, 1974.
3. J. E. Besag, On The Statistical Analysis of Dirty Pictures, *Journal of the Royal Statistical Society B*, vol. 48, pp. 259-302, 1986.
4. M. K. Cowles and B. P. Carlin, Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review, *Journal of the American Statistical Association*, vol. 91, pp. 883-904, 1996.
5. S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.
6. Z. Kato, Bayesian Color Image Segmentation Using Reversible Jump Markov Chain Monte Carlo, Research Report PNA-R9902, CWI, 1999.
7. S. Richardson and P. J. Green, On Bayesian Analysis of Mixtures with An Unknown Number of Components, *Journal of the Royal Statistical Society B*, vol. 59, pp. 731-792, 1997.
8. M. Stephens, Bayesian Methods for Mixture of Normal Distributions, PhD thesis, University of Oxford, 1997.
9. Z. Tu and S.-C. Zho, Image Segmentation By Data-Driven Markov Chain Monte Carlo, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, 2002.
10. J. Zhang, The Mean Field Theory in EM Procedures for Markov Random Fields, *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2570-2583, 1992.
11. Y. Zhang, M. Brady, and S. Smith, Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm, *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45-57, 2001.

# Feature Extraction for Oil Spill Detection Based on SAR Images

Camilla Brekke<sup>1,2</sup> and Anne H.S. Solberg<sup>2</sup>

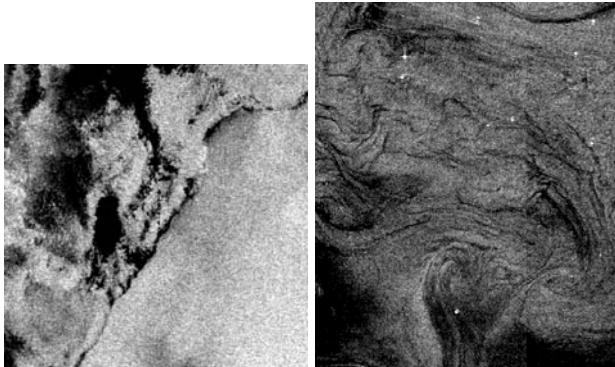
<sup>1</sup> Norwegian Defence Research Establishment,  
PO Box 25, NO-2027 Kjeller, Norway

<sup>2</sup> Department of Informatics, University of Oslo,  
PO Box 1080 Blindern, 0316 Oslo, Norway

**Abstract.** Algorithms based on SAR images for the purpose of detecting illegal oil spill pollution in the marine environment are studied. This paper focus on the feature extraction step, aiming at identifying features that lead to significant improvements in classification performance compared to earlier reported results. Both traditional region descriptors, features tailored to oil spill detection and techniques originally associated with other applications are evaluated. Experimental results show an increase from 89% to 97% in the number of suspected oil spills detected.

## 1 Introduction

Spaceborne Synthetic Aperture Radar (SAR) has proven to be the most efficient satellite sensor for oil spill monitoring of the worlds oceans. Oil spills correlate very well with the major shipping routes, and do often appear in connection to offshore installations. When taking into account how frequent illegal discharges appear, controlled regular oil spills can be a much greater threat to the marine environment and the ecosystem than larger oil spill accidents like the Prestige tanker accident in 2002. Oil spills appear as dark areas in SAR images because oil dampens the capillary waves on the sea surface. A part of the oil spill detection problem is to distinguish oil slicks from other natural phenomena (....) that dampen the short waves and create dark patches in a similar way. Oil slicks may include all oil related surface films caused by oil spills from oilrigs, leaking pipelines, passing vessels as well as bottom seepages, while look-alikes do include natural films/slicks, grease ice, threshold wind speed areas (wind speed < 3 m/s), wind sheltering by land, rain cells, shear zones, internal waves, etc. (see Fig. 1). These ambiguities put a challenge on the selection of suitable features. Research in the field of automatic processing of SAR images in order to detect illegal oil pollution has been ongoing for more than a decade. Several papers, describing fully automatic or semi automatic systems, have been published, e.g. [1, 2, 3]. Little attention seems to have been given the feature extraction step, where parameters used to discriminate oil spills from other phenomena on the sea surface are extracted. An early study on feature extraction based on ERS



**Fig. 1.** Look-alikes that occur frequently in low wind areas. ©ESA/KSAT 2003

images is described by Solberg et al. [4], and an evaluation of the discrimination efficiency of typically used features can be found in Topouzelis et al. [5].

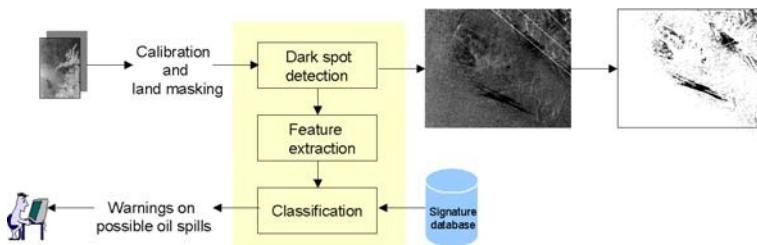
Segmentation of dark spots and feature extraction are a crucial part of algorithms for oil spill detection. If a slick is not detected during segmentation, it cannot be classified correctly. If the features have good discriminatory power, the classification problem will be easier and several classifiers can work. We hereby present results from a study aiming at identifying suitable features that lead to significant improvements in classification performance for ENVISAT ASAR Wide Swath Mode (WSM) images. Section 2 describes the automatic algorithm that constitutes the fundament of our work. Section 3 outlines the experiment design and presents the results. Finally, the conclusion can be found in Sect. 4.

## 2 The Automatic Oil Spill Detection Algorithm

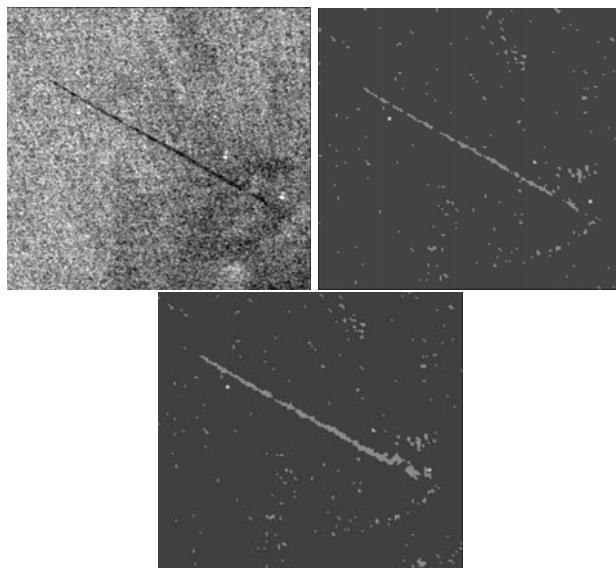
The framework of this study is a fully automatic advanced oil spill detection algorithm developed by Norwegian Computing Center (NR) and University of Oslo. It was originally intended to process ERS-1/-2 SAR images, but it has been extended to work for RADARSAT-1 and ENVISAT. The algorithm includes sensor specific modules for dark spot detection, feature extraction and a classifier discriminating between oil spills and look-alikes (see Fig. 2). Pre-processing, consisting of converting a land mask to the image grid (to avoid re-sampling the speckle pattern) and a normalization of the backscatter with respect to incidence angles, is performed ahead of segmentation for ENVISAT images.

### 2.1 Dark Spot Detection

The dark spot detector applies adaptive thresholding where the threshold is set  $k$  dB below the local mean backscatter level in a large window [3, 6]. The goal is to segment out all possible oil spills. A high number of look-alikes will also be segmented out, but these will hopefully be classified as look-alikes during



**Fig. 2.** The oil spill detection algorithm and its context. Arrows indicate data flow



**Fig. 3.** Section of an ENVISAT ASAR WSM image (19 September 2003), original segmented image and the improved result

classification. The thresholding is done in a two level pyramid after applying a simple speckle filter with a small window.  $k$  is determined based on the wind level. If no wind information is available, we use the power-to-mean (PMR) ratio of the local surroundings as an indication of the number of look-alikes in the scene. The number of observed look-alikes will vary according to local wind conditions. Thin linear slicks have a tendency to be fragmented in the segmentation process. An approach, searching for additional slick pixels in an extended object-oriented bounding box surrounding the slick, has been developed. Only pixels close to an edge are accepted and merged with the original segmented image (see Fig. 3).

## 2.2 Slick Feature Extraction

After segmentation, a set of features are computed and extracted from every region above a certain minimum size.

**Existing Set of Features.** A basic set of features was described in [3]. Due to page limits, these are not described in detail here. The features are a mix of standard region descriptors and features tailored to oil spill detection (see Table 1). Not all of these features were found to be robust and yield the best

**Table 1.** Basic feature vector components

#	Feature	Description
1	WIND	The wind level in the scene.
2	MOM (Moment)	1st planar moment of the region.
3	COMPL (Slick complexity)	$C = P^2/A$ , $P$ is the perimeter and $A$ is the area of the region.
4	PMR (Power-to-mean ratio)	Homogeneity of the surroundings. Defined as $\sigma_b/\mu_b$ . $\sigma_b$ and $\mu_b$ are the standard deviation and mean of near-by background pixels.
5	LCONT (Local contrast)	Defined as $\mu_b - \mu_r$ , $\mu_b$ is the background pixel mean and $\mu_r$ is the region pixel mean.
6	THICK (Thickness)	Thickness of the region, defined as the ratio between the area of the region and the diameter of the region skeleton.
7	NOFSN (Number of small neighbours)	The number of small neighbouring regions.
8	BGRAD (Border gradient)	The mean of the magnitude of the region border gradient. Sobel is used to compute the gradients.
9	SMC (Smoothness contrast)	Defined as the ratio between the ratio of the number of region pixels and the sum of the region gradient values, and the ratio of the number of background pixels and the sum of the background gradient values.
10	AREA	The number of pixels in the region.
11	DIST (Distance)	The distance from the region to closest bright spot (ship).
12	NLN (Number of large neighbours)	The number of large neighbouring regions.
13	NREG (Number of regions)	The total number of detected regions in the scene.

description of the type of information it was meant to extract. The goal of this paper is to find new features and compare their performance to the existing.

**New Features.** The basic feature set has been extended with the features described in the following.

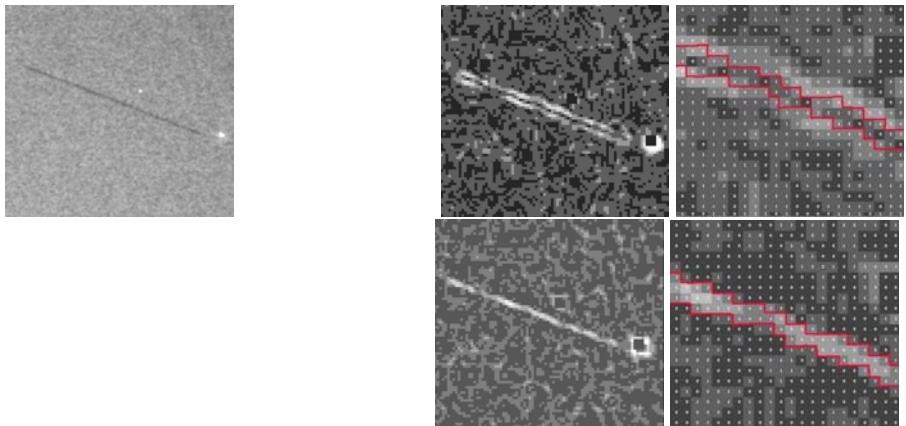
**Sobel operator.** The Sobel operator is an edge detector that has been suggested used for oil spill border gradient estimation, see e.g. [7, 3]. Originally, the mean value of the magnitude was applied in the BGRAD feature in our system (see Table 1). It works generally well, but it seems to give inaccurate

results for thin linear regions. The main problem is that the edge response does not match the real borders of the region. The top row to the right of Fig. 4 illustrates the response of the Sobel operator on the oil spill to the left in the same figure. The largest gradient magnitude appears outside the true region border. The following 4 additional convolution masks are suggested for gradient estimation of thin oil spill regions:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -4 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The magnitude of the pixel gradient is found by  $\nabla(p) = \max\{\nabla(p)_i : i = 1 \text{ to } 4\}$  where  $p$  = current pixel,  $m$  = mask. The bottom row of Fig. 4 illustrates the response to these masks. If the Sobel operator gives stronger magnitude response to any of the border pixels that value is kept, otherwise the response from the additional masks are used. The mean of this border gradient detector gives us an indication of the contrast to the surrounding background, and is used in the improved feature BGRAD\_NEW to replace BGRAD in Table 1.

Texture refers to the properties that represent the surface or structure of an object. There is no precise mathematical definition of texture due to its wide variability. In Table 1 there is no feature representing the texture of the slick it self. The PMR of the slick, defined as  $\sigma_r/\mu_r$  where  $\sigma_r$  is the standard deviation and  $\mu_r$  is the mean value of the slick, has earlier been suggested by Solberg et al. [3]. Frate et al. [1] have simply used the standard deviation of the slick as a texture measure. However, the standard deviation of the intensity values of the



**Fig. 4.** Left: Section of an ENVISAT ASAR WSM image (24th of July 2003). Top right: Response from the Sobel operator (real region borders indicated by a red line). Bottom right: Response from the improved border gradient estimation

pixels belonging to a slick is highly correlated with the area/size of the region. This is due to the inherent nature of speckle statistics. Speckle is a large problem in SAR images since even a homogeneous area has a statistical distribution with large standard deviation. As the region grows larger the variance in intensity values will increase as well. A better choice would be to look at  $\sigma_r^2/A$ , where  $\sigma_r$  is the standard deviation and  $A$  is the area of the slick. After normalization by area, the feature values of larger oil spills are comparable to smaller samples.

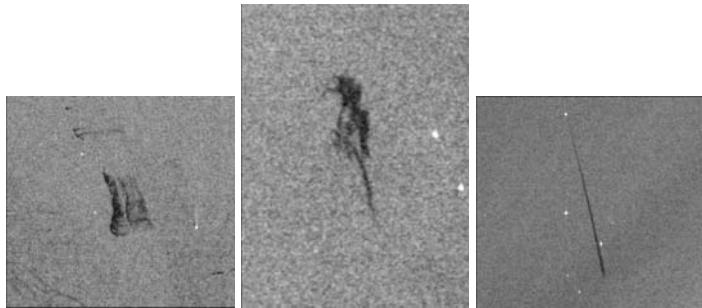
Features, based on the ratio between the perimeter  $P$  and the area  $A$ , aiming at describing the shape complexity of regions have been used in several algorithms [1, 2, 3, 5]. In [3] the complexity feature is implemented as  $C = P^2/A$  (see Table 1) while in [1] it is implemented as  $C = P/2\sqrt{\pi A}$ <sup>1</sup>. Generally, this feature is expected to get a small numerical value for regions with simple geometry, while a larger value for more complex regions. In contradiction to common intuition, the thin linear oil spill to the right of Fig. 5 gets a larger complexity value than both the others when using the formula in Table 1. Frate et al.'s [1] formula gives very similar but differently scaled results. This indicates that the ratio between perimeter and area is not a good complexity measure as it is not possible to separate complex shaped slicks from linear slicks. This weakness is also pointed out by Nixon and Aguado [8], and Topouzelis et al. [5] found that the feature gave little contribution to oil spill detection. To resolve this ambiguity we could introduce additional shape measures, or replace this measure with a more robust one. A possibility is to look at the number of branching points<sup>2</sup> in the skeleton of each region (see Fig. 6). Because we only look at the number of branching points, the information level is decreased so much that again it is often not possible to distinguish simple regions from more complex ones (e.g. a straight line would get the same feature value as an "S" shaped region). Contour or snake models are commonly applied to ultrasound image segmentation. Lobregt and Viergever [9] define local curvature  $c_i$  as the difference between the directions of two edge segments that join at a vertex:  $c_i = \hat{d}_i - \hat{d}_{i-1}$  (see Fig. 7). The local curvature has length and direction. This provides a measure of the angle between two joining edge segments. The length of the curvature vector depends only on this angle and is not influenced by the lengths of the two edge segments. In our implementation, we have traced the boundary of every region and inserted vertexes with a three-pixel spacing. The angle between two edge segments is calculated as described above, and the final CURVATURE feature is the sum of all local curvature measures (changes of slope) along the boundary.

### 2.3 Statistical Classification

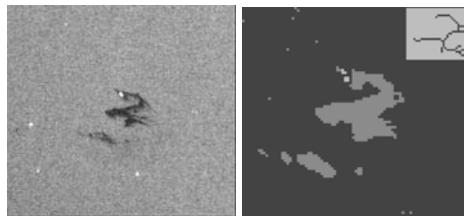
After a set of  $M$  dark spots has been detected, we want to classify them as either oil spills or look-alikes. A classification algorithm has been developed, combining

<sup>1</sup> This quantity is referred to as *compactness* in [8]. It measures the ratio between the area of the shape and the circle traced with the same perimeter:  $C = 4\pi A/P^2$ .

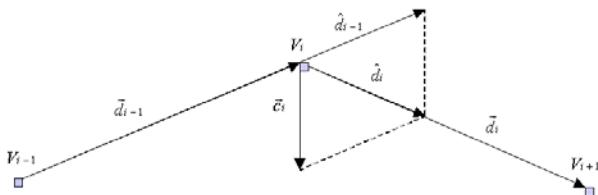
<sup>2</sup> *The number of branching points:* a point with three lines or more connected to it.



**Fig. 5.** Sections of ENVISAT ASAR WSM images (8th of March 2003, 7th of August 2003 and 12th of February 2003)



**Fig. 6.** Section of an ENVISAT ASAR WSM image (3rd of August 2003) with the segmentation result of the oil spill and its skeleton (upper right corner)



**Fig. 7.** Local curvature  $c_i$ ,  $\hat{d}_{i-1}$  and  $\hat{d}_i$  are the directions (unit vectors) of the edge segments  $d_{i-1}$  and  $d_i$  meeting at vertex  $V_i$

a statistical model for oil spills of different shapes and seen under different wind conditions, a prior model for the probability of observing oil and look-alikes, and a rule-based approach which take care of certain expert knowledge related to oil spill detection [3]. Only the features #3 - #9 from Table 1 are used to compute the probability densities. Feature #1 and #2 are used to group the samples first in two different subclasses based on wind, and then five different subclasses for each wind level according to their value of the shape descriptor. The rest of the features are included in rule-based corrections of the class-conditional densities. The classifier is trained on a large set of labelled samples. Diagonal covariance matrices are used because the number of oil spills in each sub class is small.

### 3 Performance Testing

#### 3.1 Experimental Design

Our results are based on a large set of 83 ENVISAT ASAR WSM images. We have benchmark results and aircraft verifications collected by the European Commission (EC) project Oceanides for 27 of the scenes. This is done in collaboration with Kongsberg Satellite Services (KSAT), QinetiQ, NR, German (MLZ) and Finnish (SYKE) pollution control authorities [10]. For performance testing, the SAR scenes are split into two parts. 56 of the scenes are used for training and adjusting the model parameters, and the 27 benchmark scenes are used as a validation/test set to estimate the generalization error. The training set is collected from the German and Finnish Baltic Sea, the North Sea and some along the Norwegian coastline during March to December 2003 and January to April 2004. The benchmark set is collected mainly from the German and Finnish Baltic Sea and the German North Sea between July and December 2003.

#### 3.2 Classification Results

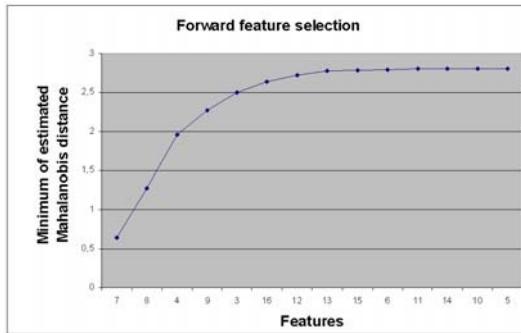
Table 2 gives a definition of the new set of features. The results from a forward selection of the features #3 - #13 in Table 1 in addition to the new features in Table 2 are plotted in Fig. 8. As the figure illustrates, adding more and more features gives little added value to the performance results. More research on which combination of features is the most optimal for oil spill detection is needed. The first line of Table 3 presents the results from classifying the complete benchmark set of 27 scenes by applying feature #3 - #9 in Table 1. A doubt category was used to mark slicks we were uncertain about. These cases are left out of the classification results. The classification was done without the rule-based corrections of the class-conditional densities described in Solberg et al. [3]. The rule-based corrections are based on the observed values of the basic

**Table 2.** Extended set of features

#	Feature	Description
14	BGRAD_NEW	The mean border gradient. A combination of Sobel and the four additional masks described in Sect. 2.2 is used as a gradient detector.
15	VAR_AREA_SLICK	Defined as the ratio, $\sigma_r^2/A$ , of the slick standard deviation $\sigma_r$ and area $A$ .
16	CURVATURE	Defined as the sum of all local curvature measures along the boundary.

**Table 3.** Classification results based on the basic set and the new feature vector

Feature set	Correctly classified oil spills	Correctly classified look-alikes
Basic set (#3-#9)	89%	90%
New set (#16, #4-#7, #14, #9, #15)	97%	90%



**Fig. 8.** Forward feature selection according to the minimum estimated Mahalanobis distance. 3: COMPL, 4: PMR, 5: LCONT, 6: THICK, 7: NOF\_SMALL\_NEIGHB, 8: BGRAD, 9: SMOOTH\_CONTR, 10: AREA, 11: DIST, 12: NOF\_LARGE\_NEIGHB, 13: NOF\_REGIONS, 14: BGRAD\_NEW, 15: VAR\_AREA\_SLICK and 16: CURVATURE

set of features on the training set. When replacing some of the features, the rules have to be modified. This is not done in the current analysis, but will be done in the near future. Thus, the rule-based corrections are left out of all performance results hereby presented. The first line of Table 3 can for this reason be used as a reference for the second line. The second line presents the final classification results after substituting the COMPL feature in Table 1 with CURVATURE, BGRAD with the improved border gradient detector BGRAD\_NEW, and adding the VAR\_AREA\_SLICK as an additional feature to the feature vector.

## 4 Conclusion

Experimental results from an evaluation of features for oil spill detection based on SAR images have been presented. We have studied properties of the border gradient and texture measures of the slicks. In addition, we have compared several features measuring geometrical complexity. The use of curvature, as adopted from the well-known concepts of contour models (snakes), is suggested as a more robust feature than those commonly applied in the oil spill remote sensing literature. The features have been evaluated on a large set of 83 ENVISAT ASAR WSM images, achieving an improvement from 89% to 97% in the number of suspected oil spills classified correctly. Further research should focus on increasing the number of 90% correctly classified look-alikes, i.e. decreasing the false alarm rate. The rule-based corrections left out in this experiment need be to modified according to the new feature set, because the rule-based corrections are important in reducing the number of false alarms. As features extracted vary between methods, our future work will also include a comparison between our final selection of features and other combinations suggested in the literature.

## Acknowledgments

The work of Camilla Brekke was funded by Norwegian Research Council and Norwegian Defence Research Establishment. The authors would like to thank Oceanides for the ENVISAT scenes.

## References

1. Frate, F.D., Petrocchi, A., Lichtenegger, J., Calabresi, G.: Neural networks for oil spill detection using ERS-SAR data. *IEEE Trans. on Geos. and Remote Sensing* **38** (2000) 2282–2287
2. Fiscella, B., Giancaspro, A., Nirchio, F., Pavese, P., Trivero, P.: Oil spill detection using marine SAR images. *Int. J. of Remote Sensing* **21** (2000) 3561–3566
3. Solberg, A.H.S., Storvik, G., Solberg, R., Volden, E.: Automatic detection of oil spills in ERS SAR images. *IEEE Trans. on Geos. and Remote Sensing* **37** (1999) 1916–1924
4. Solberg, A.H.S., Solberg, R.: A large-scale evaluation of features for automatic detection of oil spills in ERS SAR images. *Proc. IGARSS'96*, 27-31 May **3** (1996) 1484–1486
5. Topouzelis, K., Karathanassi, V., Pavlakis, P., Rokos, D.: Oil spill detection: SAR multi-scale segmentation & object features evaluation. *Proc. SPIE. Remote sensing of the ocean and sea ice 2002*, 23-27 Sept. **4880** (2003) 77–87
6. Solberg, A.H.S., Brekke, C., Solberg, R., Husøy, P.O.: Algorithms for oil spill detection in Radarsat and ENVISAT SAR images. *Proc. IGARSS'04*, 20-24 Sept. **7** (2004) 4909–4912
7. Girard-Ardhuin, F., Mercier, G., Garello, R.: Oil slick detection by SAR imagery: potential and limitation. *Proc. OCEANS'03* **1** (2003) 164–169
8. Nixon, M., Aguado, A.: Feature Extraction & Image Processing. Newnes (2002)
9. Lobregt, S., Viergever, M.A.: A discrete dynamic contour model. *IEEE Trans. on Med. Imaging* **14** (1995) 12–24
10. Indregard, M., Solberg, A., Clayton, P.: D2-report on benchmarking oil spill recognition approaches and best practice. Technical report, Oceanides, EC, Archive No. 04-10225-A-Doc, Contr. No: EVK2-CT-2003-00177 (2004)

# Light Field Reconstruction Using a Planar Patch Model

Adam Bowen, Andrew Mullins, Roland Wilson, and Nasir Rajpoot

Signal & Image Processing Group,  
Department of Computer Science,  
University of Warwick, Coventry, CV4 7AL, England  
`{fade, andy, rgw, nasir}@dcs.warwick.ac.uk`

**Abstract.** Light fields are known for their potential in generating 3D reconstructions of a scene from novel viewpoints without need for a model of the scene. Reconstruction of novel views, however, often leads to ghosting artefacts, which can be relieved by correcting for the depth of objects within the scene using disparity compensation. Unfortunately, reconstructions from this disparity information suffer from a lack of information on the orientation and smoothness of the underlying surfaces. In this paper, we present a novel representation of the surfaces present in the scene using a planar patch approach. We then introduce a reconstruction algorithm designed to exploit this patch information to produce visually superior reconstructions at higher resolutions. Experimental results demonstrate the effectiveness of this reconstruction technique using high quality patch data when compared to traditional reconstruction methods.

## 1 Introduction

A Light Field [1] captures a large array of images of a scene in a representation that allows fast reconstruction from a sufficiently arbitrary location and preserves view dependent effects. The scene is represented as a number of camera viewpoints of a common imaging plane. The pixel samples then correspond to the intersections of a ray with the image plane and the camera plane. Traditional light field reconstruction algorithms exploit this efficient data structure to rapidly sample light rays for every pixel being reconstructed. Unfortunately, it is often impractical or even impossible to capture the camera plane at sufficient resolution to represent all the desired viewpoints, resulting in noticeable artefacts in the reconstructions. Attempts have been made to alleviate this problem using variable focus and aperture [2], compensation with a low resolution model [3] and image warping [4]. Other techniques for image based rendering can also be applied to light field data, such as space carving [5] and photo-consistency approaches [6].

In fact, there is significantly more information in a light field than is exploited by a traditional reconstruction approach. Traditional reconstruction does not take advantage of the fact that all the camera views are of the same object to

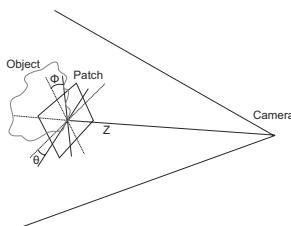
infer properties of the object. By examining the light field data we can obtain information about the object of interest that will allow us to improve our reconstructions. Typically, this is the approach taken in image warping [4]. Warping extracts disparity information from the available images to then warp them to the novel viewpoint. However, this introduces problems during reconstruction, most significantly dealing with multiple conflicting samples of the same pixel and filling ‘holes’ in the reconstructed image. These problems arise because disparity information between images is not sufficient to model the shape and orientation of the surfaces present in the scene and so occlusion boundaries cannot be properly reconstructed. Other methods for computing reconstructions from light fields include photo-consistency [6] and space carving [5]. Using a photo-consistency approach [6] for reconstruction is very slow as not much preprocessing can be performed, whilst using a space carving [5] approach discards the view-dependent information.

We present a novel representation of the surfaces present in the scene using planar patches, and an algorithm for the reconstruction of these patches when the patch estimates may be unreliable.

## 2 Multiresolution Surface Estimation

When estimating the disparity between two images, such as with a stereo image pair, we can easily obtain an estimate of the depth of each pixel. Light field data sets (irrespective of the actual representation) have significantly more viewpoints than a single stereo pair. A surface patch provides information on not just the depth of a surface but also the normal to that surface. Figure 1 shows how these patches can be represented for a given image block using three parameters.

It is possible to describe the general projective properties of a camera using the position  $\mathbf{o}_i$  of the camera  $i$ , and the direction  $\mathbf{r}_i(x, y)$  of a ray passing through pixel  $(x, y)$  on the camera’s image plane. We compute per pixel disparity values between horizontally and vertically adjacent cameras using the multiresolution approach described in [7]. Let  $\delta x_{i,j}$  and  $\delta y_{i,j}$  respectively denote the horizontal and vertical disparity between two cameras  $i$  and  $j$ . The disparity value tells us that these two pixels correspond to the same point in 3D space, hence we can obtain an equation of the form



**Fig. 1.** The parameters for a surface patch,  $z$ ,  $\theta$  and  $\phi$

$$\mathbf{o}_i + z_i \mathbf{r}_i(x, y) = \mathbf{o}_j + z_j \mathbf{r}_j(x + \delta x_{i,j}, y + \delta y_{i,j}) \quad (1)$$

for each pair of disparity estimates and some scalars  $z_i$  and  $z_j$ . Because this is an over-constrained problem we apply a least squares solver to find a value for  $z_i$  (and not  $z_j$  because the point must lie along the ray from camera  $i$  but erroneous estimates may mean it does not lie along the ray from camera  $j$ ). Given these depth values we can obtain a cloud of points in 3D space that map out the shape of the object by solving the set of equations given by the disparity maps for camera  $i$  and equation 1 and evaluating

$$\mathbf{o}_i + z_i \mathbf{r}_i(x, y) \quad (2)$$

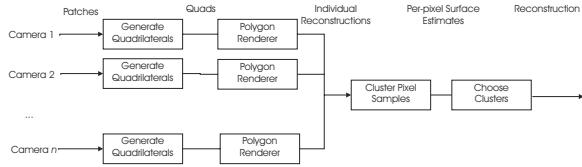
for each pixel.

Once we have a cloud of points, it is possible to obtain patch parameters by first choosing the points that correspond to a pixel block in the source image and then fitting a plane through these points using principle component analysis. Larger blocks may not be fine enough to represent details, whilst smaller blocks are prone to error. To combat these problems we apply a multiresolution approach. We start at the coarsest resolution and attempt to fit a planar patch through the entire image. If the squared error between the point cloud and the patch is greater than a preset threshold value (for the properties of the teddy light field a value of 0.1 works well) then the patch is subdivided into four quadrants and we attempt to fit a new patch through the cloud of points found in each quadrant.

If the block used to generate a patch crosses an occlusion boundary the squared error will often be very high until the block size becomes very small. Once the block size approaches  $2 \times 2$  it is often possible to fit a plane through any block in the image. However, a single plane does not model the two surfaces present at an occlusion boundary well. For this reason, we discard patches that cannot be represented using a  $4 \times 4$  patch, and patches that become oblique to the camera (patches over 80 degrees) because they are very likely to be unreliable. We generated the patch data both for perfect disparity maps found from the scene geometry and estimated disparity maps in [7].

### 3 Reconstruction Algorithm

The estimation of planar patches, as described in section 2, takes place for every camera. The generated patches are locally consistent with the viewpoint from which they were estimated. If our patch data were perfect, this would be sufficient to construct a model of the object and recreate the novel view using traditional rendering techniques. However, the patch data is computed for each camera from disparity estimates and therefore is prone to error. Because these disparity estimates are only computed between pairs of cameras, we must also consider that patches for one camera may not be consistent with patches found for a camera some distance away. Our reconstruction algorithm takes account of these potential discrepancies by dividing the process into two stages. During

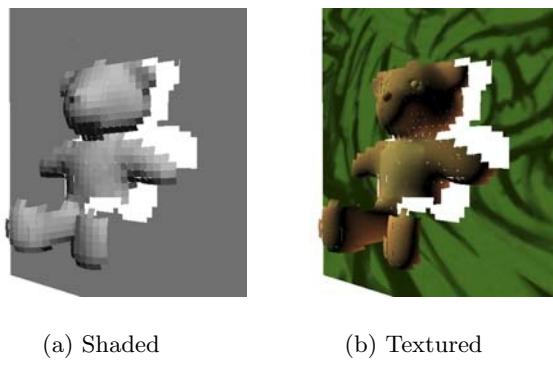


**Fig. 2.** Reconstruction Algorithm

the first stage a reconstruction is generated for every camera independently, using the patch data for that camera alone. The second stage then looks at the how consistent the data is across all the reconstructions to eliminate erroneous patches and select the best reconstruction. Figure 2 shows how the reconstruction algorithm proceeds.

### 3.1 Independent Reconstruction

Each patch is estimated using a block in the source camera's image. We generate an individual camera's estimate of the reconstruction by calculating a quadrilateral in 3D space that corresponds to the image block used to generate each patch, as illustrated by figure 1. Figure 3(a) shows the patches found from perfect disparity maps for one camera in the Teddy light field. The 'holes' seen in the image are regions that the camera cannot see, and so has no patch information for - most notably a 'shadow' of teddy is clearly visible on the background. Once the quadrilaterals have been computed, they are then textured and projected into the virtual viewpoint where a depth test is applied. Figure 3(b) shows the result of texturing and rendering the patches seen in figure 3(a) using standard OpenGL methods. We obtain an image similar to this for every available viewpoint. Only nearest neighbour interpolation is applied to the textures at this stage, to avoid blurring the textures during the second stage. This independent



**Fig. 3.** Surface patches estimated from scene geometry for a single camera

reconstruction stage can use graphics hardware to render the quadrilaterals as polygons and so is very fast.

### 3.2 Combining Reconstruction Images

Once each camera has generated an estimate of the reconstructed image, we attempt to identify the surfaces that are present at each reconstruction pixel using a clustering approach. For every pixel we wish to reconstruct, we have a colour sample and depth available from the estimate generated by each camera. Clustering these four dimensional vectors (red, green, blue and depth) gives us an estimate of the surfaces present in the reconstruction, and their corresponding depths.

To obtain these surface estimates, we apply a hierarchical clustering algorithm that finds the minimum number of clusters such that the total squared error within the cluster is below a threshold value. In our experiments we have found that, when the colour and depth values are between 0 and 1, a threshold values between 0.1 and 0.3 gave good clustering of the surfaces. The result is a variable number of clusters for each pixel that estimate the surfaces present along the ray. Small clusters may correspond to erroneous patches whilst larger clusters may correspond to genuine surfaces.

Given these clusters and their corresponding depths, we wish to select the cluster most likely to provide an accurate reconstruction. In other words, we wish to maximise the conditional probability

$$P(c_i|c_1, c_2 \dots c_n) \quad (3)$$

for the selected cluster  $c_i$  and sample clusters  $c_1, c_2 \dots c_n$ . Bayes' law gives us

$$P(c_i|c_1, c_2 \dots c_n) = \frac{P(c_1, c_2 \dots c_n|c_i).P(c_i)}{P(c_1, c_2 \dots c_n)}. \quad (4)$$

Since  $P(c_1, c_2 \dots c_n)$  is constant across our maximisation, it can ignored. This simplifies the problem to maximising

$$P(c_1, c_2 \dots c_n|c_i).P(c_i) \quad (5)$$

$P(c_i)$  is some measure of how reliable our cluster is. There are two factors to consider when calculating this measure. Firstly, we must consider the number of cameras that support the hypothesis that this cluster is a valid surface in our scene. Secondly, we must consider how much we trust the information provided by the supporting cameras. To achieve this we assign each camera  $j$  a weight  $w_j$ , the weight is computed as the dot (scaler) product of the direction of camera  $j$  and the direction of our reconstruction camera. If the direction of camera  $j$  is given by  $d_j$  and the direction of the reconstruction camera is  $d_{\text{camera}}$  then we find the weight as

$$w_j = \text{clamp}(0, (d_j \cdot d_{\text{camera}})^{\rho}, 1) \quad (6)$$

where  $\rho$  is a tuning parameter used to control how closely aligned cameras must be before they are trusted and the clamp function clamps the value to the range

[0, 1]. Typically values of 5 to 8 cut out undesirable viewpoints. We say the probability of that cluster is

$$P(c_i) = \frac{\sum_{j \in c_i} w_j}{\sum_{k=1}^C w_k} \quad (7)$$

where  $j \in c_i$  if camera  $j$  is in cluster  $c_i$  and  $C$  is the total number of cameras.

We now need to decide is how consistent the surfaces are with the selected surface - we say a surface is consistent with another surface if it occludes that surface, hence

$$P(c_1, c_2 \dots c_n | c_i) = \frac{\sum_{j=1}^n \text{occludes}(c_i, c_j)}{n} \quad (8)$$

where

$$\text{occludes}(c_i, c_j) = \begin{cases} 1 & z_i \leq z_j, \\ 0 & \text{else.} \end{cases} \quad (9)$$

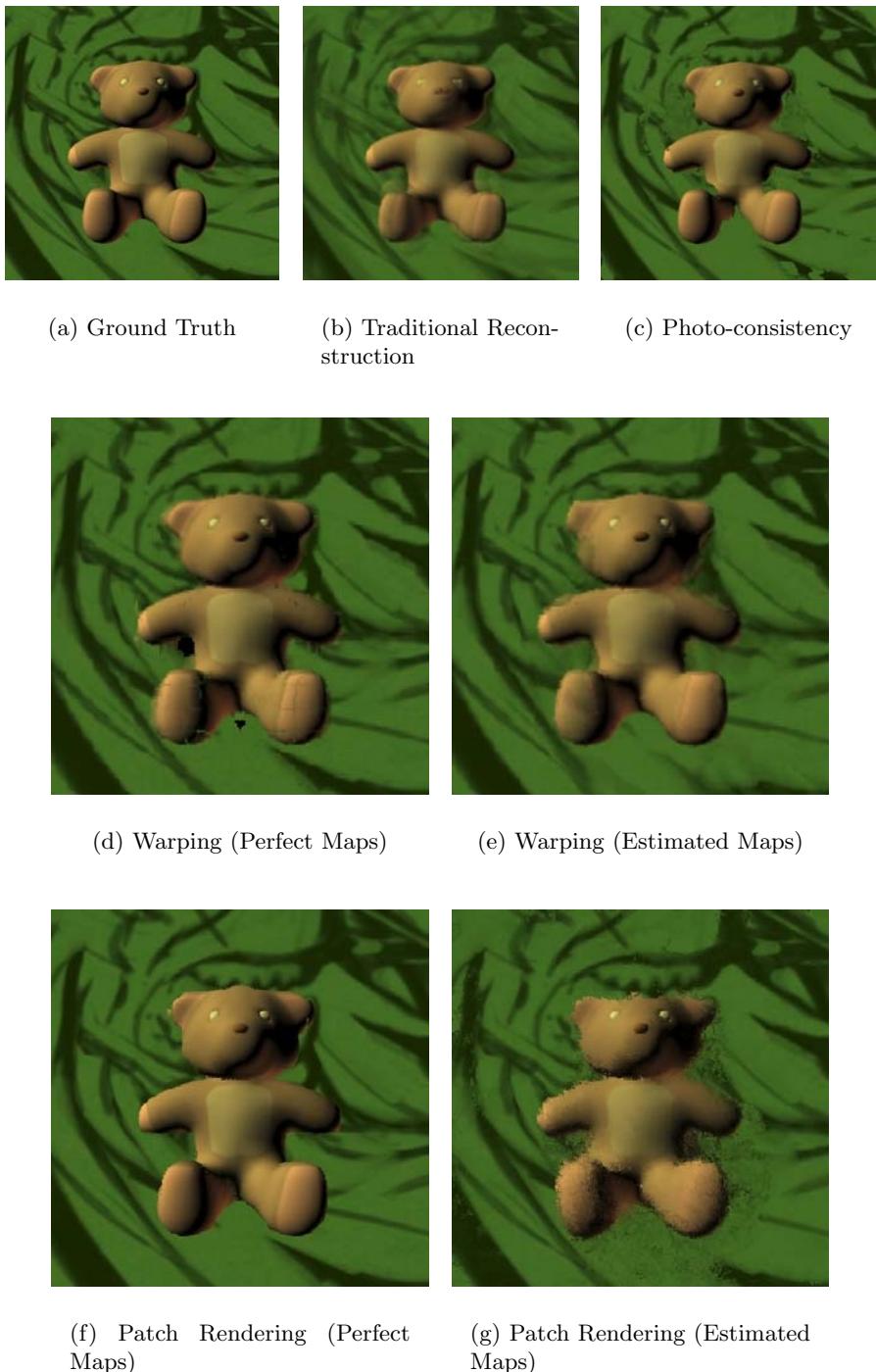
and  $z_i$  is the depth of the centroid of cluster  $c_i$ . Combining these two probabilities as in equation 5 gives us a measure of the quality of the surface represented by cluster  $c_i$  which we can then maximise for a value of  $c_i$ .

## 4 Results

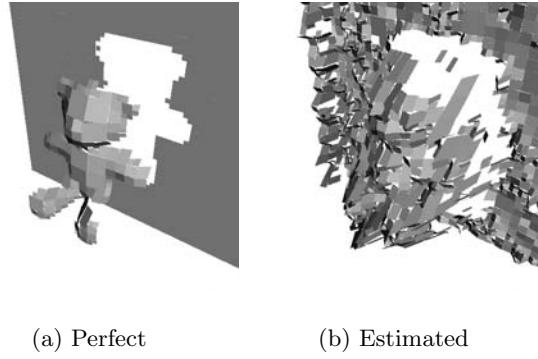
We compared results of reconstruction using our patch model based rendering with three other techniques: traditional reconstruction, warping [4], and photo-consistency based reconstruction [6]. In order to assess the quality of different reconstructions, we computed the peak-signal-to-noise-ratio (PSNR) of the reconstructed images for all viewpoints as compared to the ground truth reconstruction. The reconstruction PSNR and time complexity for all the algorithms are summarised in Table 1, where  $N$  is the number of pixels,  $C$  is the number of cameras, and  $D$  is the number of depth samples (for the photo-consistency approach). In case of the photo-consistency reconstruction, we maximised the photo-consistency metric described in [7]. For warping and patch based reconstructions, we used disparity maps from scene geometry and estimations using [7]. Whilst the PSNRs are comparable, the patch based algorithm produces noticeably sharper and higher quality reconstructions.

**Table 1.** Summary of Results

Reconstruction Algorithm	PSNR	Time Complexity
Traditional Reconstruction	24.5dB	$O(N)$
Warping (perfect disparity maps)	31.5dB	$O(N)$
Warping (estimated disparity maps)	28.4dB	$O(N)$
Photo-consistency	27.0dB	$O(N.D.C^2)$
Patch Rendering (from geometry)	32.0dB	$O(N.C^2)$
Patch Rendering (from estimates)	26.0dB	$O(N.C^2)$

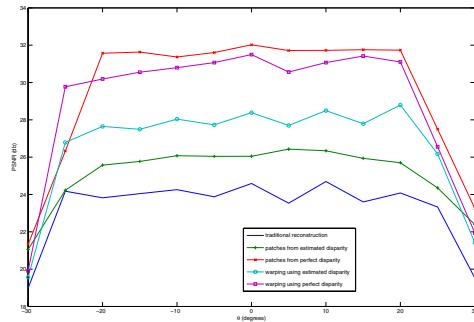


**Fig. 4.** Reconstruction Results for a Camera

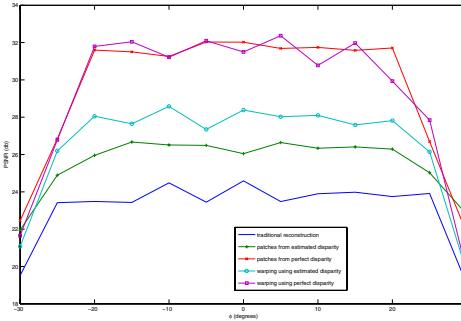


**Fig. 5.** Shaded images of some of the patches used for our reconstructions

Figure 4(b) shows the ghosting and blurring artefacts that typically result from a light field reconstruction when the camera plane is heavily under-sampled. Figure 4(c) shows the occlusion problems found with photo-consistency approaches. The photo-consistency technique performs well in unoccluded regions, but poorly in the occluded ones. Figures 4(d) and 4(e) alleviate the problems with the traditional reconstruction approach by realigning the images used in the reconstruction using disparity information. Reconstruction from perfect disparity maps suffers from hole filling problems due to occlusion between the legs and under the arm. This is because the warping approach only considers at most the 4 closest cameras for each pixel and in this case none of the cameras can see the desired region. It also suffers problems across the front of the legs. Because it has no model of how smooth or disjoint the surface is it cannot correctly interpolate nearby samples that belong to the same surface, the result is that parts of the background ‘show through’ the legs when no sample on the leg warps to the pixel. These problems are not visible when using the estimated maps because the



**Fig. 6.** Reconstruction quality (PSNR in dB) as we pan horizontally across the light field



**Fig. 7.** Reconstruction quality (PSNR in dB) as we pan vertically across the light field

error in the maps prevents the samples from aligning. However, the lack of accuracy shows through when the samples from contributing cameras are blended together. Blending samples that do not come from the same surface results in a loss of detail in the image and often undesirable blurring or ghosting in the reconstruction. Figure 4(f) shows the reconstruction using perfect patches. This reconstruction is visually significantly superior to the other methods shown due to the accurate recovery of the edges. Because these reconstructions are generated at twice the resolution of the original light field, the technique is effectively achieving super-resolution on the reconstruction - making it more suitable for reconstructing scenes at different resolutions and from closer camera positions. The notable artefacts occur where part of the ear has been lost due to few cameras providing a reliable patch and a number of single pixel errors which could easily be restored using a prior based refinement of the reconstruction. Figure 4(g) shows the reconstruction from estimated patches. Whilst the technique performs well within teddy and on the background, it has significant problems with the edges. This is caused by the poor quality of the disparity values around the edges generating noisy patches from which the reconstruction algorithm cannot recover. Figure 5 compares some of the patch estimates with the perfect estimates, illustrating the problems our algorithm has reconstructing from the underlying data. Figure 6 shows how the reconstruction PSNR varies as we pan horizontally around the light field and figure 7 as we pan vertically around the light field. The peaks correspond to regions where the viewpoint aligns more closely with the source viewpoints. There are significant drops in PSNR towards the extreme angles because the arrangement is such that no camera can see some of the background needed to create the reconstruction.

## 5 Conclusions

We have presented a novel method of representing and reconstructing from light field data sets. The traditional and warping reconstruction approaches are computationally efficient, but do not exploit all the information that can be extracted

from the data set to produce the highest quality reconstructions. Instead they rely on a high volume of data to create accurate and high quality reconstructions - which is not ideal when it comes to the coding and transmission of light field data sets. Although our method is more computationally demanding, it is still relatively simple in terms of the approach and the scalability to higher resolutions. It provides more information on the structure of a scene whilst retaining the view-dependent properties of the surfaces in the scene. We can also generate visually superior reconstructions utilising the inherent super-resolution information available in light field data sets. While our algorithm is designed to be robust to erroneous data from a fraction of the input cameras, unfortunately it does not perform well when the patch data is extremely noisy. This leads us to believe that superior methods of estimating patch data are required, we are currently working on estimating patch properties directly from the light field data sets.

## Acknowledgements

This research is funded by EPSRC project ‘Virtual Eyes’, grant number GR/S97934/01.

## References

1. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of ACM Siggraph ’96, New Orleans, LA, August 1996, ACM Press, New York (1996) 31–42
2. Isaksen, A., McMillan, L., Gortler, S.J.: Dynamically reparameterized light fields. In: Akeley, K., ed.: Proceedings of ACM Siggraph 2000, New Orleans, Louisiana, July 2000, ACM Press, New York (2000) 297–306
3. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of ACM Siggraph ’96, New Orleans, LA, August 1996, ACM Press, New York (1996) 43–54
4. Schirmacher, H.: Warping techniques for light fields. In: Proc. Grafiktag 2000, Berlin, Germany, September 2000. (2000)
5. Matusik, W.: Image-based visual hulls. Master of science in computer science and engineering, Massachusetts Institute of Technology (2001)
6. Fitzgibbon, A., Wexler, Y., Zisserman, A.: Image-based rendering using image-based priors. In: Ninth IEEE International Conference on Computer Vision, Nice, France, October 2003. Volume 2. (2003) 1176–1183
7. Bowen, A., Mullins, A., Rajpoot, N., Wilson, R.: Photo-consistency and multiresolution based methods for light field disparity estimation. In: Proc. VIE 2005, Glasgow, Scotland, April 2005. (2005)

# Spectral Estimation of Skin Color with Foundation Makeup

M. Doi<sup>1</sup>, R. Ohtsuki<sup>2</sup>, and S. Tominaga<sup>3</sup>

<sup>1</sup> Department of Telecommunications and Computer Networks,  
Faculty of Information and Communication Engineering,  
Osaka Electro-communication University,

18-8 Hatsucho, Neyagawa, Osaka, 572-8530, Japan

<sup>2</sup> Graduate School of Engineering, Osaka Electro-communication University,  
18-8 Hatsucho, Neyagawa, Osaka, 572-8530, Japan

<sup>3</sup> Department of Engineering Informatics,  
Faculty of Information and Communication Engineering,  
Osaka Electro-Communication University,

18-8 Hatsucho, Neyagawa,  
Osaka, 572-8530, Japan

**Abstract.** The analysis of skin color with makeup is needed to examine the effect of makeup to human skin. Foundation is cosmetics to cover undesirable color on skin and gives basic color to skin. It is important evaluate a change of skin color by the foundation. The present paper modeled the optics of skin with foundation makeup by two layers. Then, the surface spectrum of skin with foundation makeup is estimated by the Kubelka-Munk theory for radiation transfer in the turbid medium. The proposed algorithm can predict the spectral shape of skin surface with different types of foundation by appropriately determining model parameters. As an application of the skin reflectance estimation, we have developed a color simulator for human skin with foundation makeup. Experimental results showed a good accuracy of our estimation results.

## 1 Introduction

The estimation of human skin color is important for many fields including computer graphics, medical imaging, and cosmetic development. Especially, the analysis of skin color with makeup is needed to examine the effect of makeup to human skin. Foundation is cosmetics to cover undesirable color on skin and gives basic color to the skin. Therefore it is important to evaluate a change of skin color by the foundation. There are some reports on the analysis of skin color [1,2,5]. In the previous paper [2], the authors described a method for estimating surface-spectral reflectance of human skin based on a skin optics model. However, there is no scientific discussion on the spectral analysis of human skin with makeup.

The present paper analyzes the surface-spectral reflectance function of human skin with foundation makeup. The main idea is based on the fact that skin color with foundation makeup can be computed from the reflectance of the skin without makeup and

the optical property of the foundation. We propose a method for modeling human skin coloring with the makeup and estimating the surface-spectral reflectances by using the Kubelka-Munk theory [3,4]. Moreover we develop a skin color simulator based on the proposed spectral estimation method. Some essential points of the present paper are follows:

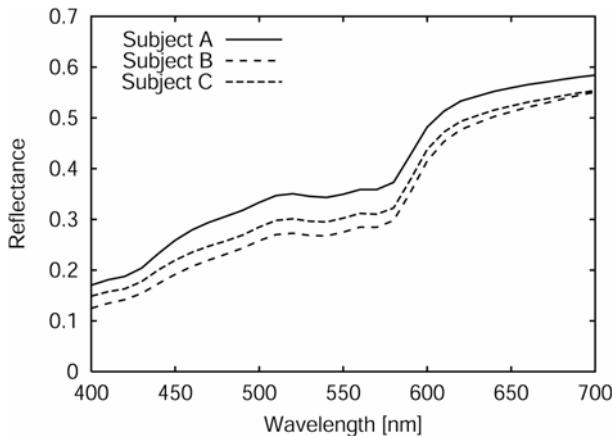
1. We do not estimate a three-dimensional color model for the human skin color, but do estimate the spectral reflectance function for the human skin surface. Note that the color model based on RGB data is device-dependent, that is unavailable for the detailed spectral analysis of skin coloring.
2. We use the Kubelka-Munk theory for calculating the transmission and reflection within the foundation layers. The Kubelka-Munk theory is used for modeling the radiation passing through a turbid medium. In this paper, the theory is applied to modeling transmission and reflection within foundation so that the surface-spectral reflectance can be predicted.
3. We evaluate the performance of the estimation method in an experiment in detail. The accuracy of the spectral reflectances estimated for facial skin is compared with the direct measurement results using a spectrophotometer.
4. We develop a color simulator of human skin with foundation makeup by using a computer graphics technique. The skin color is created based on spectral computation using the estimated spectral reactance of a subject. The system has a graphical user interface (GUI) and displays the skin color on a calibrated CRT monitor. This system simulates the change of skin color easily in the display.

## 2 Optical Model of Human Skin and Foundation

### 2.1 Human Skin

The spectral reflectance of human skin has special features on its spectral shape. Figure 1 shows surface-spectral reflectances measured from the skin surfaces of cheek of three women by a spectrophotometer. All the spectral reflectance data are represented in the visible wavelength range from 400 nm to 700 nm through this paper. One feature is that the reflectance curves increase almost monotonically as wavelength increases. This leads to the skin object colors of pink, yellow, and brown. Another feature is the "W" shaped or "U" shaped hollow in the range from 520 nm to 600 nm. This decrease in reflectance is caused by the absorption of hemoglobin. Thus, the spectral reflectance of skin is based on the influence by various pigments inside the skin tissue.

Human skin has complex optical characteristics. The skin consists of three layers of stratum corneum, epidermis and dermis. Therefore, pigments in these tissue layers influence the skin color. The epidermis has brownish pigment of melanin. The dermis has reddish pigment of hemoglobin and yellowish pigments of carotene and bilirubin. The melanin and hemoglobin cause skin color mainly. The stratum corneum does not much contribute the coloring.



**Fig. 1.** Example of spectral reflectances measured from human skin

## 2.2 Foundation

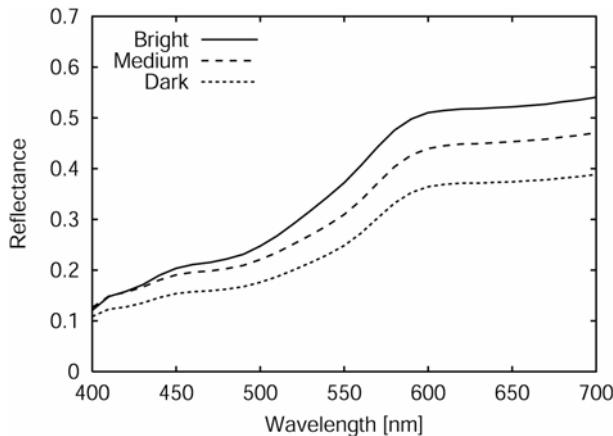
In this paper, we analyze the effect of liquid foundation. The liquid foundation consists of color pigments and medium. This foundation can be regarded as a turbid material. Therefore we use the Kubelka-Munk theory for inferring skin color with makeup of the liquid foundation.

Figure 2 shows the measured spectral reflectances for three foundation materials which layer is thick enough to be opaque optically. The measurement was done with a spectrophotometer on a black background. The colors of the foundations have resemblance to the color of the skin. However, The reflectance curves do not show such “W” shaped or “U” shaped hollow as shown in skin reflectance. The skin spectra with foundation makeup depend on the depth of foundation layer. Therefore the depth is a variable parameter in our analysis.

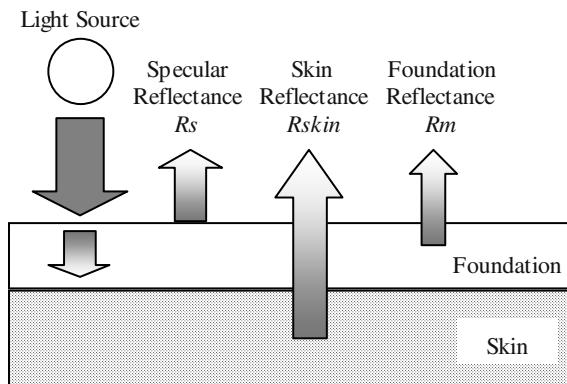
## 2.3 Optical Model

In order to determine a relationship between the spectral reflectance function of skin with makeup and the optical characteristics for skin and foundation, an optics model is assumed for the skin with foundation makeup as shown in Figure 3. The skin optics model consists of two layers of foundation and skin, where the foundation layer is the outer layer contacting the air and the skin layer is located under the foundation layer.

It should be noted that such optical properties as scattering, refracting, and absorption in each layer are functions of wavelength. In this model, incident light is partly reflected at an interface between the foundation surface and the air. The light penetrating the interface is absorbed and scattered in the foundation layer. The light ray that reaches the skin surface is reflected and absorbed in skin layer.



**Fig. 2.** Example of spectral reflectances measured from foundation materials



**Fig. 3.** Optics model for the skin with foundation makeup

### 3 Kubelka-Munk Theory

The theory by Kubelka-Munk [3,4] is used for estimating the spectral reflectance of skin with foundation makeup based on the above optics model. In general, the optical values of reflectance and transmittance within a layer consisting of turbid materials can be calculated using the Kubelka-Munk theory, where we are not consider the complex path of scattered light inside the medium. Because we assume the skin layers consisting turbid materials, use of the Kubelka-Munk theory is reasonable for the surface reflectance estimation.

Figure 4 shows the Kubelka-Munk model for radiation transfer in an absorbing medium of turbid materials. Let define the symbol  $I$  be the intensity of light traveling forward direction and the symbol  $J$  be the intensity of light traveling backward direction for the incident light. A relationship between the light intensities  $I$  and  $J$  is described as two differential equations in a variable of depth  $x$  as

$$\begin{aligned}\frac{dI}{dx} &= -SI - KI + SJ \\ -\frac{dJ}{dx} &= -SJ - KJ + SI\end{aligned}\quad (1)$$

where  $S$  and  $K$  are, respectively, coefficients of back-scattering and absorption in the media. We can derive the reflectance  $R$  and the transmittance  $T$  of the turbid layer with thickness  $D$  from solving the above equations under some assumptions as follows

$$\begin{aligned}R &= \frac{1}{a + b \coth bSD} \\ T &= \frac{b}{a \sinh bSD + b \cosh bSD} \\ a &= \frac{S + K}{S}, b = \sqrt{a^2 - 1}\end{aligned}\quad (2)$$

When the object consists of two layers, multiple reflections in the interface between the higher layer and the lower layer is considered as shown in Figure 5. In the two-layer model, the total reflectance  $R_{l,2}$  including the inter-reflection is described as

$$R_{l,2} = R_l + T_l^2 R_2 (1 + R_l R_2 + R_l^2 R_2^2 + \dots) = R_l + \frac{T_l^2 R_2}{1 - R_l R_2} \quad (3)$$

where  $T_l$  and  $R_l$  are the transmittance and reflectance of Layer 1, respectively, and  $R_2$  is the reflectance of Layer 2.

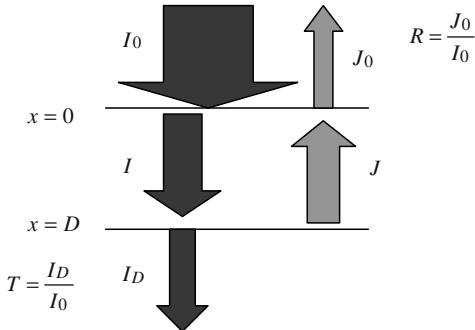
The scattering coefficients  $S$  and absorption coefficients  $K$  of the layer of turbid material can be derived from the measured spectral reflectance of a thick layer of the material and thin layer of material with background by the following equations. The thick layer must be thick enough to be opaque optically.

$$S = \frac{1}{bD} \left( \coth^{-1} \frac{a - R_0}{b} - \coth^{-1} \frac{a - R_g}{b} \right) \quad (4)$$

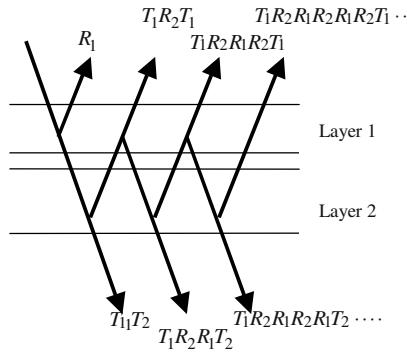
$$K = S(a - 1) \quad (5)$$

$$a = \frac{1}{2} \left( \frac{1}{R_\infty} + R_\infty \right), \quad b = (a^2 - 1)^{\frac{1}{2}}$$

where  $D$  is the thickness of the thin layer,  $R_0$  is the reflectance of the thin layer,  $R_g$  is the reflectance of the background of the thin layer and  $R_\infty$  is the reflectance of the thick layer.



**Fig. 4.** The Kubelka-Munk model for a single layer



**Fig. 5.** Two layered model

## 4 Color Estimation

The spectrum of skin with makeup is estimated by the Kubelka-Munk theory. The object color of human skin is created from three kinds of spectral data of surface-spectral reflectance, an illuminant spectrum, and a spectral response of a display device used for computer graphics. The proposed algorithm can predict the spectral shape of skin surface by appropriately determining model parameters. We control the specular reflectance and the depth of the foundation layer.

First, we defined a set of equations for estimating spectral skin reflectance using the Kubelka-Munk theory as Equation 1.

$$R(\lambda) = (I - R_s) \left( R_m(\lambda) + \frac{T_m(\lambda)^2 R_{skin}(\lambda)}{1 - R_m(\lambda)R_{skin}(\lambda)} \right) + R_s \quad (6)$$

The functions of  $\lambda$  in this equation are described

$$\begin{aligned}
 R_m(\lambda) &= \frac{I}{a_m(\lambda) + b_m(\lambda) \coth D_m b_m(\lambda) S_m(\lambda)} \\
 T_m(\lambda) &= \frac{b_m(\lambda)}{a_m(\lambda) \sinh D_m b_m(\lambda) S_m(\lambda) + b_m(\lambda) \cosh D_m b_m(\lambda) S_m(\lambda)} \\
 a_m &= \frac{S_m(\lambda) + K_m(\lambda)}{S_m(\lambda)} \\
 b_m(\lambda) &= \sqrt{a_m(\lambda)^2 - 1}
 \end{aligned}$$

We note that the spectral reflectance  $R(\lambda)$  is determined by the two parameters; the specular reflectance between the air and the skin surface  $R_s$  and the thickness of foundation layer  $D_m$ . We assume that  $R_s$  doesn't depend on wavelength.  $R_{skin}(\lambda)$  is the spectral reflectance of human skin.  $K_m(\lambda)$  is the absorption of foundation and  $S_m(\lambda)$  is the scattering of foundation. It should be noted these coefficients depend on wavelength.

The skin color with foundation makeup in RGB color space is calculated from the estimated spectrum, illuminant spectrum and spectral responses of a display device.

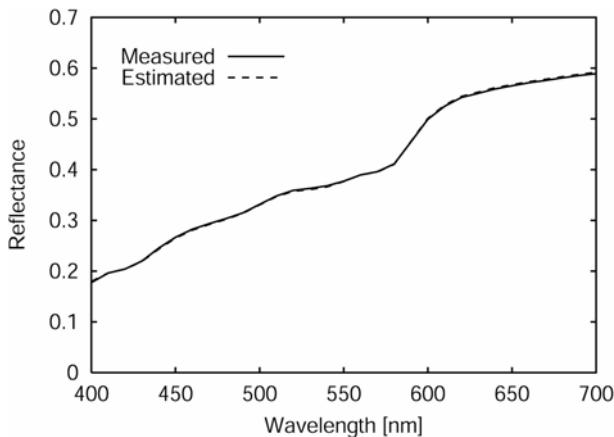
## 5 Experiments

We made some experiments to evaluate the proposed estimation method. First, we measured the spectral reflectances of both the human skins and foundation materials in different conditions by using a spectrophotometer (Konica Minolta CM2600d). The skin spectral reflectances were measured without makeup and with foundation makeup. The surface reflectances of the foundation materials were measured in a thick layer and a thin layer on a black background. The subjects were five young women. The colors of foundations used are six colors of bright beige (Bb), beige (Bc), dark beige (Bd), bright ocre (Ob), ocre (Oc), and dark ocre (Od). The measured area was the inside of a circle with 8mm diameters on the cheek of each subject. The wavelength range was from 400nm to 700nm with the interval of 10nm. The absorption coefficients and scattering coefficients of foundations were derived from the spectral reflectances for the thick and thin layers of foundations by Equations (4) and (5).

Next, the spectral reflectance of skin with foundation makeup was estimated. We determined the skin surface  $R_s$  and the thickness of foundation layer so that the estimated spectral reflectance  $R(\lambda)$  of Equation (6) can be fitted to the measured spectral reflectance in the sense of least squared error.

Figure 6 shows the estimation result for one subject with the foundation Bb. It compares the estimated spectral curve of skin reflectance with the foundation makeup to the direct measurement the same skin reflectance. The figure demonstrates a good accuracy of our estimation results. Table 1 shows numerical errors between the estimated reflectance and the measurements. Note that the errors are quite small. Table 2 shows the average value of estimated thickness for each foundation over five subjects.

The thickness values were almost same as the thickness that a person spreads foundation on her cheek with. Table 3 lists the average value of the estimated specular reflectance  $Rs$  for each foundation over five subjects. These values are convincing because the foundations we used had matt effect.



**Fig. 6.** A result of the estimation

**Table 1.** Errors in the estimation

	Bb	Bc	Bd	Ob	Oc	Od	Average
Min ( $\times 10^{-6}$ )	36	332	178	497	352	89	-
Max ( $\times 10^{-6}$ )	1696	1360	1799	4539	2225	1793	-
Ave ( $\times 10^{-6}$ )	885	732	848	2031	1102	1003	1100

**Table 2.** Average values of the estimated thickness values over five subjects

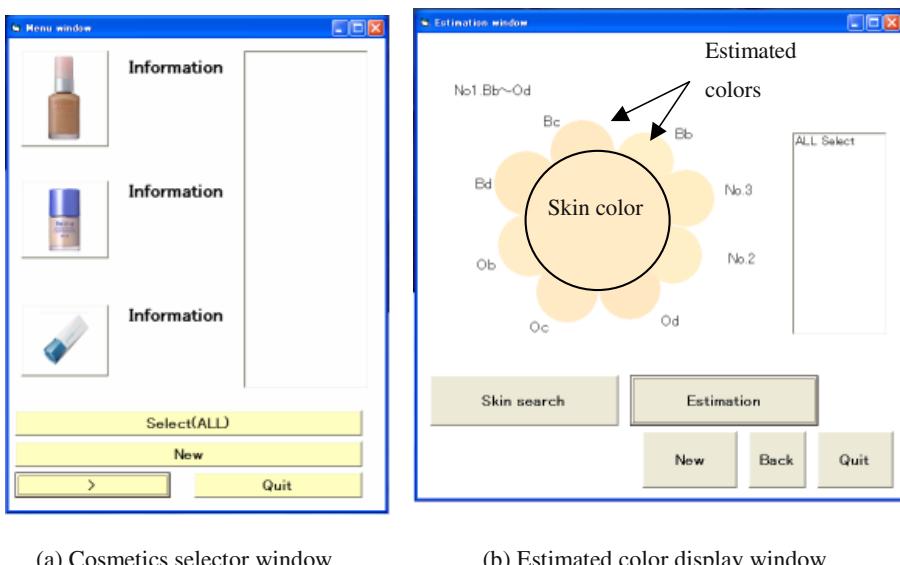
	Bb	Bc	Bd	Ob	Oc	Od
Thickness [ $\mu\text{m}$ ]	1.8	2.5	2.8	2.8	3.4	4.1

**Table 3.** Average values of estimated specular reflectance values over five subjects

	Bb	Bc	Bd	Ob	Oc	Od
Specular reflectance [%]	2.27	1.94	1.79	0.49	1.65	0.78

## 6 Application to a Skin Color Simulator

We have developed a color simulator for human skin with foundation makeup. The simulator is programmed by Microsoft Visual Basic and has a graphical user interface (GUI). Figure 7 shows the windows of the GUI. A user selects the type of cosmetics in the window in Figure 7 (a). Next, the system estimates the spectral reflectance of skin with the foundation makeup. The skin color is then calculated in terms of the tristimulus values CIE-XYZ under such a standard illumination as CIE D65. Finally, the color values are transformed into the device RGB values to produce realistic skin colors on a calibrated CRT display. Figure 7 (b) shows the display window, where the color inside the center circle shows the original skin color without makeup. Thus the user can easily compare the color of her original bare skin without makeup to the skin with any foundation makeup.



**Fig. 7.** The GUI for the color simulation

## 7 Conclusions

In this paper, we described the optical model of skin with makeup, an estimation algorithm of surface-spectral reflectance and a color simulator using the estimated reflectance. The present paper modeled the optics of skin with foundation makeup by two layers. The surface spectrum of skin with foundation makeup was estimated by the Kubelka-Munk theory for radiation transfer in the turbid medium. The proposed algorithm can predict the spectral shape of skin surface with different types of foundation by appropriately determining model parameters. The object color of human skin with makeup was created from three kinds of spectral data of surface-spectral reflec-

tance, an illuminant spectrum, and the spectral response of a display device used for display. As an application of the skin reflectance estimation, we have developed a color simulator for human skin with foundation makeup. Experimental results showed a good feasibility of the proposed method.

## Acknowledgments

We appreciate Kanebo Cosmetics Inc. Cosmetic Laboratory supporting for measurement of skin reflectance with foundation makeup.

## References

1. Angelopoulou E., Molana R., Danilidis K.: Multispectral skin color modeling, Proceedings of IEEE Conference of Computer Vision and Pattern Recognition (2001) 635-442
2. Doi M., Tanaka N., Tominaga S.: Spectral Reflectance-Based Modeling of Human Skin and its Application, Proceedings of 2nd European Conference on Color in Graphics, Imaging and Vision (2004) 388-392
3. Kubelka P.: New Contributions to the Optics of Intensely Light-Scattering Materials. Part I, Journal of Optical Society of America, vol. 38, no. 5. (1948) 448-457
4. Kubelka P.: New Contributions to the Optics of Intensely Light-Scattering Materials. Part II: Nonhomogeneous Layers, Journal of Optical Society of America, vol. 44, no. 4. (1954) 330-335
5. Tsumura N., Ojima N., Sato K., Shiraishi M., Shimizu H., Nabeshima H., Akazaki S., Hori K., Miyake Y.: Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin, acm Transactions on Graphics, vol. 22, no. 3. (2003) 770-779.

# Color Measurements with a Consumer Digital Camera Using Spectral Estimation Techniques

Martin Solli<sup>1</sup>, Mattias Andersson<sup>2</sup>, Reiner Lenz<sup>1</sup>, and Björn Kruse<sup>1</sup>

<sup>1</sup> Center for Creative Media Technology, ITN, Campus Norrköping, Linköpings Universitet,  
SE-601 74 Norrköping, Sweden

[martin.solli@medietekniker.se](mailto:martin.solli@medietekniker.se), [{reile, bjokr}@itn.liu.se](mailto:{reile, bjokr}@itn.liu.se)  
<http://www.media.itn.liu.se>

<sup>2</sup> Digital Printing Center, Mid Sweden University,  
SE-891 18 Örnsköldsvik, Sweden  
[mattias.andersson@miun.se](mailto:mattias.andersson@miun.se)  
<http://www.miun.se/dpc>

**Abstract.** The use of spectrophotometers for color measurements on printed substrates is widely spread among paper producers as well as within the printing industry. Spectrophotometer measurements are precise, but time-consuming procedures and faster methods are desirable. The rapid development of digital cameras has opened the possibility to use consumer digital cameras as substitutes for spectrophotometers for certain applications such as production control. Two methods for estimating the reflectance properties of objects from camera RGB measurements using linear estimation techniques combined with linear and non-linear constraints are presented. In the experiments, we have investigated if these techniques can be used to measure the reflectance properties of flat objects such as printed pages of paper. Reflectances were converted to CIELAB color values, and the minimization of color errors were evaluated with CIE color difference formulas. Our experiments show that a consumer digital camera can be used as a fast and inexpensive alternative to spectrophotometers for color measurements on printed substrates.

## 1 Introduction

The standard instrument to measure the reflectance properties of an object is a spectrophotometer. This gives very precise results but the measurements are done point-wise. The procedure is therefore time-consuming and it is difficult to measure spatial variations or to obtain image information. Furthermore, it is a comparatively expensive instrument. The rapid improvement of digital cameras (and their rapidly falling prices) on the other hand has triggered considerable interest in the question if these cameras can be used as measurement devices to obtain sufficiently accurate color information. If this is possible, then one could produce fast measurements for a large number of measurement points in parallel at a relatively low cost. In this paper, we describe some of our experiments in which we investigated the properties of a standard consumer digital camera as a colorimetric measurement device.

We describe two methods to estimate reflectance spectra from RGB measurements. The first method, called Camera Estimation, first estimates the response properties of

the camera. This camera model is then used to estimate the reflectance spectra of object points. Known techniques will be evaluated, and a variation called MultiSensor will be introduced and investigated. The second method, here named Direct Spectral Estimation, estimates the reflectance spectra directly from the RGB vectors. Since the CIE color difference formulas are widely used, CIELAB values were calculated from the estimated spectra in order to simplify the evaluation and calculate the color difference between reference and reproduced colors.

## 2 Background

One way to use a camera as a colorimeter is to find a model, often some type of regression, which directly maps RGB values to CIELAB values. Examples are described in Hong et. al [1], Wu et. al. [2], MacDonald & Ji [3], Brydges et. al. [4], and Orava et. al. [5]. For printed substrates, this approach has been shown to produce the best results when calibration and measurements are made on samples printed with the same combination of printer and substrate (see for example Hägglund [6]). Moreover, the dependence of the CIELAB values on the selected white-point is another disadvantage which implies that the regression has to be recomputed for new selections of the white-point.

Instead of estimating CIELAB values from camera RGB values one can try to estimate the spectral distribution. Then it is possible to compute the coordinate of the color in different color spaces, such as CIELAB. The availability of the spectral distribution also allows different types of post-processing, for instance simulating the color coordinates under different illuminations. One way to compute such an spectral estimation is to approximate the sensitivity functions of the RGB filters in the camera and use this to estimate the spectral distributions. A tunable monochromatic light source can generate narrow-band light distributions of different wavelengths which can be measured by the camera sensor. The recorded response vectors characterize the transfer functions of the camera filters. Examples of this approach can be found in Wu et. al. [2] and MacDonald & Ji [3]. Here, we will use another common approach where targets containing several colored patches will be acquired with the camera. The recorded camera RGB values will be combined with color values measured by a spectrophotometer to find the spectral characteristics of the camera. This approach avoids the usage of the monochromator and in addition, broadband techniques are often less sensitive to noise than corresponding narrowband approaches.

One common method for finding color filter characteristics is to use a method called Projections Onto Convex Sets (POCS), see for example Sharma [7]. POCS is an estimation technique that combines *a priori* information with the measurements to obtain the estimates. The disadvantage of this method is that it can be strongly dependent on *a priori* knowledge. Without a good estimation, the result can be poor.

Finally, we mention methods using Neural Networks (see Ramanath et. al. [8] as a recent example) that use non-linear techniques to "learn" the desired relations from examples. Neural Networks require a significant amount of time for learning the structure of the data, but once trained, the execution time of the network is comparable to other methods.

### 3 Camera as a Colorimeter

In this section, we describe two methods that are used to estimate the reflectance spectra from the RGB measurements. In the following discussion, we will use two matrices  $\mathbf{R}$  and  $\mathbf{C}$ : in the matrix  $\mathbf{R}$  we collect the vectors with the spectral reflectance for each color patch as measured with a spectrophotometer. The matrix  $\mathbf{C}$  contains the vectors with the corresponding RGB values from the camera. The goal is to find a transformation that maps  $\mathbf{C}$  to  $\mathbf{R}$  and we will describe two approaches. The first method follows the traditional approach which is to invert the camera and estimate the transfer functions. We will mainly follow previous approaches, but also extend with a multi-sensor-type generalization.

The second method can be seen as the inverse of the first approach. Color spectra will be reproduced, but in this case, we treat it as a one-step estimation problem.

#### 3.1 Camera Estimation

The first method requires that the cameras sensitivity functions are estimated first. We describe the camera as a linear system with a matrix  $\mathbf{W}$ , containing three rows, one for each filter function. This gives the following model:

$$\mathbf{C} = \mathbf{RW} \quad (1)$$

The matrices  $\mathbf{C}$  (RGB values) and  $\mathbf{R}$  (spectral reflectance functions) are known and Eq. (1) has to be solved for the matrix  $\mathbf{W}$ . We choose the Moore-Penrose pseudo inverse,  $\text{pinv}(\mathbf{R})$  of  $\mathbf{R}$ , to obtain the following estimation of  $\mathbf{W}$ :

$$\mathbf{W} = \text{pinv}(\mathbf{R}')' \mathbf{C} \quad (2)$$

Since  $\text{pinv}(\mathbf{R}) = (\mathbf{R}'\mathbf{R})^{-1}\mathbf{R}'$  we see that:  $\mathbf{RW} = \mathbf{R} \text{pinv}(\mathbf{R}')' \mathbf{C} = \mathbf{R}((\mathbf{R}'\mathbf{R})^{-1}\mathbf{R}')' \mathbf{C} = \mathbf{R}\mathbf{R}'(\mathbf{R}\mathbf{R}')^{-1}\mathbf{C} = \mathbf{C}$  and in the absence of noise, this is an exact solution. However, under real-world conditions, the result can be difficult to use. This noise sensitivity can be incorporated into the estimation, for example using the Singular Value Decomposition (SVD), Principal Component Analysis (PCA) or Wiener estimation techniques (see Hardeberg [9] for one description of the PCA-technique).

Further improvements in the spectral sensitivity estimation can be gained by introducing additional constraints. Three constraints proposed by Finlayson et. al. [10], later adopted and slightly modified by Barnard and Funt [11] were used in our estimation. The first constraint is related to the Fourier transform. Here a smoothness constraint is introduced by restricting the spectral sensitivity function to linear combinations of a few low-degree Fourier basis functions. Since a device cannot have a negative response to a stimulus, the second constraint is the non-negativity of the functions involved. The third constraint states that the number of peaks in a sensor curve is small. This is referred to as the modality constraint. An alternative to the incorporation of these constraints is a method called Parametric Fitting, proposed by Thomson & Westland in [12]. This method results in similar robust estimates, but the constraints are through the adoption of a low-dimensional parametric model.

In the next step, the estimated camera characteristics are used to estimate the input spectra from measured camera values. The simplest way is to use a pseudo-inverse approach but this turns out to be very noise sensitive. One solution (see Hardeberg

[9]) is to take advantage of *a priori* knowledge about the spectral reflectance functions that are to be constructed. A set of smooth basis functions were constructed. Starting with a large number of different colors, we selected those colors having the most different spectral distributions. This selection procedure is described in Section 4. Singular Value Decomposition is then carried out with these colors to extract only the most significant values. Finally, Fourier based smoothness criteria and positivity constraints are applied to further improve the result.

Inspired by multispectral digital cameras with more than three filters, we found that by using combinations of existing RGB filters, the estimation of spectra from camera values could be improved. The matrix  $\mathbf{C}$  introduced in Equation 1 has the form:

$$\mathbf{C} = [\mathbf{R} \ \mathbf{G} \ \mathbf{B}] \quad (3)$$

We extend  $\mathbf{C}$  by adding the sums of color channel pairs, and obtain a new matrix  $\mathbf{C}'$ :

$$\mathbf{C}' = [\mathbf{R} \ \mathbf{G} \ \mathbf{B} \ \mathbf{R}+\mathbf{G} \ \mathbf{R}+\mathbf{B} \ \mathbf{G}+\mathbf{B}] \quad (4)$$

This variation decreased the maximum error obtained in the evaluation tests.

### 3.2 Direct Spectral Estimation

In the second approach, we ignore the role of the camera filters and directly consider the desired relation between the  $\mathbf{C}$  and  $\mathbf{R}$ , matrices. The model we use can be described as follows: Given the matrices  $\mathbf{C}$  and  $\mathbf{R}$  find the best matrix  $\mathbf{M}$  with

$$\mathbf{R} = \mathbf{CM}' \quad (5)$$

Again,  $\mathbf{R}$  contains spectral reflectance functions,  $\mathbf{C}$  contains the corresponding RGB values, and  $\mathbf{M}$  is a linear operator. Using the pseudo-inverse,  $\mathbf{M}$  can be expressed as:

$$\mathbf{M} = \mathbf{R}' \text{pinv}(\mathbf{C}') \quad (6)$$

where  $\text{pinv}(\mathbf{C}')$  is the Pseudo-Inverse of  $\mathbf{C}'$ . In this method, the illumination is removed from the spectral distribution before the calculations are performed. Once again, a constraint on the Fourier transform can be used to improve the result by smoothening the functions. Here we also introduce an additional constraint. Since the illumination is removed from the spectral distribution, all spectral values in the result must be within the interval from zero to one.

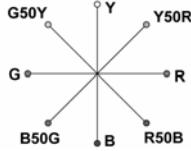
Once the matrix  $\mathbf{M}$  is known, the spectral distributions are calculated by:

$$\mathbf{R} = \mathbf{CM}' \quad (7)$$

Again, we can improve the result with a constraint on the Fourier transform or by forcing the reconstructed spectra to be linear combinations of a spectral basis. We constructed such a basis by first choosing those color spectra that had different spectral distributions. Thereafter Singular Value Decomposition was used to extract the most significant basis vectors.

## 4 Color Patches

Eight pages from NCS Color Atlas 96 [13], selected according to Fig. 1, were used. This resulted in 365 color patches in total. All these colors were used for evaluation. But for characterization, either a small number of these NCS patches were used, or charts printed with inkjet on matte-coated substrate.



**Fig. 1.** The selection of NCS pages used in this study

We followed the proposal by Hardeberg [9] to select the optimal colors to be used in spectral sensitivity characterization. The strategy for the selection of the reflectance samples  $r_{S1}, r_{S2}, r_{S3}, \dots$  which are most significant in the characterization of the camera is as follows: Starting from the full set of all available spectral reflectance functions  $r_p, p = 1 \dots P$ , we first select that  $r_{S1}$  which has maximum RMS value:

$$\|r_{S1}\| \geq \|r_p\| \text{ for } p=1 \dots P \quad (8)$$

Next, we select  $r_{S2}$  which minimizes the condition number of  $[r_{S1} \ r_{S2}]$ , the ratio of the largest to the smallest singular value. Denoting  $w_{\min}(X)$  and  $w_{\max}(X)$  as the minimum and maximum singular values of a matrix  $X$ , this minimization may be expressed by the following expression:

$$\frac{w_{\max}([r_{S1} \ r_{S2}])}{w_{\min}([r_{S1} \ r_{S2}])} \leq \frac{w_{\max}([r_{S1} \ r_p])}{w_{\min}([r_{S1} \ r_p])} \text{ for } p = 1 \dots P, p \neq S_1 \quad (9)$$

Further sample spectra are added according to the same rule:

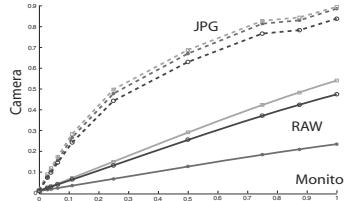
$$\frac{w_{\max}([r_{S1} \ r_{S2} \dots r_{Si}])}{w_{\min}([r_{S1} \ r_{S2} \dots r_{Si}])} \leq \frac{w_{\max}([r_{S1} \ r_{S2} \dots r_{Si-1} \ r_p])}{w_{\min}([r_{S1} \ r_{S2} \dots r_{Si-1} \ r_p])} \text{ for } p=1 \dots P, p \notin \{S_1, S_2, \dots, S_{i-1}\} \quad (10)$$

For each iteration step, a reflectance spectrum that is as different as possible from the already selected spectra is chosen.

## 5 Equipment and Measurements

In our experiments we used a Canon EOS 10D digital camera with a Canon EF 50mm 1:1.8 II lens. The image sensor is a CMOS sensor, with a Red, Green or Blue filter on top of each photodiode. Each captured image is saved in the Canon RAW image format. In an initial test, we measured the response of the camera to inputs of varying intensity. For this purpose, a set of pictures with varying portions of black and white

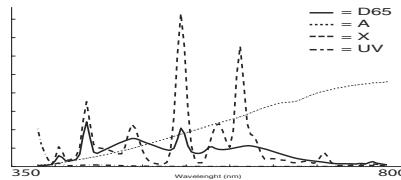
pixels were displayed on a laptop monitor, and photos were captured and saved in both RAW and JPG image format. In Fig. 2, mean intensity values from each type of filter are plotted against the laptop monitor coverage.



**Fig. 2.** Mean intensity values from each color channel (R, G and B) and image format (RAW and JPG) plotted against laptop monitor coverage

The result shows that the JPG responses are not linear to the input optical energy. This is the result of the internal processing of the raw images with the goal to produce “more appealing” pictures. Two common processing steps are white balancing and gamma curve compensations (see Orava et. al. [5] for a discussion of these properties). The RAW image on the other hand, displays output intensities that are almost perfectly linear to the input optical energy. This indicates that using the RAW format is an advantage if the camera is to be used as a measuring instrument.

The illumination was provided by the Minispectra light cabinet from Largo AB. It provides a selection of different light sources but here we only used the Daylight (D65) and light bulb (A). The spectral distributions for these sources are shown in Fig. 3. Notice that the illumination D65 has a different spectral distribution compared to the one defined by CIE. Experiments showed that the sharp peaks in the illumination from Largo could cause problems in the spectral estimation process. This observation is confirmed by many authors, e.g. Romero et. al. [14] and Hardeberg et. al. [15], who have reported that the estimation becomes more difficult if fluorescent illumination is used, especially if it has sharp peaks in the spectral radiance.



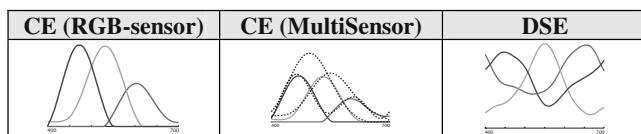
**Fig. 3.** The spectral distributions of the different light sources available in the light cabinet

By using a measurement geometry were the illumination angle is set to  $10^\circ$ , and the measurement angle to  $55^\circ$ , problems with non-uniform illumination and gloss effects were avoided as much as possible. Before images were used in calculations, they were compensated for optical and natural vignetting. The spectrophotometer measurements where carried out with a PR-650 SpectraColorimeter for inkjet charts, and a MacBeth ColorEye 700 for NCS charts. The illumination was measured with a

reflectance standard (SRS-3 from Photo Research) and obtained values were used as white reference in the conversion between CIEXYZ and CIELAB values.

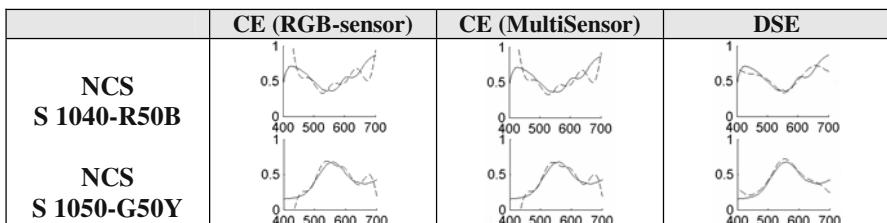
## 6 Results

In the Camera Estimation (CE) method, 22 inkjet colors with varying spectral distributions were used for calibration. The evaluation is based on 365 NCS colors, and illumination A was used for both calibration and evaluation. In the Direct Spectral Estimation (DSE) method, 25 NCS colors with varying spectral distributions were used for calibration, but both illumination A and D65 were used. Hence, 50 calibration colors were used. The evaluations were carried out using 365 NCS colors under illumination A. Fig. 4 shows the resulting filter functions.



**Fig. 4.** Filter functions for original RGB-sensor, MultiSensor, and Direct Spectral Estimation

These functions are then used to reproduce the input spectra from measured camera values. Fig. 5 shows two examples of reproduced spectra.



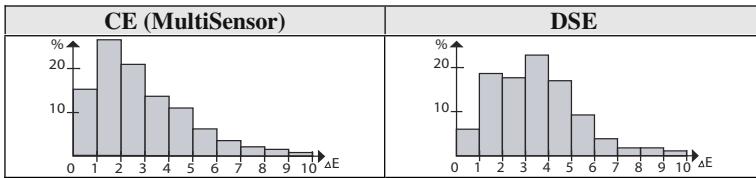
**Fig. 5.** The solid lines represent the original spectra measured with the spectrophotometer and the dashed lines represent the reproduced spectra

In order to simplify the evaluation, CIE Color Matching Functions were used to convert each spectrum to CIEXYZ values. Then CIEXYZ values were converted to CIELAB values and  $\Delta E$ , the difference in CIELAB color space, was calculated between measured and reproduced colors. The result can be seen in Table 1.

**Table 1.** Mean and maximum color difference for the three methods

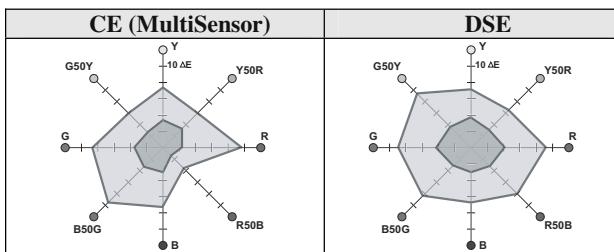
		CE (RGB-sensor)	CE (MultiSensor)	DSE
$\Delta E$	mean	2.84	2.84	3.50
	max	12.30	9.81	9.14

In Fig. 6, the reproduction errors for both the MultiSensor method and the Direct Spectral Estimation method are separated into different  $\Delta E$  intervals.



**Fig. 6.**  $\Delta E$  distributions for MultiSensor and Direct Spectral Estimation

For the MultiSensor method, 61.9 % of the reproduced spectra have an error less than 3  $\Delta E$  units. Only 7.7 % have a color error greater than 6  $\Delta E$ . When Direct Spectral Estimation is used, 42.2 % of the reproduced spectra have an error less than 3  $\Delta E$ , and only 8.7 % have a color error greater than 6  $\Delta E$ . In Fig. 7, the reproduction errors are separated into different locations in the color space, corresponding to different pages in the NCS color system.



**Fig. 7.** Color difference separated into different NCS pages. Inner circle represents mean values, the outer maximum values

For the MultiSensor method, the largest error can be found on the red page, but at the same time the neighboring page, the red and blue, has the smallest error. The conclusion is that the reproduction error varies considerably between different locations in color space. When Direct Spectral Estimation is used, the color differences for both mean and maximum, are almost the same for all pages.

Preliminary experiments with the D65 equivalent illumination shown in Fig. 3 gave the following results: Mean and maximum color difference with Camera Estimation (MultiSensor) was 10.05 and 18.96  $\Delta E$  units. With Direct Spectral Estimation the mean and maximum error was calculated as 4.81 and 13.01  $\Delta E$ .

The results from the color measurement methods presented in this article can be compared to results presented by other authors. Hardeberg [9] calculated a mean  $\Delta E$  value about 6, and a maximum value in the region of 16  $\Delta E$ . Orava et. al. [5] present a mean error greater than 10  $\Delta E$ , but they also mention that with a high-quality camera, the color difference can be as low as 5  $\Delta E$  units. Wu and Allebach [2] achieved a mean color error in the region of 2  $\Delta E$ . All these results were obtained with regression-based methods. Instead, if a spectral estimation technique is used,

Hardeberg's [9] calculated mean and maximum values were almost 7 respectively 17  $\Delta E$  units. In the special context of art reproduction Imai et. al. [16] [17] reported mean and maximum  $\Delta E$  errors for a professional three channel camera about 2 and 6  $\Delta E$  units. They used six channels (filters) in their acquisitions.

## 7 Conclusions

We have investigated how a moderately priced digital camera can be used for color measurements. Two different methods to map the camera RGB values to reflectance spectra were presented. We found that the camera used in this work had a linear sensor response to incoming intensity when the RAW picture format was used. We also found that illumination sources with sharp peaks in the spectral distribution, such as fluorescent lamps should be avoided.

The two methods for the estimation of spectral distributions described in this paper produced similar results. Since reflectance spectra are estimated it is possible to estimate the appearance of the target under different illuminations, a result that cannot be obtained with regression methods. Another advantage is that relatively few colors are required in the calibration procedure. In this case, we used only 22 and 25 for each method respectively. A larger number of calibration colors will not necessarily improve the result. It is important, though, to notice that the calibration colors should be selected carefully. One way to do this is to select a set of colors with spectral distributions that are as different as possible from each other.

The best result with the Camera Estimation method was a mean color difference of 2.84  $\Delta E$ , and a maximum color difference of 9.81  $\Delta E$ . It should be noticed that 61.9 % of the measurements had a  $\Delta E$  below 3, and only 7.7 % of them had a  $\Delta E$  above 6. With the MultiSensor, the maximum errors were significantly reduced. This method also seems to be rather sensitive to fluorescent lamps with sharp peaks in the illumination spectra. The best result achieved with the Direct Spectral Estimation method had a mean color difference of 3.51  $\Delta E$ , and a maximum color difference of 9.15  $\Delta E$ . For this method, 42.2 % of the measurements had a reproduction error below 3  $\Delta E$ , and 8.7 % of them had an error above 6  $\Delta E$ . This is higher than for the Camera Estimation method, but the advantage with this method is that it is less sensitive to peaks in the illumination spectra. With fluorescent lamps, both calibration and evaluation can be made with only a little degeneration compared to a more uniform illumination source using this method.

For both methods the largest measurement errors were obtained for the same type of colors. The best reproduction was found for neutral colors, followed by bright colors. Dark and highly saturated colors were the most difficult to measure.

## Acknowledgments

This article is based on a Master thesis work which was formulated and initiated by MoRe Research in Örnsköldsvik, Sweden and it has been carried out with financial support from the Swedish national research program T2F.

## References

1. Hong, Guowei, et.al, "A Study of Digital Camera Colorimetric Characterization Based on Polynomial Modeling", *Color Research and Applications*, Vol. 26, Nr 1, Feb. 2001, (2001).
2. Wu, Wencheng, et. al, "Imaging Colorimetry Using a Digital Camera", *The Seventh Color Imaging Conference: Color Science, Systems, and Applications*, IS&T, (1999).
3. MacDonald, Lindsay, & Ji, Wei, "Colour Characterisation of a High-Resolution Digital Camera", *Colour in Graphics, Imaging and Vision (CGIV) Conference*, Poitiers, (2002).
4. Brydges, D., et. al, "Application of 3-CCD color camera for colorimetric and densitometric measurements".
5. Orava, Joni, et. al, "Color Errors of Digital Cameras", *Color Research and Applications*, volume 29, Number 3, June 2004, (2004).
6. Hägglund, Åsa, "Colour gamut and colour sensitivity of a desktop scanner", Diploma thesis, Linköping University, Sweden, (2003).
7. Sharma, Gaurav, "Targetless Scanner Calibration", *Journal of Imaging Science and Technology* 44: 301-307, (2000).
8. Ramanath, Rajeev, et. al, "Spectral Spaces and Color Spaces", *Color Research and Applications*, volume 29, Number 1, February 2004, (2004).
9. Hardeberg, Jon Yngve, "Acquisition and Reproduction of Color Images: Colorimetric and Multispectral Approaches", *Ecole National Supérieure des Télécommunications, Département TSI*, France, (1999).
10. Finlayson, Graham, et. al, "Recovering Device Sensitivities with Quadratic Programming", *The Sixth Color Imaging Conference, Systems, and Applications*, IS&T, (1998).
11. Barnard, Kobus, & Funt, Brian, "Camera Characterization for Color Research", *Color Research and Applications*, volume 27, Number 3, June 2002, (2002).
12. Thomson, M., & Westland, S., "Colour-Imager Characterization by Parametric Fitting of Sensor Responses", *Color Research and Applications*, Vol. 26, Nr 6, Dec. 2001, (2001).
13. Natural Color System (NCS), Scandinavian Colour Institute, <http://www.ncs.se>
14. Romero, Javier, et. al, "Color-signal filtering in the Fourier-frequency domain", *J. Opt. Soc. Am. A*/Vol. 20, No. 9/September 2003, (2003).
15. Hardeberg, Jon Yngve, et. al, "Spectral characterization of electronic cameras", *École Nationale Supérieure des Télécommunications*, Paris, France.
16. Imai, Francisco H., Berns, Roy S., & Tzeng, Di-Y, "A Comparative Analysis of Spectral Reflectance Estimated in Various Spaces Using a Trichromatic Camera System", *Journal of imaging and technology*, Volume 44, Number 4, July/August 2000, (2000).
17. Imai, Francisco H., & Berns, Roy S., "Spectral estimation of artist oil paints using multi-filter trichromatic imaging", *Proc. of the 9<sup>th</sup> Congress of the International Colour Association*, Rochester, NY, 2001, pp. 504-507, (2001).

# Image Analysis with Local Binary Patterns

Matti Pietikäinen

Machine Vision Group,

Infotech Oulu and Department of Electrical and Information Engineering,

P.O. Box 4500, FI-90014 University of Oulu, Finland

mkp@ee.oulu.fi, <http://www.ee.oulu.fi/mvg>

**Abstract.** The local binary pattern approach has evolved to represent a significant breakthrough in texture analysis, outperforming earlier methods in many applications. Perhaps the most important property of the LBP operator in real-world applications is its tolerance against illumination changes. Another equally important is its computational simplicity, which makes it possible to analyze images in challenging real-time settings. Recently, we have begun to study image analysis tasks which have not been generally considered texture analysis problems. Our excellent results suggest that texture and the ideas behind the LBP methodology could have a much wider role in image analysis and computer vision than was thought before.

## 1 Introduction

Image texture analysis is an important fundamental problem in computer vision. During the past few years, we have developed theoretically and computationally simple, but very efficient nonparametric methodology for texture analysis based on Local Binary Patterns (LBP). The LBP texture analysis operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. For each pixel in an image, a binary code is produced by thresholding its value with the value of the center pixel. A histogram is created to collect up the occurrences of different binary patterns. The basic version of the LBP operator considers only the eight neighbors of a pixel, but the definition has been extended to include all circular neighborhoods with any number of pixels [1, 2, 3].

Through its extensions, the LBP operator has been made into a really powerful measure of image texture, showing excellent results in terms of accuracy and computational complexity in many empirical studies. The LBP operator can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its tolerance against illumination changes. Another equally important is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.

The LBP method has already been used in a large number of applications all over the world, including visual inspection, image retrieval, remote sensing, biomedical image analysis, face image analysis, motion analysis, environment modeling, and outdoor scene analysis. For a bibliography of LBP-related research, see [4].

Recently, we have begun to study machine vision tasks which have not been previously considered texture analysis problems. The LBP methodology has been adapted to outdoor scene classification, face recognition, face detection, facial expression recognition and content based image retrieval with excellent success. We have also developed the first texture-based method for subtracting the background and detecting moving objects in real time. A short description of this recent progress is presented in the following section.

## 2 Recent Progress

In [5], we proposed a new method for view-based recognition of 3D textured surfaces using multiple LBP histograms as texture models. Our method provided the leading performance for the Columbia-Utrecht database (CUReT) textures taken in different views and illuminations. The method also performed well in pixel-based classification of outdoor scene textures. These images had wide variability within and between images due to changes in illumination, shadows, foreshortening, self-occlusion, and non-homogeneity of the texture classes.

Finding proper features and representative training samples can be very problematic in this kind of problems. Earlier we proposed a visualization based approach for training a texture classifier, in which LBP features are used for texture description and a self-organizing map (SOM) is employed for visual training and classification [6]. Recently, we developed a more comprehensive framework for texture image labeling [7]. Textures are modeled with complementary measures including various versions of the LBP and Gabor features. Combined use of active learning, co-training, and visualization based learning is applied to feature data, enabling comprehensive, accurate, and user friendly texture labeling.

A novel approach to face recognition was developed which considers both shape and texture information to represent face images [8]. The face area is first divided into several small regions from which the LBP features are extracted and concatenated into an enhanced feature vector to be used as a face descriptor. In extensive experiments using FERET test images and protocol, considering variations in facial expressions, lighting and aging of the subjects, our methodology outperformed the state-of-the-art methods. However, it was unclear whether the high performance was due to the use of local regions (instead of an holistic approach) or to the discriminative power of LBP. Experimental results with four different texture descriptors clearly showed the superiority of the LBP based approach [9].

A compact LBP based descriptor was also developed for face detection and for the recognition of low-resolution face images [10]. Considering the derived feature space, a second-degree polynomial kernel SVM classifier was trained to detect frontal faces in gray scale images. Experimental results using several complex images showed that our approach performs favorably compared to the state-of-the-art. Additionally, experiments with detecting and recognizing low-resolution faces from video sequences were carried out, demonstrating that the same facial representation can be efficiently used for both detection and recognition.

Two approaches to facial expression recognition from static images were developed using LBP histograms computed over non-overlapping blocks for face

description. In the first method, the Linear Programming technique is adopted to classify seven facial expressions (anger, disgust, fear, happiness, sadness, surprise and neutral) [11]. In another approach, a coarse-to-fine classification scheme was used [12]. Good results were obtained for the Japanese Female Facial Expression (JAFFE) database used in the experiments.

Approaches to using LBP in content-based image retrieval were also studied [13]. Block based methods dividing the query and database images (or database images only) into blocks and comparing their LBP histograms were found to perform significantly better than methods based on global LBP histograms. The results for the block based LBP approach were also better than those obtained with the widely used color correlogram features. Image databases taken from the Corel Image Gallery and from a German stamp database were used in experiments.

A novel texture-based method for modeling the background and detecting moving objects from video sequences was developed [14, 15]. Each pixel is modeled as a group of adaptive local binary pattern histograms that are calculated over a circular region around the pixel. The method was evaluated against several video sequences including both indoor and outdoor scenes. It was shown to be tolerant to illumination variations, the multimodality of the background, and the introduction or removal of background objects. Furthermore, the method is capable for real-time processing.

### 3 Conclusions

The local binary pattern approach has evolved to represent a significant breakthrough in texture analysis, outperforming earlier methods in many applications. Recently, we have begun to study image analysis tasks which have not been generally considered texture analysis problems. Our excellent results suggest that texture and the ideas behind the LBP methodology could have a much wider role in image analysis and computer vision than was thought before. Our future plan is to explore this thoroughly.

### Acknowledgments

I wish to thank Timo Ahonen, Abdenour Hadid, Marko Heikkilä, Topi Mäenpää, Timo Ojala, Valtteri Takala, Markus Turtinen, and Feng Xiaoyi for their contributions. The financial support of the Academy of Finland is gratefully acknowledged.

### References

1. Ojala, T., Pietikäinen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition* **29** (1996) 51-59
2. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 971 - 987

3. Mäenpää, T., Pietikäinen, M.: Texture Analysis with Local Binary Patterns. In: Chen, C.H., Wang, P.S.P. (eds.): *Handbook of Pattern Recognition and Computer Vision*, 3<sup>rd</sup> edn. World Scientific (2005) 197-216
4. <http://www.ee.oulu.fi/research/imag/texture/>
5. Pietikäinen, M., Nurmela, T., Mäenpää, T., Turtinen, M.: View-Based Recognition of Real-World Textures. *Pattern Recognition* **37** (2004) 313-323
6. Turtinen, M., Pietikäinen, M.: Visual Training and Classification of Textured Scene Images. In: *The 3rd International Workshop on Texture Analysis and Synthesis* (2003) 101-106
7. Turtinen, M., Pietikäinen, M.: Labeling of Textured Data with Co-Training and Active Learning. In review.
8. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Recognition with Local Binary Patterns. In: *Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science* 3021 (2004) 469-481
9. Ahonen, T., Pietikäinen, M., Hadid, A., Mäenpää, T.: Face Recognition Based on the Appearance of Local Regions. In: *17th International Conference on Pattern Recognition* (2004) III: 153-156
10. Hadid, A., Pietikäinen, M., Ahonen, T.: A Discriminative Feature Space for Detecting and Recognizing Faces. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2004) II: 797-804
11. Feng, X., Pietikäinen, M., Hadid, A.: Facial Expression Recognition with Local Binary Patterns and Linear Programming. *Pattern Recognition and Image Analysis* **15** (2005) 550-552
12. Feng, X., Hadid, A., Pietikäinen, M.: A Coarse-to-Fine Classification Scheme for Facial Expression Recognition. In: *Image Analysis and Recognition, ICIAR 2004 Proceedings, Lecture Notes in Computer Science* 3212 (2004) II: 668-675
13. Takala, V., Ahonen, T., Pietikäinen, M.: Block-Based Methods for Image Retrieval Using Local Binary Patterns. In: *Image Analysis, SCIA 2005 Proceedings, Lecture Notes in Computer Science* 3540 (2005)
14. Heikkilä, M., Pietikäinen, M., Heikkilä, J.: A Texture-Based Method for Detecting Moving Objects. In: *The 15th British Machine Vision Conference* (2004) I: 187-196
15. Heikkilä, M., Pietikäinen, M.: A Texture-Based Method for Modeling the Background and Detecting Moving Objects. In review.

# Object Evidence Extraction Using Simple Gabor Features and Statistical Ranking\*

J.-K. Kamarainen<sup>1</sup>, J. Ilonen<sup>1</sup>, P. Paalanen<sup>1</sup>, M. Hamouz<sup>2</sup>, H. Kälviäinen<sup>1</sup>,  
and J. Kittler<sup>2</sup>

<sup>1</sup> Dept. of Information Technology,  
Lappeenranta University of Technology, Finland  
<sup>2</sup> University of Surrey, United Kingdom

**Abstract.** Several novel methods based on locally extracted object features and spatial constellation models have recently been introduced for invariant object detection and recognition. The accuracy and reliability of the methods depend on the success of both tasks: evidence extraction and spatial constellation model search. In this study an accurate and efficient method for evidence extraction is introduced. The proposed method is based on simple Gabor features and their statistical ranking.

## 1 Introduction

By object evidence extraction we refer to the detection of local descriptors and salient sub-parts of objects. This approach can recover from object occlusion in a natural way; occlusion prevents the detection of all features, but the detection can still be based on a sub-set of features. Thus, it seems that the approach is a good candidate for general object detection and recognition. The idea of partitioning an object into smaller pieces which together represent the complete object is not new (e.g. [1]), but existing implementations have lacked sufficient accuracy and reliability until recently.

In 2D object detection and recognition local object feature detectors must perform reliably in a rotation, scale, and translation invariant manner. For real applications they should also exhibit sufficient robustness against noise and distortions. The problem of extracting local descriptors can be divided into two categories: 1) unsupervised and 2) supervised. The unsupervised approach is more challenging since it must first solve a more general problem of what is really “important” in images - the question which intrigues brain and cognitive science researchers as well. In the literature, several unsupervised descriptors have been proposed, e.g., combined corner and edge detectors by Harris and Stephens [2], but only very recently more representative and theoretically sound methods such as salient scale descriptors by Kadir [3] and SIFT (scale invariant feature transform) features by Lowe [4] have been introduced. The major advantage of unsupervised local descriptors is the unsupervised nature itself and the

---

\* Academy of Finland (#204708) and EU (#70056/04) are acknowledged for support.

main disadvantage is the disability to exclusively label the findings; an object is described by a spatially connected distribution of non-unique labels. However, unsupervised descriptors may provide information about position, scale, and rotation, and thus, object detection can be based on an inspection of both the configuration and the properties of extracted evidence making the detection more efficient (see, e.g., [5]).

Unsupervised descriptors have recently been a more popular topic, but this study promotes the supervised approach. It seems improbable that either of the two approaches would have an overall superiority since they possess distinct advantages and disadvantages and enable different approaches in upper processing layers. Supervised detection of local descriptors is based on a detection scheme where important image sub-parts (evidence), are known in advance, and thus, detectors can be optimized. It is clear that since a supervised detector is more specific it can be made more reliable and accurate, but a new problem is how to select which image parts to use. The supervised descriptor detection (evidence extraction) is a similar problem to object detection itself, but an explicit assumption is made that local image patches are less complex than a complete object. Consequently simpler feature detection methods can be applied. Furthermore, since supervised descriptors are more reliable and accurate than unsupervised, simpler spatial models can be used to detect objects - a single detected evidence creates already a hypothesis that an object is situated in that location (see, e.g., [6]). Respectively, in the unsupervised descriptors based detection the number of descriptors required is often large. Several occurrences of descriptors in the vicinity of a correct spatial configuration compensates the low reliability of detecting a single descriptor. The selection of image sub-parts in the supervised detection is an application specific task, but it can also be automated if evidence items which are most typical for specific objects are selected; the theory of unsupervised detection can be utilized.

In this study a novel supervised evidence extraction method is introduced. The method is based on simple Gabor features introduced by the authors [7] and statistical ranking using Gaussian mixture model probability densities proposed by the authors in [8]. The method has been successfully applied in face localization [6]. This study describes the approach in more detail, introduces accompanying theory and algorithms and presents the latest experimental results.

## 2 Simple Gabor Features

The simple Gabor feature space and its properties have been introduced in [7]. Here the properties are explained more carefully in order to demonstrate the practical use.

### 2.1 Structure of Simple Gabor Feature Space

The phrase “simple” in the context of simple Gabor feature space refers to a fact that the feature space considers phenomena, here evidence, at a single

spatial location. A single spatial location does not straightforwardly correspond to a single pixel in digital images since effective area, envelope, of Gabor filter stretches over a substantially larger area; yet the reconstruction accuracy is highest near the centroid. It is clear that complex objects cannot be represented by a simple Gabor feature which is concentrated near a single location but a spatial (constellation) model must be built upon the features and combine them (see, e.g., [6]).

The main idea in simple Gabor feature space is to utilize a response of Gabor filter  $\psi(x, y; f, \theta)$  at a single location  $(x, y) = (x_0, y_0)$  of image  $\xi(x, y)$

$$r_\xi(x, y; f, \theta) = \psi(x, y; f, \theta) * \xi(x, y) = \iint_{-\infty}^{\infty} \psi(x - x_\tau, y - y_\tau; f, \theta) \xi(x_\tau, y_\tau) dx_\tau dy_\tau \quad (1)$$

The response is calculated for several frequencies  $f_k$  and orientations  $\theta_l$ .

The frequency corresponds to scale which is not an isotropic variable, the spacing of frequencies must be exponential [7]

$$f_k = c^{-k} f_{max}, \quad k = \{0, \dots, m - 1\} \quad (2)$$

where  $f_k$  is the  $k$ th frequency,  $f_0 = f_{max}$  is the highest frequency desired, and  $c$  is the frequency scaling factor ( $c > 1$ ).

The rotation operation is isotropic, and thus, it is necessary to position filters in different orientations uniformly [7]

$$\theta_l = \frac{l2\pi}{n}, \quad l = \{0, \dots, n - 1\} \quad (3)$$

where  $\theta_l$  is the  $l$ th orientation and  $n$  is the number of orientations to be used. The computation can be reduced to half since responses on angles  $[\pi, 2\pi[$  are complex conjugates of responses on  $[0, \pi[$  for real valued signals.

**Feature Matrix.** The Gabor filter responses can be now arranged into a matrix form as

$$\mathbf{G} = \begin{pmatrix} r(x_0, y_0; f_0, \theta_0) & r(x_0, y_0; f_0, \theta_1) & \cdots & r(x_0, y_0; f_0, \theta_{n-1}) \\ r(x_0, y_0; f_1, \theta_0) & r(x_0, y_0; f_1, \theta_1) & \cdots & r(x_0, y_0; f_1, \theta_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0, y_0; f_{m-1}, \theta_0) & r(x_0, y_0; f_{m-1}, \theta_1) & \cdots & r(x_0, y_0; f_{m-1}, \theta_{n-1}) \end{pmatrix} \quad (4)$$

where rows correspond to responses on the same frequency and columns correspond to responses on the same orientation. The first row is the highest frequency and the first column is typically the angle  $0^\circ$ .

## 2.2 Feature Matrix Manipulation for Invariant Search

From the responses in the feature matrix in Eq. (4) the original signal  $\xi(x, y)$  can be approximately reconstructed near the spatial location  $(x_0, y_0)$ . It is thus possible to represent and consequently also recognize evidence using the Gabor feature matrix.

The additional property which makes simple Gabor features useful is the fact that linear row-wise and column-wise shifts of the response matrix correspond to scaling and rotation in the input space. Thus, invariant search can be performed by simple shift operations, by searching several spatial locations (spatial shift) and by shifting response matrices.

Rotating an input signal  $\xi(x, y)$  anti-clockwise by  $\frac{\pi}{n}$  equals to the following shift of the feature matrix

$$\mathbf{G} = \begin{pmatrix} r(x_0, y_0; f_0, \theta_{n-1})^* & r(x_0, y_0; f_0, \theta_0) & \Rightarrow & r(x_0, y_0; f_0, \theta_{n-2}) \\ r(x_0, y_0; f_1, \theta_{n-1})^* & r(x_0, y_0; f_1, \theta_0) & \Rightarrow & r(x_0, y_0; f_1, \theta_{n-2}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0, y_0; f_{m-1}, \theta_{n-1})^* & r(x_0, y_0; f_{m-1}, \theta_0) & \Rightarrow & r(x_0, y_0; f_{m-1}, \theta_{n-2}) \end{pmatrix} \quad (5)$$

where \* denotes complex conjugate.

Downscaling the same signal by a factor  $\frac{1}{c}$  equals to the following shift of the feature matrix

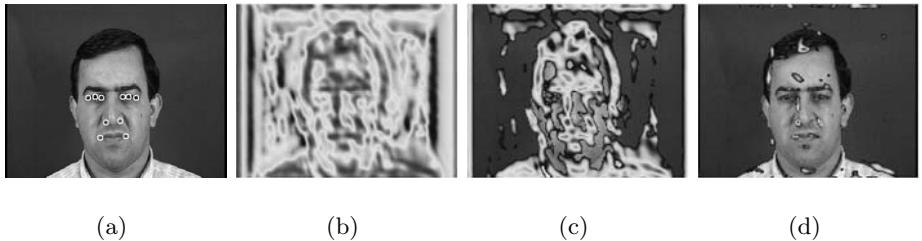
$$\mathbf{G} = \begin{pmatrix} r(x_0, y_0; f_1, \theta_0) & r(x_0, y_0; f_1, \theta_1) & \cdots & r(x_0, y_0; f_1, \theta_{n-1}) \\ r(x_0, y_0; f_2, \theta_0) & r(x_0, y_0; f_2, \theta_1) & \cdots & r(x_0, y_0; f_2, \theta_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_0, y_0; f_m, \theta_0) & r(x_0, y_0; f_m, \theta_1) & \cdots & r(x_0, y_0; f_m, \theta_{n-1}) \end{pmatrix} \quad (6)$$

It should be noted that responses on new low frequencies  $f_m$  must be computed and stored in advance while the highest frequency responses on  $f_0$  vanish in the shift.

### 3 Statistical Classification and Ranking of Features

In general, any classifier or pattern recognition method can be used to train and to classify features into evidence classes. However, certain advantages advocate the use of statistical methods. Most importantly, not only class labels for observed features are desired but also it should be possible to rank evidence items in a scene and to sort them in the best matching order for returning only a fixed number of the best candidates. The ranking reduces search space of a spatial model (e.g., [9]), and furthermore, rank values can be integrated into a statistical spatial model as well. Ranking requires a measure for confidence, that is, a quantitative measure which represents the reliability of classification into a certain class. It is possible to introduce ad hoc confidence measures for the most classifiers, but statistical measures, such the value of the class-conditional probability density function (pdf) are more sound [8].

In order to apply statistical classification and ranking it is necessary to estimate class conditional pdf's for every evidence. Since Gabor filters are Gaussian shaped in both spatial and frequency domains they typically enforce observations into a form of Gaussian distribution in the feature space [10]. However, a single Gaussian cannot represent class categories, such as eyes, since they



**Fig. 1.** Example of using density quantile and pdf values as confidence : (a) Face image and 10 evidence classes; (b) Pdf surface for the left nostril (left in image); (c) Pdf values belonging to 0.5 density quantile; (d) Pdf values belonging to 0.05 density quantile

may contain inherited sub-classes, such as closed eye, open eye, Caucasian eye, Asian eye, eye with eye glasses, and so on. Inside a category there are instances from several sub-classes which can be distinct in the feature space. In this sense Gaussian mixture model is a more effective principal distribution to represent the statistical behavior of simple Gabor features.

There are several methods to estimate parameters of Gaussian mixture models (GMM's) and for example the unsupervised method by Figueiredo and Jain [11] seems to be an accurate and efficient method [8]. The Figueiredo-Jain algorithm is unsupervised in the sense that it automatically estimates the number of components in a GMM. The original method can be extended to complex vectors constructed from the Gabor feature matrices in (4) as [8]

$$\mathbf{g} = [r(x_0, y_0; f_0, \theta_0) \ r(x_0, y_0; f_0, \theta_1) \dots r(x_0, y_0; f_{m-1}, \theta_{n-1})] . \quad (7)$$

Using estimated pdfs it is possible to assign a class for features extracted at any location of an image by simply applying the Bayes decision making. However, as posteriors do not act as inter-class measures but as between-class measures for a single observation, class-conditional probability (likelihood) is a preferred choice to act as a ranking confidence score [8]. It is a measure of how reliable the class assignment of the evidence is. Now, evidence with the highest confidence can be delivered for consistency analysis first. The use of confidence values may reduce search space by an arbitrary degree by discarding evidence beyond a requested density quantile [8]. In Fig. 1 the use of density quantile for reducing the search space is demonstrated; it is clear that the correct evidence is already within 0.05 (0.95 confidence) density quantile.

## 4 Evidence Extraction

By combining simple Gabor features in Section 2 and statistical classification and ranking in Section 3 a novel evidence extraction method can be devised. Next, Algorithms 1 and 2, one for estimating evidence specific pdfs using the training set images and the other for extracting evidence, are introduced on a general level and discussed in detail.

**Algorithm 1** *Train evidence classifier*

```

1: for all Training images do
2:   Align and normalize image to represent an object in a standard pose
3:   Extract simple Gabor features at evidence locations
4:   Normalize simple Gabor features
5:   Store evidence features  $P$  and their labels  $T$ 
6: end for
7: Estimate GMM pdf for each evidence with data in  $P$ 

```

In Algorithm 1 the fundamental steps to generate a pdf-based classifier for evidence extraction are shown. First, training images must be aligned to a standard pose. The standard pose corresponds to a pose where objects have roughly the same scale and orientation. In the supervised evidence extraction the normalization and aligning is possible since keypoint locations are known. In the standard pose, simple Gabor features in (4) are then computed at evidence locations. Feature matrices can be energy-normalized if a complete illumination invariance is required. Each feature matrix is reformatted into a vector form in (7) and stored in a sample matrix  $P$  along with corresponding labels,  $T$ . Finally, complex pdfs are estimated for each evidence separately, e.g., utilizing GMM and the FJ algorithm.

**Algorithm 2** *Extract  $K$  best evidences of each type from an image  $I$* 

```

1: Normalize image
2: Extract simple Gabor features  $G(x, y; f_m, \theta_n)$  from image  $I(x, y)$ 
3: for all Scale shifts do
4:   for all Rotation shifts do
5:     Shift Gabor features
6:     Normalize Gabor features
7:     Calculate confidence values for all classes and for all  $(x, y)$ 
8:     Update evidence confidence at each location
9:   end for
10: end for
11: Sort evidences for each class
12: Return  $K$  best evidences for every evidence class

```

In Algorithm 2 the main steps to extract evidence from an image are shown. First, the image is normalized, that is, scale and grey levels are adjusted to correspond to average object presence used in the training. From a normalized image simple Gabor features are extracted at every spatial location and confidence values are computed for all requested invariance shifts. If features were energy normalized in the training phase the same normalization must be applied before calculating confidence values from GMM pdfs. In a less memory requiring implementation, confidence values can be iteratively updated after each shift in order to store only the best candidates of each evidence at each location. After the shifts have been inspected it is straightforward to sort them and return the best candidates. In this approach one location may represent more than one evidence, but each evidence can be in one pose only.

## 5 Experiments

In this section we present the results of an application of the algorithm to a practical problem of detecting facial evidence in images from XM2VTS database.

### 5.1 XM2VTS Database

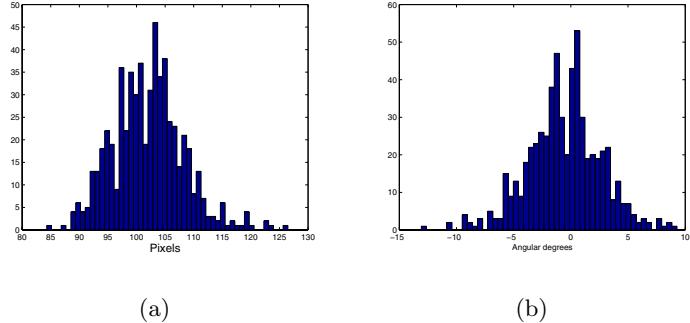
XM2VTS facial image database is a publicly available database for benchmarking face detection and recognition methods [12]. The frontal part of the database contains 600 training images and 560 test images of size  $720 \times 576$  (width  $\times$  height) pixels. Images are of excellent quality and any face detection method should perform well with the database.

To train the evidence detectors a set of salient face regions must be selected first. The regions should be stable over all objects from the same category, but also discriminative comparing to other object regions and backgrounds. For facial images ten specific regions (see Fig. 3(a)) have been shown to contain favourable properties to act as evidence [9].

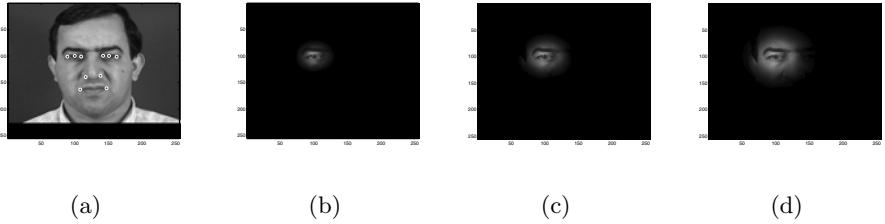
**Selecting Simple Gabor Feature Parameters.** The first problem in the parameter selection is the number of frequencies,  $m$ , and orientations,  $n$ , to be used in feature matrix in (4). Many factors contribute to the final performance, but generally the more frequencies and orientations are used, the better is the representation power of the simple Gabor feature. By increasing the numbers, shift sensitivity increases as well, allowing a more accurate determination of evidence pose. Generally, sharpness values of the filter, which also affect to the representation power, can be set to  $\gamma = \eta = 1.0$  and a good initial number of filters are four orientations  $n = 4$  on three frequencies  $m = 3$  making the feature matrix of size  $3 \times 4$ . The effect of changing parameter values can be later evaluated experimentally.

Using only 4 orientations affects the angular discrimination to be  $45^\circ$ , which is much broader than the rotations in the XM2VTS training set (Fig. 2(b)). The selection of frequencies is a more vital question. First of all in Fig. 2(a) it can be seen that in the XM2VTS database the mean distance between eyes is 102 pixels and the distribution is approximately normal. Thus, for optimal accuracy, training images should be normalized to the eye center distance of 102 pixels. Alternatively for recognizing also the smallest faces the training distance can be normalized to 84 pixel distance and the frequency factor  $c$  set to  $\frac{102}{84} \approx 1.2$  in order to have exactly the mean value for the first scale shift. Second, shift would correspond to the eye distance 122 which is near the maximal value of eye center distances (126) and now the whole interval is covered. The interval can be sub-divided further, but this increases the computational complexity and does not infinitely increase the accuracy due to the scale sensitivity.

Setting the frequency factor to 1.2 would be optimal, but it would be a very small value causing a significant overlap of Gabor filters. The amount of overlap can be controlled by adjusting the filter sharpness,  $\gamma$  and  $\eta$ , but still, the smaller the frequency factor is, the more frequencies are needed to cover a broad



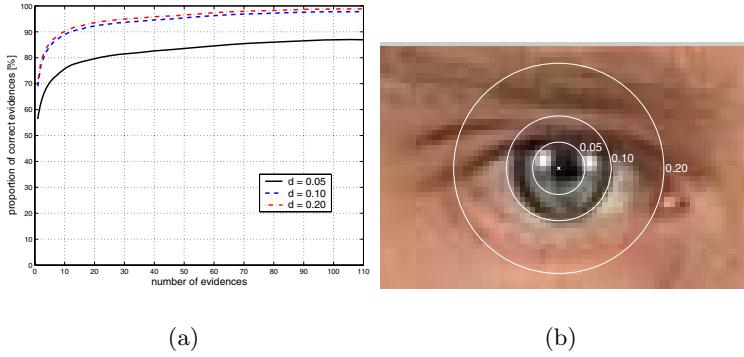
**Fig. 2.** Scale and orientation contents of XM2VTS training data computed using coordinates of left and right eye centers: a) Distribution of eye center distances (min. 84 pix, max. 126 pix, mean 102 pix); b) Distribution of eye center rotation angles (abs. min.  $0^\circ$ , abs. max.  $13.0^\circ$ , abs. mean  $2.5^\circ$ , mean  $-0.5^\circ$ )



**Fig. 3.** Normalized facial image and effective areas of Gabor filters on different frequencies: (a) 10 salient evidences (left and right outer eye corners, left and right inner eye corners, left and right eye centers, left and right nostrils, and left and right mouth corners); (b)  $f_0 = \frac{1}{1.15}$ , (c)  $f_1 = \frac{1}{\sqrt{2} \cdot 1.15}$ , (d)  $f_2 = \frac{1}{2 \cdot 1.15}$

frequency range and to represent objects accurately. In the case of XM2VTS database the whole scale variation can be covered without any scale shifts and by just selecting filters that can efficiently represent the various evidence. Thus, the frequency factor  $c$  was set to  $\sqrt{2}$ . In Fig. 3 an example of aligned image for extracting Gabor features is shown. The distance of the eye centers is normalized to 51 which is half of the mean value, and thus, test images can be processed in a half scale for faster computation. Furthermore, the angle between the eye centers is rotated to  $0^\circ$ , which roughly corresponds to the expectation. Images are cropped to the size of  $256 \times 256$ . In Fig. 3 effective areas of selected filters are also shown and it can be seen that they extract information on several different scales providing distinct information. With the given heuristics it can be assumed that the represented parameter values could perform well for the XM2VTS.

Furthermore, it seems that the simple Gabor features form smooth probability distributions for facial evidences, and thus, the methods for estimating



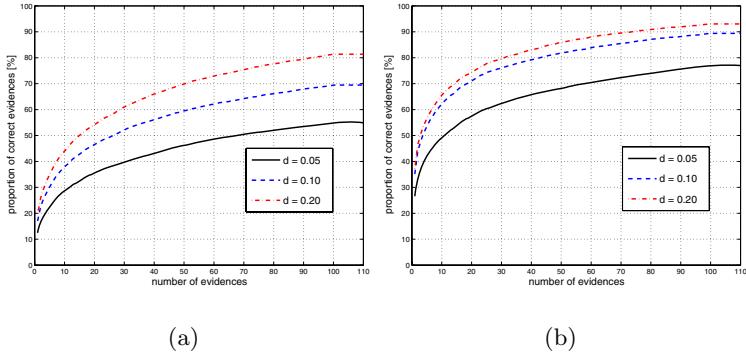
**Fig. 4.** Results for evidence extraction from XM2VTS test images: (a) Accuracy; (b) Demonstration of accuracy distance measure

parameters of pdf's perform accurately and robustly converging to the same estimates repeatedly with random initializations.

**Results for Original Images.** Evidence items were extracted in a ranked order and an evidence item was considered to be correctly extracted if it was within a pre-set distance limit from a correct location. In Fig. 4(a) are shown the accuracies for three different distance limits. The distances are scale normalized, so that the distance between the centers of the eyes is 1.0 (see Fig. 4(b)). From the figure it can be seen that all evidence cannot be extracted within the distance of 0.05, but on average 8 items of correct evidence are already included in the first ten items of evidence (one from each class) and by increasing the number to 100, only a small improvement can be achieved. However, within the distance 0.10 nine items of correct evidence were included already in the first ten items of evidence from each class and by extracting 100 items of evidence almost perfect detection rate was achieved. It should be noted that for constellation model methods it is possible to handle several thousands items of evidence (e.g. [9]).

**Results for Artificially Rotated and Scaled Images.** The main problem with XM2VTS data set was that faces did not comprehensively cover different scales and rotations (see Fig. 2), and thus, invariance of evidence extraction cannot be reliably verified. In the second experiment the same images were used, but they were artificially scaled by a uniform random factor between  $[1, \sqrt{2}]$ , which corresponds to the scale factor  $c$ , and rotated by  $[-45^\circ, 45^\circ]$  where  $45^\circ$  corresponds to the angle between two adjacent filters. In Fig. 5 the results for an experiment where no invariance shifts were applied and for another experiment where shifts were applied are shown. It is clear that the shifts provided more invariance for the extraction since at the error  $d = 0.05$  the accuracy increased from 45% to almost 70% when the total of 50 items of evidence were fetched.

A significant increase in the accuracy was achieved by adding only single shifts of features, but it is not necessary to tie shifts to the configuration of simple



**Fig. 5.** Results for evidence extraction from artificially rotated and scaled XM2VTS test images: (a) No shifts; (b)  $\{0, 1\}$  scale shifts and  $\{-1, 0, 1\}$  rotation shifts applied

Gabor features in the training. In the extraction, the spacing can be tighter, e.g., orientations by  $45^\circ/2 = 22.5\%$  and scales by  $\sqrt{\sqrt{2}} = \sqrt[4]{2}$  to establish a double density. With the double density only every second feature in the feature matrix is used in the classification, but the invariance is further increased.

## 6 Conclusions

In this study, evidence based object detection was studied. We have argued that it is an accurate and reusable approach to general object detection. In the pursuance of this approach, a complete method and algorithms for invariant evidence extraction have been proposed. The proposed method is supervised by its nature and is based on simple Gabor features and statistical ranking. The analytical results were verified by experiments using real data of facial images. The method has been proved to be sufficiently accurate and reliable in practice and the future research will focus on developing a spatial model which can optimally utilize the provided evidence.

## References

1. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. *IEEE Trans. on Computers* **22** (1973) 67–92
2. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of the Fourth Alvey Vision Conf. (1988) 147–151
3. Kadir, T.: Scale, Saliency and Scene Description. PhD thesis, Oxford University (2002)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision* **60** (2004) 91–110
5. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. (2003)

6. Hamouz, M., Kittler, J., Kamarainen, J.K., Paalanen, P., Kälviäinen, H.: Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model. In: Proc. of the 6th Int. Conference on Automatic Face and Gesture Recognition, Seoul, Korea (2004) 67–72
7. Kyrki, V., Kamarainen, J.K., Kälviäinen, H.: Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters* **25** (2003) 311–318
8. Paalanen, P., Kamarainen, J.K., Ilonen, J., Kälviäinen, H.: Feature representation and discrimination based on Gaussian mixture model probability densities – practices and algorithms. Research report 95, Department of Information Technology, Lappeenranta University of Technology (2005)
9. Hamouz, M.: Feature-based affine-invariant detection and localization of faces. PhD thesis, University of Surrey (2004)
10. Kämäräinen, J.K.: Feature Extraction Using Gabor Filters. PhD thesis, Lappeenranta University of Technology (2003)
11. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 381–396
12. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: XM2VTSDB: The extended M2VTS Database. In: Proc. of the 2nd Int. Conf. on Audio and Video-based Biometric Person Authentication. (1999) 72–77

# Dynamically Visual Learning for People Identification with Sparsely Distributed Cameras

Hidenori Tanaka<sup>1,2</sup>, Itaru Kitahara<sup>1</sup>, Hideo Saito<sup>2</sup>, Hiroshi Murase<sup>3</sup>,  
Kiyoshi Kogure<sup>1</sup>, and Norihiro Hagita<sup>1</sup>

<sup>1</sup> Intelligent Robotics and Communication Laboratories, ATR,  
2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan

{hidenori, kitahara, kogure, hagita}@atr.jp

<sup>2</sup> Graduate School of Science and Technology, Keio University,  
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, Japan  
saito@ozawa.ics.keio.ac.jp

<sup>3</sup> Graduate School of Information Science, Nagoya University,  
Furo-cho, Chikusa-ku, Nagoya, Japan  
murase@is.nagoya-u.ac.jp

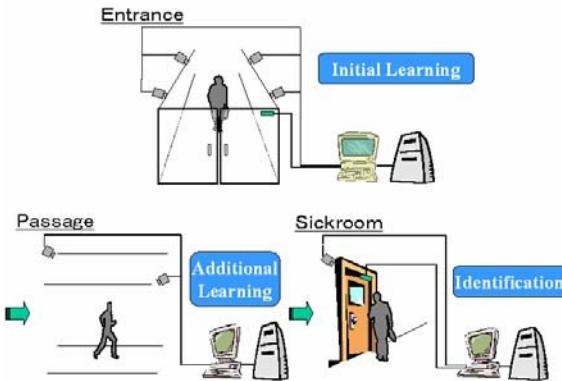
**Abstract.** We propose a dynamic visual learning method that aims to identify people by using sparsely distributed multiple surveillance cameras. In the proposed method, virtual viewpoint images are synthesized by interpolating the sparsely distributed images with a simple 3D shape model of the human head, so that virtual densely distributed multiple images can be obtained. The multiple images generate an initial eigenspace in the initial learning step. In the following additional learning step, other distributed cameras capture additional images that update the eigenspace to improve the recognition performance. The discernment capability for personal identification of the proposed method is demonstrated experimentally.

## 1 Introduction

The recent deterioration of public safety is causing serious concern. Biometrics is one of the most promising technologies for alleviating this anxiety [1][2]. We are currently researching a form of biometrics that uses surveillance cameras set up in an actual space like a hospital or a railway station. For instance, we assume the hospital shown in Fig. 1. It is hoped that we obtain more appearance information at the entrance because at that point a suspicious person's invasion is obstructed.

Generally, because there is a broad field of view at the entrance, the images from different directions can be captured by using multiple cameras. If the monitoring system confirms that enough learning of an object's appearance has been performed, the automatic door opens and entry to the hospital is permitted. While the object is walking along the passage from the entrance to the sickroom, new images are captured with a surveillance camera arranged at each chosen position. The appearance information on the object is then updated by using the new images. When the object tries to enter the sickroom, another image of the object is captured by the surveillance camera set up in front of the sickroom. The personal identification processing is then performed by

using captured images, and when the result corresponds with the sickroom's authorization list, the automatic door opens and entry to the sickroom is permitted. A person's action history is generated with the processing of additional learning and identification. It is considered that different lighting conditions at each location has a strong influence on the accuracy of individual identification, though we assume to be able to control the lighting conditions almost constantly in indoor environments such as hospitals.



**Fig. 1.** Surveillance cameras in hospital

In this paper, we show a proposed method for dynamic visual learning based on a parametric eigenspace using sparsely distributed multiple cameras. We also describe some experiments to demonstrate the effectiveness of the proposed method.

## 2 Related Works

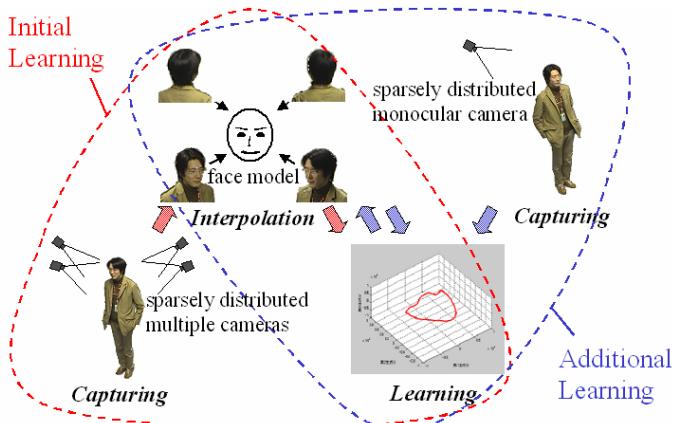
As people generally utilize facial appearances to identify individuals, the human face has potential for use as the most important type of information for biometric technology, making face recognition is one of the most important reasons for installing surveillance video sensors [3][4]. Most of these sensors demand a frontal or near-frontal facial view as the input appearance, and extract points of interest for the identification process (e.g., eyes, brows, nose and mouth). However, it is not always possible to capture the desired appearance with practical surveillance cameras. Therefore, in order to achieve a high recognition rate, the systems have to severely restrict people's activities, as in a portrait studio.

Parametric eigenspace is a well-known method for identifying objects with various appearances from images [5]. In order to generate a parametric eigenspace that achieves accurate identification, a number of different appearances, which can be collected by densely distributed multiple cameras, are generally required. However, it is not practical to set up a dense network of surveillance cameras around objects in the real world; general-purpose surveillance cameras are sparsely distributed, because the primary objective of the cameras is to monitor the widest area possible.

The objective of this paper is to realize a dynamic visual learning method based on parametric eigenspace to identify people captured with sparsely distributed multiple surveillance cameras. If we simply generate an eigenspace with a small number of sparsely distributed images, it is not possible to identify people from various viewing angles because the eigenspace cannot effectively learn the variety of appearances. Murase et al. attempted to fill the gap between the multiple images with a spline interpolation technique in a roughly generated eigenspace [6]. In our case, however, the gap is much larger than the one they dealt with in their research. The reason why spline interpolation does not work well with sparsely distributed images is that changes in the captured object's appearance are not considered in the interpolation process. The Virtualized Reality popularized by Kanade synthesizes a novel view with a 3D shape reconstructed from images captured with multiple cameras [7]. This technique makes it possible to interpolate multiple images by considering changes in appearance. In our proposed method, we mainly employ this technique to virtually capture multiple images and to generate an initial eigenspace. However, we need to modify this technique by simply using a 3D face model provided by the Galatea project [8], rather than recovering the 3D shape of the object from the captured multiple images, because it is still difficult to recover the 3D shape of the object from sparsely distributed multiple surveillance cameras.

### 3 Proposed Method for People Identification with Sparse Multiple Cameras

As illustrated in Fig. 2, the proposed method consists of two phases. We call the first phase the “initial learning phase,” and the second one the “additional learning phase.”



**Fig. 2.** Parametric eigenspace method with sparsely distributed multiple cameras

**Initial Learning:** In this phase, a view interpolation technique is used to generate an initial eigenspace. The surrounding sparsely distributed multiple cameras capture the target object at the same time. The 3D visual hull of the object is then reconstructed by

merging the multiple images using the shape from a silhouette method [9]. A simple 3D face model is fitted to the visual hull to mask the face region, and as a result, a 3D shape model is estimated. This method virtually captures multiple images by interpolating the appearance information of sparsely distributed images with the 3D shape model, and generates an initial eigenspace.

**Additional Learning:** To improve the method's ability to identify individuals in the eigenspace, the additional learning phase dynamically updates the eigenspace generated in the first phase. The extrinsic parameter of the additional capturing cameras is estimated as a relative pose with respect to the object. Then, the captured image is projected onto the 3D shape model with the parameter as texture information to improve the appearance of the interpolated images. By regenerating the eigenspace of the updated image data set, the discernment capability for personal identification of the eigenspace is improved.

## 4 Initial Learning Phase

### 4.1 Extraction of Head Region

As illustrated in Fig. 3, we set up a camera that looks down from the ceiling of the target space. Koyama et al. estimated 3D positions ( $X$ ,  $Y$ ,  $Z$ ) from the 2D coordinates in an overhead image ( $u, v$ ) while assuming that all target objects are at a constant height  $Y$  [10]. Under this assumption, a homographic transformation  $H$  is calculated that projects 2D coordinates in the overhead image onto a 3D plane. Eq. (1) is the equation of the projection. However, this assumption imposes a limitation on detecting the objects.

$$\lambda[X \ Z \ 1]^T = H[u \ v \ 1]^T \quad (1)$$

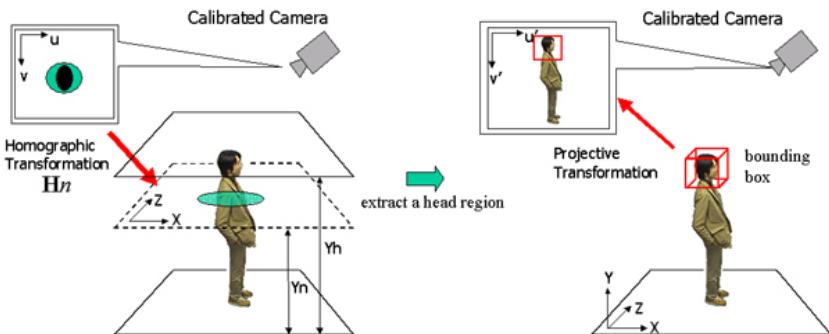


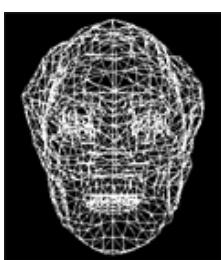
Fig. 3. Extraction of the head region

We improve this plane-based object-tracking method to detect the object's 3D position with arbitrary height information [11]. In this method, two base-planes are placed

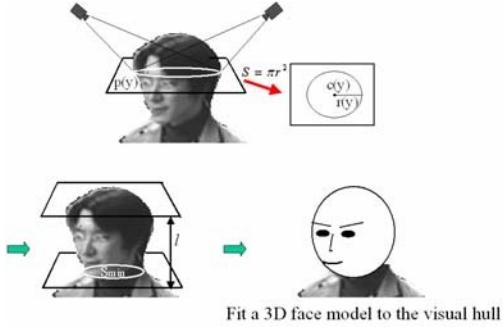
in 3D space, one on the ground ( $height=0$ ) and the other at height  $Y_h$ , after which the homographic transformations  $H0$  and  $H1$  are calculated. If the height of a new plane is given as  $Y_n$ , the homographic transformation  $Hn$  is estimated by interpolating  $H0$  and  $H1$ , as in Eq. (2).

$$\mathbf{H}_n = ((Y_h - Y_n)\mathbf{H}_0 - (Y_n)\mathbf{H}_1) / Y_h \quad (2)$$

The segmented foreground region is projected onto the 3D plane- $n$  by changing the height  $Y_n$  from 0 to  $Y_h$ . If the target object stands vertically like a human, the projected foreground region always includes a certain point ( $X, Z$ ) on plane- $n$  where the actual 3D object exists. By merging all of the n-planes (e.g., by an AND operation), the 3D position of the target object is estimated.



**Fig. 4.** A simple 3D face model



**Fig. 5.** 3D model estimation

## 4.2 3D Shape Model Estimation and View Interpolation

The accuracy of the estimated 3D shape seriously affects the quality of the interpolating images. To solve this problem, we employ a simple 3D face model provided by the Galatea project and fit the model to the visual hull to mask the face region. Fig. 4 shows the wire frame model of the 3D face model.

As illustrated in Fig. 5, we set a horizontal plane  $p(y)$  and project all of the foreground regions in all of the multiple images that have been calibrated in advance. The sliced 3D shape of the object is estimated as the overlapped region of all the projected regions [12]. We calculate the size and position of the captured head as the radius  $r(y)$  and the center of the circle  $c(y)=(X, Y, Z)$  by fitting a circle to the estimated shape on the plane, and execute the same process while shifting the horizontal plane along with the vertical axis to cover the head region. The head height  $l$  is estimated by searching for the minimum nonzero radius and the highest position of the head. As the right-hand side of Fig. 5 shows, we scale the 3D face model up/down with respect to head height and put the 3D model at the center of the head region. With this scaling process, we can reflect individual differences in head size.

The input multiple images are texture-mapped onto the estimated 3D face model using Projective Texture Mapping [13]. We render interpolation images while rotating a virtual camera around the 3D model in  $1^\circ$  increments. The blending parameter of each image (texture) is then calculated using the distance from the input multiple cameras to the viewpoint currently being interpolated.

### 4.3 Parametric Eigenspace Generation

#### 4.3.1 Normalization

Normalization consists of two steps: scale normalization and brightness normalization. In scale normalization, the extracted head region is resized to a square region (e.g.,  $128 \times 128$  pixels) with its center at the top of the head. In brightness normalization (Eq. (3)), each image  $\hat{x}_i$  is transformed to a normalized image  $x_i$ . In this normalizing process, our method has an advantage in that it is possible to completely control the conditions while generating the input image set, because they are synthesized images.

$$\mathbf{x}_i = \hat{\mathbf{x}}_i / \|\mathbf{x}_i\| \quad (3)$$

#### 4.3.2 Creating the Initial Eigenspace

To compute the eigenspace, we first subtract the average  $c$  of all images in the image set from each image in the set as shown in Eq. (4), where  $N$  is the total number of images in the image set. Then, to compute the eigenvectors of the image set, we define the covariance matrix  $Q$  also given in Eq. (4). The eigenvectors  $e_i$  and the corresponding eigenvalues  $\lambda_i$  of  $Q$  are to be computed by solving the eigenvector decomposition problem using Eq. (5). All of the eigenvectors of  $Q$  constitute a complete eigenspace. However, only a small number of eigenvectors is generally sufficient for capturing the primary appearance characteristics of objects. These  $k$  eigenvectors correspond to the largest  $k$  eigenvalues of  $Q$  and constitute the eigenspace. The number  $k$  of eigenvectors to be computed is selected based on recognition ability.

$$\mathbf{Q} = \mathbf{X}\mathbf{X}^T, \quad \mathbf{X} = [\mathbf{x}_1 - \mathbf{c}, \mathbf{x}_2 - \mathbf{c}, \dots, \mathbf{x}_N - \mathbf{c}] \quad (4)$$

$$\lambda_i \mathbf{e}_i = \mathbf{Q} \mathbf{e}_i \quad (5)$$

Each image  $x_i$  is projected onto the eigenspace. This is done by subtracting the average image  $c$  from  $x_i$ , and then finding the dot product of the result with each of the  $k$  eigenvectors, or dimensions, of the eigenspace as in Eq. (6), where  $i$  indicates the pose parameter. The result is a single point in the eigenspace, and by projecting all of the image sets, we obtain a set of discrete points. Pose variation between any two consecutive images is small; as a result, their projections in eigenspace are close to one another. Such a sampling creates a continuous loop in the eigenspace.

$$g_i = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k]^T (\mathbf{x}_i - \mathbf{c}) \quad (6)$$

## 5 Additional Learning Phase

### 5.1 Global Search

A surveillance camera captures a person who has already registered his/her appearance information in the initial learning phase. We extract the normalized head region from a captured image  $y_j$  with the above-described method, and project the region onto the calculated eigenspace. Concretely, the average  $c$  of the entire image set used to compute the eigenspace is subtracted from the input image  $y_j$ . The resulting image is projected onto eigenspace to obtain a point  $z_j$ . The equation for this the projection is Eq. (7). In order to search for the interpolated image most similar to the input image, we calculate the Euclidean distance of each eigenvector in the eigenspace between the input image  $z_j$  and the view-interpolated images  $g_i$ . The parameter of the interpolated image that has the most similar eigenvector to the input image's vector is estimated as the rough relative observing orientation.

$$z_j = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k]^T (\mathbf{y}_j - \mathbf{c}) \quad (7)$$

### 5.2 Local Search

Since the activity of the captured people is not controlled, their poses might be different from the poses in the initial learning phase. Thus, the estimated observing orientation contains a measure of error. In this section we describe a method for correcting the estimation error.

#### 5.2.1 Generating a Synthetic View

Once a 3D model has been generated, it is possible to render synthetic views of the modeled face with various rotation and translation parameters of the cameras. We assume the 3D model to be fixed, and the camera moves relative to it. In order to render images that have a slight difference in appearance with the matched image in the global search, synthetic view generation is repeated for a range of rotations around the  $x$ ,  $y$  and  $z$  axes (the  $x$  axis is through the neck, the  $y$  axis is through the head, and the  $z$  axis is perpendicular to the  $x$  and  $y$  axes). Typically we use plus or minus  $30^\circ$ , plus or minus  $5^\circ$ , and plus or minus  $10^\circ$  around the  $x$ ,  $y$  and  $z$  axis, respectively, quantized in  $5^\circ$  intervals, for a total of 195 synthetic views. Fig. 6 shows some sample generated images.



**Fig. 6.** Synthetic views that have slight differences in appearancefrom the matched image in a global search

### 5.2.2 Matching Against Synthetic Views

To find the best match, we compare the input image with each of the synthetic views. Consider the input image  $I$  that is being matched against a synthetic image  $S$ . The goodness of the match  $M$  between the two is found by computing Eq. (8), where  $I(i,j)$ ,  $S(i,j)$  are the image intensity at pixel  $(i,j)$  in the input and synthetic images, respectively. This is the well-known SAD method. The best-matching synthetic view is the one that minimizes this score.

$$M(s,t) = \sum |I(i,j) - S(i+s, j+t)| \quad (8)$$

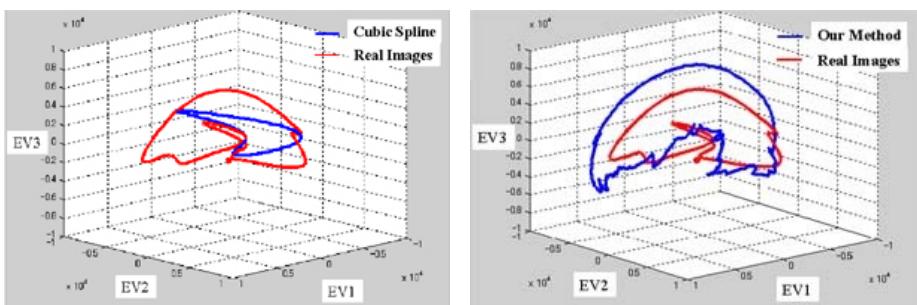
We are not, however, aiming to obtain the best-matched image but to get the camera parameters. To do this we use a downhill simplex method [14], which is a multi-dimensional optimization method, to estimate the camera parameters in order to minimize Eq. (8). To avoid a local minimum solution, we start from random values, and then pick the solution corresponding to the lowest minimum value.

### 5.3 Updating the Eigenspace

By projecting the input image with the estimated camera parameters, the texture information of the captured object's 3D shape model is updated. Then, as in the processing in Section 4.2, the interpolating images are regenerated while rotating a virtual camera around the 3D model. We thus again calculate an eigenspace with the updated image data set. If the appearance information of the 3D model becomes more accurate with additional texture mapping, the discernment capability for personal identification of the eigenspace improves further. We demonstrate the effectiveness of the proposed method in the next section.

## 6 Experiments

In these experiments, all images are captured by Sony network cameras (SNC-Z20) with a 640x480-pixel image size. All eigenspace figures in this section show only three of the most significant dimensions of the eigenspace since it is difficult to display and visualize higher-dimensional spaces. In other processes, the number of eigenspace dimensions used is 25, and in this case the accumulation-contributing ratio exceeds 95%.



**Fig. 7.** Comparison between the two types of interpolation methods

## 6.1 Interpolation Method Results

To evaluate the effect of the viewpoint interpolation in the proposed method, we compared the locus of the proposed method with the locus interpolated using a cubic spline function. In this experiment, the real images captured at intervals of about  $22.5^\circ$  were assumed to be the standard. Fig. 7(a) illustrates the result of the real images that are regarded as the standard and the result of interpolation using the cubic spline function. In this figure, to improve the visibility of the loop features, we have interpolated the real images with the cubic spline function that is generally used to interpolate eigenspace. Fig. 7(b) represents the result of the real images that are regarded as the standard and the result of interpolation using a 3D model. Comparing Fig. 7(a) with Fig. 7(b), we see that interpolation using a 3D model is more complex than that using the cubic spline function for the real images. From this result, when we have only sparsely input images, it can be said that interpolation using a 3D model can create a higher discernment capability eigenspace than interpolation using the cubic spline function.

## 6.2 Additional Learning Results

Next, we examine how the additional learning process updates the initial interpolation images. The upper row in Fig. 8 shows interpolated images from the four initially captured images. With our proposed additional learning method, the encircled regions are updated by newly captured images. On the bottom row of Fig. 8 we can see that the realism of the interpolated images is improved by the replacements provided by additional learning.



**Fig. 8.** Result of additional learning phases

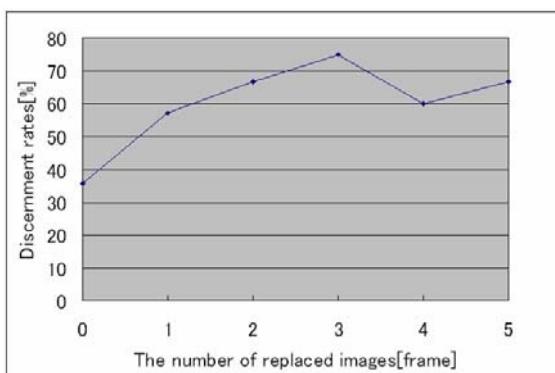
## 6.3 Results of Discernment Capability

We have already experienced how the discernment capability among four persons varies as replacement is performed. In this experiment, first, one person from a group of four is chosen as the identification object, and interpolation images of the person are subsequently generated with a 3D model. Then, 50 face images of all four people are projected to the eigenspace generated by using the interpolation images, and we calculate the distance in the eigenspace among the interpolation images and the projection

point. The distance (threshold) is obtained by comparing the projection point of the 50 face images of the four persons with the projection point of another 50 face images of the same four persons. Fig. 9 shows how the discernment capability among the four persons varies as replacement is performed. We can see that the discernment rate improves as additional learning progresses, and that discernment capability has improved. However, the discernment rate decreased when the number of additional images became four from three. We think this loss of performance occurs due to the gap in the texture mapping and errors in extraction.

## 7 Conclusions

We proposed a learning method for parametric eigenspace using sparsely distributed multiple cameras. We have demonstrated that the discernment capability of the initial eigenspace is improved by repeating the updating process, and that interpolation using a 3D model more closely resembles the real image than interpolation using the cubic spline function. Future work will include reducing errors in extraction and a method to put together various pieces of information for personal identification. This research was supported in part by the National Institute of Information and Communications Technology.



**Fig. 9.** Number of updates vs. Discernment capability results

## References

- [1] A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, A. Ross, Biometrics: A Grand Challenge, *Proc. of IC PR*, 2004, Vol. 2, pp. 935-942
- [2] A. Pentland, Looking at People: Sensing for Ubiquitous and Wearable Computing, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000, Vol. 22, No. 1, pp. 107-118.
- [3] S. Lao, T. Kozuru, T. Okamoto, T. Yamashita, N. Tabata, M. Kawade, A fast 360-degree rotation invariant face detection system, *ICCV*, 2003
- [4] [http://www.identix.com/products/pro\\_security\\_bnp\\_argus.html](http://www.identix.com/products/pro_security_bnp_argus.html)

- [5] H. Murase, S.K. Nayar, Parametric Eigenspace Representation for Visual Learning and Recognition, *Workshop on Geometric Method in Computer Vision, SPIE*, 1993, pp. 378-391.
- [6] H. Murase, S.K. Nayar, Illumination Planning for Object Recognition Using Parametric Eigenspaces, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1995, Vol. 16, No. 12, pp. 1219-1227.
- [7] T. Kanade, P.J. Narayanan, and P.W. Rander, Virtualized reality: concepts and early results, *Proc. of IEEE Workshop on Representation of Visual Scenes*, 1995, pp. 69-76.
- [8] <http://hil.t.u-tokyo.ac.jp/~galatea/>
- [9] A. Laurentini, The Visual Hull Concept for Silhouette-Based Image Understanding, *IEEE Trans. on Pattern Analysis and Machines Intelligence*, 1994, Vol. 16, No. 2, pp. 150-162.
- [10] T. Koyama, I. Kitahara, Y. Ohta, Live Mixed-Reality 3D Video in Soccer Stadium, *Proc. of ISMAR*, 2003, pp. 178-187
- [11] I. Kitahara, K. Kogure, N. Hagita, Stealth Vision for Protecting Privacy, *Proc. of ICPR*, 2004, Vol. 4, pp. 404-407
- [12] I. Kitahara, H. Saito, S. Akimichi, T. Ono, Y. Ohta, and T. Kanade, Large-scale Virtualized Reality, *CVPR, Technical Sketches*, 2001.
- [13] C. Everitt, Projective Texture Mapping, *White paper, NVidia Corporation*, 2001.
- [14] J.A. Nelder and R. Mead, A Simplex Method for Function Minimization, *Computer Journal*, 1965, Vol. 7, pp. 308-313.

# 3D-Connected Components Analysis for Traffic Monitoring in Image Sequences Acquired from a Helicopter

Matthieu Molinier, Tuomas Häme, and Heikki Ahola

VTT Technical Research Center of Finland,  
Remote Sensing Group, P.O.Box 1201,  
FIN-02044 VTT, Finland

{matthieu.molinier, tuomas.hame, heikki.ahola}@vtt.fi  
<http://www.vtt.fi/tte/research/tte1/tte14/>

**Abstract.** The aim of the study was to develop methods for moving vehicle tracking in aerial image sequences taken over urban areas. The first image of the sequence was manually registered to a map. Corner points were extracted semi-automatically, then tracked along the sequence, to enable video stabilisation by homography estimation. Moving objects were detected by means of adaptive background subtraction. The vehicles were identified among many stabilisation artifacts and tracked, with a simple tracker based on spatiotemporal connected components analysis. While the techniques used were basic, the results turned out to be encouraging, and several improvements are under scrutiny.

## 1 Introduction

Traffic monitoring in urban areas has become a necessity for transportation and infrastructure planning. Ground-based video surveillance systems partially fulfil this need. However, this kind of systems are not easily transportable as they are often adapted to a given road configuration, and only cover a narrow area. On the contrary, aerial imagery acquired from a helicopter offers both a mobile system and a wide field of view, for monitoring several urban traffic spots in a same image sequence.

Hoogendoorn et. al [12] used a straightforward approach to monitor highway traffic from a helicopter, with camera motion compensation and residual motion detection by background subtraction. 90% of the vehicles were successfully detected by thresholding, then tracked by template matching. Yet this approach relied on greyscale imagery, used a same background image over the sequence regardless of illumination changes, and its efficiency in congested situations was altered. Han and Kanade [9] proposed a method to recover, given a monocular image sequence, the scene structure, the trajectories of the moving objects and the camera motion simultaneously. The technique is appealing since it does not require any camera motion compensation to detect and track vehicles. It has been successfully applied on an aerial video, but relies on the assumption that

the objects are in linear motion. Fuse et al. [4], [5] developed a system for urban traffic monitoring with colour image sequences acquired from high-altitude platforms. It includes video stabilisation, background subtraction then update by Kalman filtering, and vehicle tracking by an original spatiotemporal clustering of two features - the value after background subtraction and the optical flow value. Vehicle shadows were discarded following [3]. The overall approach turned out to be very effective, but is rather complicated and requires 8 thresholds. Moreover, little is said about how to handle artifacts due to an inaccurate video stabilisation followed by background subtraction. Still, an interesting conclusion of their study is that temporal resolution (frame rate) is more important than spatial resolution, if one wants to achieve good tracking performance.

Aiming at an operative system, we present a method for moving vehicle detection and tracking in high-resolution aerial colour image sequences. The goal of this study was to accurately estimate the speed of traffic objects like cars, buses or trams, and make the information available on GIS (Geographic Information System) for later analysis. The long-term purpose of this work is to collect traffic data over wide urban areas by means of aerial imagery, and assess overall traffic conditions for city planning. The method relied on simple techniques, yet achieved satisfying results.

## 2 Data and Pre-processing

### 2.1 Material

Acquisition campaigns were made over the city of Helsinki in June 2002, with a helicopter and a standard digital camera oriented near nadir direction. Monocular 25fps colour videos were acquired at altitudes ranging from 200m to 500m, over traffic spots like crossroads. The image sequence used in this study consists of 500 uncalibrated frames taken at 200m during 20s ; each frame has 768\*560 pixels and a 17cm ground resolution (Fig. 1(a)). City of Helsinki provided us with digital map of the monitored urban areas (Fig. 1(b) covers an area of 134\*110m).

### 2.2 Image Sequence Registration

Before detecting moving objects, the camera motion needed to be compensated. Following Censi et al. [1], the scene was assumed to be planar. Points of interest were manually extracted in the first frame, then tracked along the sequence. Other frames were automatically registered to the reference image, through the robust estimation of the plane-to-plane homography that links images of a planar scene taken at different viewpoints.

**Homography Estimation Between Two Consecutive Frames.** Because a fully automated selection of relevant points proved challenging in such a complex urban scene (with a lot of salient objects), points were first selected manually on the road surface. The Harris corner detector [10] extended to colour images [13] was then run to refine point extraction. These points were tracked in the following

frames by considering the local maximum response of the corner detector around their previous locations.

The homography can be estimated with 4 point correspondences. In practice, 20 point correspondences were established to solve an overconstrained linear system through Singular Value Decomposition [1], [7]. Points incorrectly tracked were declared as outliers by the X84 rule [6] ; noting  $\hat{H}$  the homography estimated between points  $p_j^k$  of frame  $I^k$  and  $p_j^{k+1}$  of frame  $I^{k+1}$ , outliers were found by examining statistics of the residuals  $r_j$ :

$$r_j = \|p_j^k - \hat{H}p_j^{k+1}\|. \quad (1)$$

Only inliers were used for the final estimation of  $H$ . The expected positions of those points that generated an outlier residual were guessed by applying the robustly estimated homography [1]. The final homography estimate was applied to frames by inverse mapping, using nearest neighbour interpolation.

**Image Sequence Geocoding and Homography Composition.** Under the same assumption of scene planarity, a homography was estimated between the first image of the sequence and the GIS map, after selecting and tieing points manually, as shown in Fig. 1. The first frame was then aligned to the map, to produce a geocoded image - Fig. 2.

The actual point correspondences used for the homography estimation, in the stabilisation stage, were between a given frame and the geocoded first frame. By doing so, each frame of the sequence was only warped once to generate a geocoded, stabilised video sequence. The whole process of video stabilisation and geocoding is summed up on Fig. 3. Once the camera motion was compensated, the image sequence looked almost as if taken with a static camera, with the exception of buildings - mainly because of parallax effect. Being of no interest for vehicle tracking, the buildings were masked out before further processing.

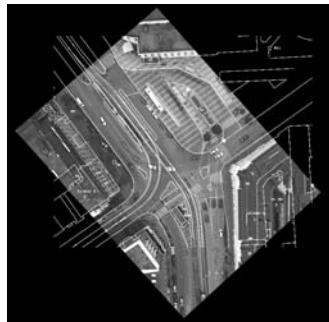


(a) Corner points in 1st frame

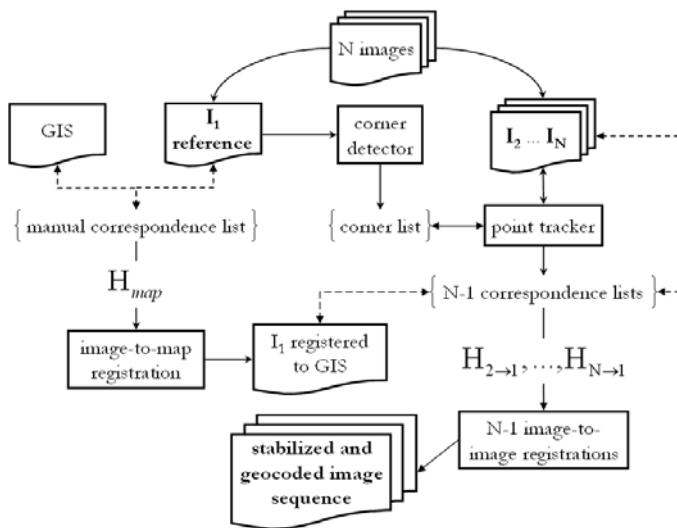


(b) Corresponding points in GIS

**Fig. 1.** Manual image-to-map registration



**Fig. 2.** First frame of the sequence registered to GIS



**Fig. 3.** Image sequence stabilisation and geocoding

### 3 Methods

#### 3.1 Motion Detection by Adaptive Background Subtraction

Background subtraction is a widely used technique for motion detection in traffic monitoring applications [2]. In its adaptive form, a model of the background is generated and updated as the image sequence runs [8], to account for illumination changes. In our study, detection was carried out in HSV colour space (Hue Saturation Value), considering only the V component.

**Background Initialisation.** Because a vehicle-free image of a traffic scene does not generally exist, it has to be generated. The median was computed

pixel-wise over the 101 first frames of the sequence, forming an initial image of the background. The same approach for background initialisation was used in [4]. This requires that some frames of the sequence are known in advance ; in a real-time system, that could be done during a setup phase.

Because the video was not perfectly stabilised<sup>1</sup>, computing the median over the sequence also introduced a spatial averaging : the background image appeared smoothed. Still, this resulted in a better initial background image than if we had taken the first image of the sequence. When initialising the background with the first frame of the sequence, all moving vehicles in the following frames already appear in the background, delaying their detection by the time the background updates.

**Moving Object Detection.** Once a background image was available, it was subtracted to the current image, forming the difference image ( $DI$ ). Double thresholding was applied to  $DI$  in order to detect moving vehicles brighter (high threshold  $t_H$ ) and darker (low threshold  $t_L$ ) than the background. Both thresholds were set automatically by histogram analysis, considering a fixed percentage of the histogram maximum [8]. Values were empirically set to 5% for  $t_L$  and 3% for  $t_H$ , in other words  $t_L$  is the negative intensity level that has a number of elements representing 5% of the histogram maximum number of elements.

By thresholding  $DI$ , an object mask ( $OM$ ) was obtained, that was cleaned by morphological operations to remove isolated pixels. Because a car windshield is a dark feature when seen from aerial imagery, its colour is close to that of the background (grey road surface). It often happened that the detected moving cars were split into two nearby blobs separated by their windshield, not detected as a moving region. A series of morphological operators was applied to the object mask  $OM$  to join the blobs and fill each object. A 8-connected components analysis was then run on  $OM$  to remove small objects.

**Background Update.** Once motion detection was done for the current image  $CI$ , the background image was updated as follows. First, the so-called instantaneous background was formed [8]. For all pixels in  $OM$  where a moving object was detected, the current background  $CB$  was sampled. For other pixels, where no motion was detected, the current image  $CI$  was sampled. The instantaneous background  $IB$  was thus computed as :

$$IB = OM.*CB + \overline{OM}.*CI . \quad (2)$$

where  $.*$  denotes a pixel-wise multiplication and  $\overline{OM}$  the complement to 1 of the binary mask. The background update  $BU$  was a weighted average of that instantaneous background obtained in Eq. 2 and the current background :

$$BU = \alpha IB + (1 - \alpha)CB . \quad (3)$$

---

<sup>1</sup> Strictly speaking, it can hardly be, even with a deformation model more complete than plane-to-plane homography.

$\alpha$  is the background learning rate : background updating has to be fast enough to adapt to illumination changes, while ignoring momentary changes [8]. A value of  $\alpha = 0.3$  was found empirically to be satisfactory.

### 3.2 Vehicle Tracking by 3D-Connected Components Analysis

**Motivation.** The outcome of moving object detection is a sequence of masks containing pixels declared as in motion. Some of those pixels correspond to actual moving vehicles in the scene, but others can be gathered in regions not representing any real moving object, e.g. residual effects of illumination changes not modelled in the background update. Fig. 4 shows masks of pixels in motion, in two consecutive frames. While the actual vehicles appeared in both masks, regions appeared in Fig. 4(a) that were not present in the other mask. The region on top of Fig. 4(b) corresponds to a passenger crosswalk, detected as in motion because of misalignment of frames in the stabilisation process, and because its white colour over the dark road surface makes it a salient feature. These false detections had to be discarded before tracking vehicles.

A threshold on the size of the moving regions would not have been suitable for distinguishing stabilisation artifacts from vehicles, because some artifacts turned out to be bigger than cars. Based on the observation that artifacts flickered throughout the sequence (i.e. appeared then disappeared in consecutive frames), whereas actual vehicles appeared in all masks, we used temporal consistency as a criterion to identify vehicles, through 3D-connected components analysis.

Vehicle tracking was achieved at the same time, by labelling spatiotemporal regions. The video frame rate (25fps) guaranteed that there was an overlap between the same vehicle in two consecutive frames. This kind of approach may not have been used in traditional traffic monitoring applications, due to the many occlusions of vehicles by other vehicles when using a ground-based or aboveground camera. However, in aerial image sequences acquired near nadir direction, there is no such occlusion. Detected moving objects supposedly appear disjoint in each mask  $OM$ . Consequently, 3D-connected components analysis should allow vehicle tracking without risking that two vehicles are tracked as a single moving blob.

**Tracker and Vehicle Speed Estimation.** A 26-connected components analysis was run on the binary masks sequence after motion detection stage, defining 3D-regions in the spatiotemporal domain (Fig. 5(a)). Regions whose temporal extension did not exceed 25 frames (1s) were discarded as stabilisation artifacts, while vehicles left long trails in spatiotemporal domain (Fig. 5(b)).

Since the image sequence has been previously registered to a GIS, there exists a correspondence between dimensions in the image and real dimensions, allowing speed estimations. The centroids of the vehicles were extracted, and used to estimate their speed in an interval of 12 frames. The speed estimation would not be accurate if run between consecutive frames, because the inter-frame vehicle displacement is low at such frame rate. The speeds were then mapped on the registered sequence (Fig. 6). For each vehicle, speed and position was recorded for later traffic data analysis.

## 4 Results and Discussion

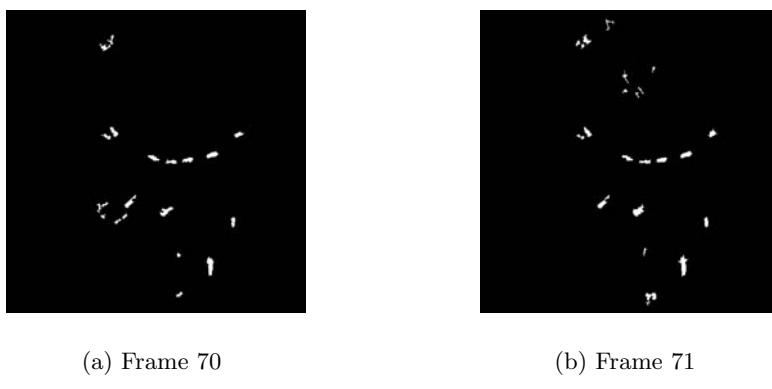
Almost all moving vehicles were correctly detected in the sequence, but along with stabilisation artifacts (Fig. 4). Ways to address this issue of false detection could range from improving image stabilisation, to using a more elaborate background subtraction technique, e.g. some of those tested in [2].

Moving vehicle detection was problematic, especially with dark cars, which led us to empirically choose thresholds that would probably not be suited to another aerial image sequence with different lighting conditions. There were some frames in which a dark grey moving car was not detected at all. Besides, windshields tend to split moving cars into two moving blobs, which later altered the tracking quality. We used morphological operators in an attempt to join the two blobs of a same car. These operators included flood-fill to get full objects in the masks (*OM*). This was a straightforward attempt to obtain complete moving objects, that turned out to also accentuate false detections due to misalignment of frames. Instead of trying to join moving blobs in the masks, we could have used spatiotemporal clusters merging [4] as a post-processing after tracking.

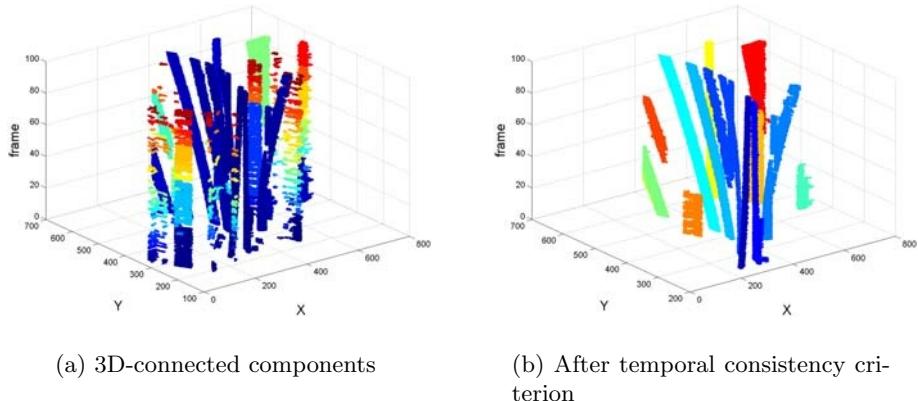
The 3D-connected components analysis allowed to identify vehicles among stabilisation artifacts, and track them. It also provided a spatiotemporal visualisation of vehicles trajectories in the image sequence (Fig. 5).

Vehicles correctly detected were also successfully tracked. The dark car that was undetected during some frames in the middle of the sequence was not tracked during those frames, since tracking relied solely on detection. One way to circumvent the direct effect of misdetection on tracking efficiency, could be to add a predictor in the tracking process, such as the widely used Kalman filter. For those vehicles correctly detected and tracked, the velocity vectors showed a consistent direction along the sequence (Fig. 6). Once the position of vehicles was known, speed estimation was rather straightforward.

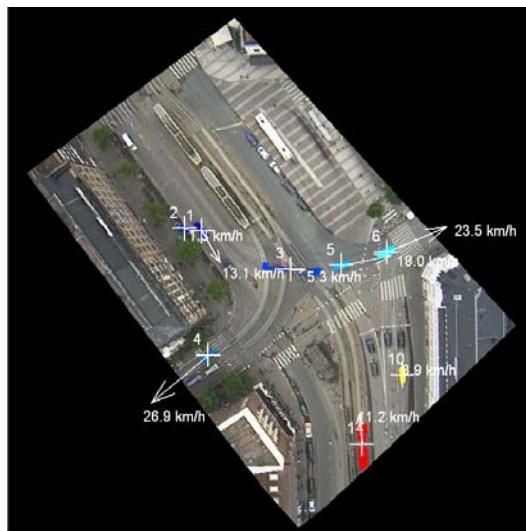
Spatiotemporal connected component analysis was a simple and appealing method to track vehicles, and at the same time to deal with false detections resulting from misalignment of frames, but it also has its drawbacks. With the



**Fig. 4.** Moving objects masks



**Fig. 5.** Moving objects trajectories in spatiotemporal domain (2D+t view)



**Fig. 6.** Vehicles speeds

current implementation, it can only be applied to an image sequence treated as a whole, i.e. with all frames loaded in memory. The number of frames that could be processed was limited to about a hundred with Matlab and our computer, reducing the interest of the method for long image sequences. 3D-connected components analysis could still be applied in sliding spatiotemporal windows of some frames<sup>2</sup>, for both tackling the memory limitation and allowing online tracking

<sup>2</sup> Enough frames so that a stabilisation artifact appears as a discontinuous region in spatiotemporal domain.

of objects, as the sequence runs. Another issue is that tracking efficiency was very dependent from the detection quality. There was no mechanism to prevent two cars moving closely together from being merged into a single spatiotemporal region, when their corresponding masks were 26-connected - which can occur when moving object detection is not accurate. Including colour, shape information or velocity continuity in the tracking procedure [4] might help increasing its robustness. Last, a connected components-based tracker is inherently limited as for the maximum speed of a vehicle that can be tracked. This depends on the video frame rate, image resolution, size and speeds of vehicles as well as the quality of moving object detection.

In the video stabilisation stage, point tracking relied on an unconventional technique, namely running a corner detector in each frame - the point correspondence problem was not rigorously solved. State-of-the-art point tracking algorithms, such as the Kanade-Lucas-Tomasi feature tracker [14], would certainly improve the point tracking quality, hence video stabilisation.

Last, the whole approach would need to be validated with ground truth data, ideally a vehicle of which we would know the speed or position at each moment during the sequence.

## 5 Conclusion

We proposed a simple processing chain for moving vehicle detection and tracking in image sequences acquired from a helicopter. After compensating the camera motion and registering the video to a GIS, residual motion was detected by adaptive background subtraction and vehicles tracked by spatiotemporal connected components analysis. The main loopholes of the system lie in the detection stage, and greatly affected the overall tracking performance. Cars correctly detected were successfully tracked hereafter, and the estimation of their speed seemed consistent - while this would need validation with ground truth data.

Lots of improvements can be made to catch up with state-of-the-art methods. The developed approach relied mainly on pixel-based methods, and some steps like video stabilisation still lack automation. Integration of higher-level image processing should also help making tracking more robust, especially when detection partly fails. One possibility would be to consider detailed vehicle models for tracking, such as Hinz [11] developed to detect cars in high-resolution images.

Yet, the results obtained were fairly good, and are encouraging for the development of an operative traffic monitoring system with aerial platforms.

## Acknowledgments

This work was included in a larger scale project, ENVIMON, partly funded by the National Technology Agency of Finland (Tekes). We wish to acknowledge the Helsinki City Planning Department for its interest in the traffic monitoring application. We also thank Markku Rantasuo for the data acquisition.

## References

1. A. Censi, A. Fusiello, V. Roberto : Image stabilization by features tracking. Proceedings of the 9th International Conference on Image Analysis and Processing, Venice (1999).
2. S.-C. Cheung, C. Kamath : Robust techniques for background subtraction in urban traffic video. Proceedings of SPIE Electronic Imaging : Visual Communications and Image Processing, San Jose (2004).
3. R. Cucchiara, C. Grana, M. Piccardi, A. Prati : Detecting Moving Objects, Ghosts, and Shadows in Video Streams. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. **25** no. 10 (2003) 1337–1342.
4. T. Fuse, E. Shimizu, R. Maeda : Development of techniques for vehicle manoeuvres recognition with sequential images from high altitude platforms. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Corfu (2002), vol. **34** part. 5, 561–566.
5. T. Fuse, E. Shimizu, T. Shimizu, T. Honda : Sequential image analysis of vehicle manoeuvres : evaluation of vehicle position accuracy and its applications. Journal of the Eastern Asia Society for Transportation Studies, vol. **5** (2003) 1991–2002.
6. A. Fusiello, E. Trucco, T. Tommasini, V. Roberto : Improving Feature Tracking with Robust Statistics. Pattern Analysis and Applications, vol. **2** no. 4 (1999) 312–320.
7. R. Garcia Campos : A proposal to estimate the motion of an underwater vehicle through visual mosaicking. PhD Thesis, University of Girona (2001), chap. 5, 131–137.
8. S. Gupte, O. Masoud, R.F.K. Martin, N.P. Papanikolopoulos : Detection and classification of vehicles. IEEE Transactions on Intelligent Transportation Systems, vol. **3** no. 1 (2002) 37–47.
9. M. Han, T. Kanade : Reconstruction of a Scene with Multiple Linearly Moving Objects. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island (2000).
10. C. J. Harris, M. Stephens : A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference, Manchester (1988), 147–151.
11. S. Hinz : Detection and counting of cars in aerial images. Proceedings of IEEE International Conference on Computer Vision, Barcelona (2003), vol. **III**, 997–1000.
12. S.P. Hoogendoorn, H.J. van Zuylen, M. Schreuder, B. Gorte, G. Vosselman : Microscopic traffic data collection by remote sensing. 82nd Annual Meeting of Transportation Research Board (TRB), Washington D.C. (2003).
13. P. Montesinos, V. Gouet, R. Deriche : Differential invariants for color images. Proceedings of IAPR International Conference on Pattern Recognition, Brisbane (1998), 838–840.
14. C. Tomasi, T. Kanade : Detection and tracking of feature points. Carnegie Mellon University Technical Report, CMU-CS-91-132, Pittsburgh, PA (1991).

# Joint Modeling of Facial Expression and Shape from Video

T. Tamminen\*, J. Kätsyri, M. Frydrych, and J. Lampinen

Laboratory of Computational Engineering,  
Helsinki University of Technology,  
P.O. Box 9203, 02015 HUT, Finland

[toni.tamminen@tkk.fi](mailto:toni.tamminen@tkk.fi), [katsyri@lce.hut.fi](mailto:katsyri@lce.hut.fi), [frydrych@lce.hut.fi](mailto:frydrych@lce.hut.fi),  
[jouko.lampinen@tkk.fi](mailto:jouko.lampinen@tkk.fi)

**Abstract.** In this paper, we present a novel model for representing facial feature point tracks during an facial expression. The model is composed of a static shape part and a time-dependent expression part. We learn the model by tracking the points of interest in video recordings of trained actors making different facial expressions. Our results indicate that the proposed sum of two linear models - a person-dependent shape model and a person-independent expression model - approximates the true feature point motion well.

## 1 Introduction

Human facial expression is a widely studied topic both in computer vision and psychology as a great deal of human communication is carried out through facial expressions, in addition to words. In computer vision, the focus has been on recognition and modeling, while psychologists are interested in both the emotional processes behind facial expressions as well as the brain mechanisms underlying the recognition of emotions from expressions [1]. A most comprehensive system for analyzing facial displays is the Facial Action Coding System (FACS) [2]. It is based on anatomy and has been widely used by psychologists and recently also for automated classification of facial actions. A limitation of FACS is, however, the lack of detailed spatial and temporal information [3]. Improved systems include the FACS+ system [3], the AFA system [4], and many others [5].

Most of the proposed approaches to facial expression modeling are rather holistic in nature, i.e. they model expressions as a whole instead of tracking individual facial feature points. Furthermore, often only the expression is modeled, and no attention is paid to the shape of the face. The combination of these poses a serious problem in some applications such as feature-based object recognition. To deal with the problem, we present a new model for the fiducial feature points of the human face which aims to encompass both the interpersonal facial shape

---

\* Author supported by the Finnish Cultural Foundation, the Jenny and Antti Wihuri Foundation, and the Nokia Foundation.

variation and the expression-dependent dynamic variation. We aim to represent the sources of variation with orthogonal linear vector bases, which facilitates the analysis and use of the model.

The paper is organized as follows. Section 2 describes our data and our feature tracking system. Section 3 introduces our face model, and Sect. 4 presents analysis of the model and some reconstruction results. Section 5 concludes.

## 2 Data Acquisition and Feature Tracking

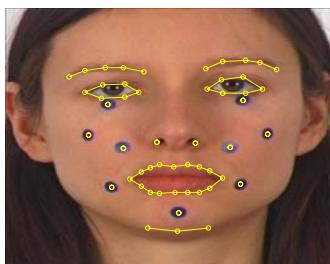
### 2.1 The Data

Facial expressions were recorded from actors trained to express certain prototypical emotional facial expressions. The recordings included seven facial expressions related to basic emotions [6] (two different happiness recordings), two facial expressions related to blends of basic emotions and one emotionally meaningless facial expression. The facial expression prototypes (Table 1) were based on existing descriptive literature [2] [7] and defined for FACS by a certified FACS coder.

The recordings were made from 6 actor students from the Theatre Academy of Finland (3 men and 3 women, age range 23–32 years); hence, there were 60 video streams in total. The actors were asked both to express the given facial configuration exactly and to experience the required emotion. The actors

**Table 1.** Facial Expression Prototypes

Facial expression	FACS Action units	Facial expression	FACS Action units
Anger	4+5+7+24	Sadness	1+4+7+15+17
Disgust	9+10+17	Surprise	1+2+5+25+26
Fear	1+2+4+5+7+20+25	Happiness + surprise	1+2+5+6+12+25+26
Happiness (mouth open)	6+12+25	Happiness + disgust	6+9+10+12+17
Happiness (mouth closed)	6+12	Mouth opening	25+26



**Fig. 1.** A sample feature graph, with the added dark markers showing. The light dots mark the tracked features

practised the facial expressions individually for approximately 5-10 hours. One practise recording was carried out with the possibility for feedback before the actual recording session.

The recordings contained short (1-2s.) video sequences showing the change from neutral to the target state. Nine markers were placed on perceptually meaningful locations (Fig. 1) to ease the tracking of facial changes unrelated to clear facial features. The recording setup included 2 professional photographing lamps (Elinchrom scanlite 1000) and a digital camcorder (Sony DSR-PD100AP). The recordings were made at 25 (interlaced) frames per second (fps) with a resolution of 572\*726 pixels. To reduce computational cost and memory usage, the videos were clipped to include only the facial area and resized to 256\*256 pixels.

## 2.2 Feature Tracking

The KLT tracker and its derivatives are used widely in visual feature tracking [8] [9] [10]. However, we decided to test the possibilities of an automated tracker based on Gabor filters [11] and Bayesian inference [12] as an extension of our static object matching system [15]. A similar approach (without the Bayesian context) has previously been presented by Mckenna et al. [13].

To reduce clutter and to make the features more distinctive, each image  $\mathbf{I}^t$  in a video sequence (with time steps  $t$ ) is first transformed into feature space,  $\mathbf{I}^t \mapsto \mathbf{T}^t$ , by filtering it with a Gabor filter bank with 3 frequencies and 6 orientations. All computations are then performed using the transformed images  $\mathbf{T}^t$ . The face is represented as a planar graph containing  $n$  nodes (Fig. 1) with coordinates  $\mathbf{X}^t = \{x_1^t, \dots, x_n^t\}$ . Each node  $i$  has an associated feature vector  $\mathbf{g}_i^t$ , which is formed by stacking the responses of the Gabor filter bank as a vector.

The features are tracked by finding, at each time step, the maximum a posteriori estimate of the location of each feature around its previous location. That is, we compute the posterior density of each feature in some search area  $A_i$  given the transformed image, the corresponding feature vector  $\mathbf{g}_i^t$  and the other feature locations  $\mathbf{x}_{\setminus i}^t$ , and maximize it:

$$\max_{x_i^t \in A_i} p(x_i^t | \mathbf{T}^t, \mathbf{g}_i^t, \mathbf{x}_{\setminus i}^t) \propto p(\mathbf{T}^t | x_i^t, \mathbf{g}_i^t) p(x_i^t | \mathbf{x}_{\setminus i}^t), \quad (1)$$

where we have used Bayes's formula to write the posterior probability as the product of the likelihood and prior parts. The likelihood measures the probability of observing the image given a feature configuration, while the prior gives the distribution of the feature location given the locations of the other features.

We can not measure the probability of observing an image directly, since we do not have a comprehensive image model. Hence, we approximate the likelihood by computing the similarity between the stored feature vectors  $\mathbf{g}$  and the perceived image  $\mathbf{T}$ . We use the criterion presented by Wiskott et al. [14] to obtain the similarities. As the prior we use a simple model in which the features are allowed independent Gaussian variations from a known mean shape  $\mathbf{r}^t$  [15]:

$$p(x_i^t | \mathbf{x}_{\setminus i}^t) = N(f(\mathbf{x}_{\setminus i}^t, \mathbf{r}^t), \sigma^2), \quad (2)$$

where  $f$  is a function that translates and scales the mean shape to correspond to the current graph, and  $\sigma^2$  is the variance of the Gaussian, which was set to some suitable value so that the tracker would function well (with  $256 \times 256$  images, we used  $\sigma = 5$ ).

As the video sequence progresses, both the features  $\mathbf{g}$  and the mean shape  $\mathbf{r}$  change. To adapt the tracker to this, at each time step we change  $\mathbf{g}$  and  $\mathbf{r}$  according to the newly obtained values:

$$\mathbf{g}^{t+1} = \alpha_g \mathbf{g}^t + (1 - \alpha_g) \mathbf{g}^1 \quad (3)$$

$$\mathbf{r}^{t+1} = \alpha_r \mathbf{r}^t + (1 - \alpha_r) \mathbf{r}^1, \quad (4)$$

where  $\alpha_g$  and  $\alpha_r$  are parameters controlling the extent of the adaptation. Using  $\mathbf{g}^1$  and  $\mathbf{r}^1$  as the baseline values reduces the probability of the tracker adapting to track a completely spurious feature, as the effect of the original Gabor jets and mean shape never disappears completely.

The initial feature locations  $\mathbf{X}^1$  and Gabor jets  $\mathbf{g}^1$  are obtained by manually annotating the features on the first image of one video sequence and then using the image and the annotations as training data for matching the features in the first images of other sequences (for details of the matching, see [15]). The mean shape  $\mathbf{r}^1$  is taken to be equal to  $\mathbf{x}^1$ .

The performance of the tracker was varying. In some streams it tracked the features perfectly, in some streams there were considerable errors. The tracking could be improved in numerous ways such as including a systematic model for the motion of the features or designing a more sophisticated adaptation scheme. However, since the tracking was not the main object of interest in this paper, the improvements were left to a further study.

### 3 Face Model

In our model, our aim is to find separate orthogonal bases for representing variations due to face shape and facial expression. A similar approach has been proposed by Abboud and Davoine [16]; however, they do their modeling in the AAM framework [17] and model only the start- and endpoints of expressions, whereas we are interested in the the whole track of the fiducial feature points during an expression.

To model the dynamics of the expression, we include the time correlations of the feature point tracks into our expression model, that is, the expressions are described by vectors of length  $n \times n_t$ , where  $n_t$  is the number of time steps. We assume that the tracks  $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^{t_f}\}$  can be represented as the sum of two linear models: a person-dependent shape model and a person-independent expression model so that

$$\mathbf{X} = \mathbf{1} \otimes (\mathbf{m} + \mathbf{S} \beta_{person}) + \mathbf{E} \beta_{expression} + \epsilon, \quad (5)$$

where  $\mathbf{m}$  is the mean shape,  $\mathbf{S}$  is the matrix of the base vectors of the shape space,  $\mathbf{E}$  is the matrix of the base vectors of the expression space,  $\beta_{person}$  is

the person-dependent vector of coordinates in the shape space,  $\beta_{expression}$  is the expression-dependent vector of coordinates in the expression space,  $\mathbf{1}$  is a vector of ones,  $\otimes$  is the Kronecker product, and  $\epsilon$  is Gaussian noise. Note that the Kronecker product is required to make the computation of the sum possible, as the shape and expression vectors are of different lengths. At time step  $t$  the graph is

$$\mathbf{X}^t = \mathbf{m} + \mathbf{S}\beta_{person} + \mathbf{E}^t\beta_{expression} + \epsilon^t, \quad (6)$$

where  $\mathbf{E}^t$  contains the elements of the expression base vectors that apply to time step  $t$ .

To estimate the base vectors of the shape and expression spaces, we need to separate the shape and expression effects. This is done in two phases:

1. Estimate the mean shape and the shape base vectors via PCA [18] from the initial feature graphs  $\mathbf{X}^1$ . We assume that the video streams start from a neutral expression, that is,  $\mathbf{E}^1 = 0$ .
2. To remove the effect of the shape from subsequent images in the stream, subtract the projection of the initial graph onto the shape base  $\mathbf{SS}^T\mathbf{X}^1$  from the subsequent graphs. Then stack the graphs as vectors and perform PCA to obtain the expression base vectors.

Note that in phase 2, the PCA is performed on the correlation matrix of the vectors, that is, we do not subtract a “mean expression” from the graphs.

The model can be described also in a slightly different way as the sum of two Gaussian distributions:

$$p(\mathbf{X}) = \mathbf{1} \otimes N(\mathbf{m}, \Sigma_{shape}) + N(0, \Sigma_{expression}), \quad (7)$$

where  $\Sigma_{shape}$  is the covariance matrix of the shape distribution and  $\Sigma_{expression}$  the correlation matrix of the expression distribution (with  $\mathbf{SS}^T\mathbf{X}^1$  removed). The eigenvectors of these matrices are the base vectors mentioned above.

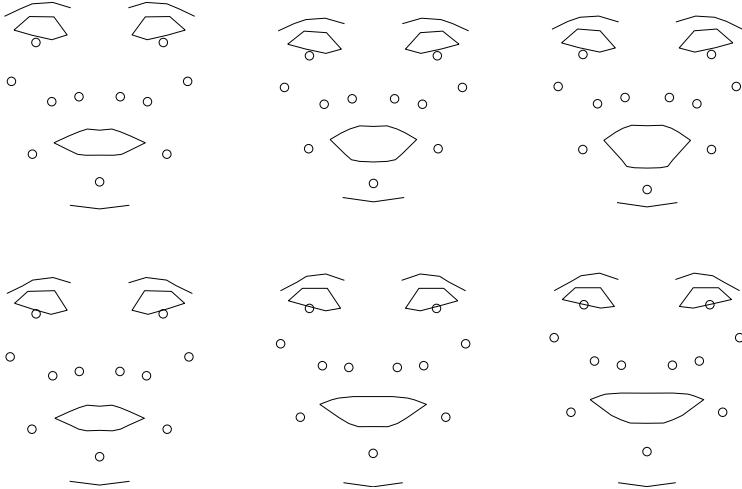
In practice we need to normalize our tracking results before they can be used to learn the model parameters. First we translate and scale the graphs so that their mean locations and their scale factors are the same. We define the scale factor as

$$s = \sqrt{0.5\sigma_x^2 + 0.5\sigma_y^2}, \quad (8)$$

where  $\sigma_x$  and  $\sigma_y$  are the standard deviations of the graph  $x$ - and  $y$ -coordinates. Then, to make the model symmetrical, we insert a mirrored replicate graph for every measured graph in the data. Finally, the lengths of the tracks are normalized by selecting a common frame number (larger than the length of the longest video sequence) and interpolating the tracks as necessary so that their lengths match.

## 4 Analysis and Reconstruction

To analyze the model and assess its capabilities, we performed a set of reconstruction-related tests. The shape and expression bases were computed using the measured tracking results and the principal components were inspected visually. The first two expression principal components are illustrated in Fig. 2. We then projected the measured tracks onto the obtained bases and analyzed the coordinates to see whether our separability assumption (person-dependent shape, person-independent expression) held. Some projection coordinate plots are shown in Fig. 3 and Fig. 4. It would seem that the separability assumption holds: the shape space coordinates remain in most cases approximately equal for the same person, while the expression space coordinates are similar for the same expression.

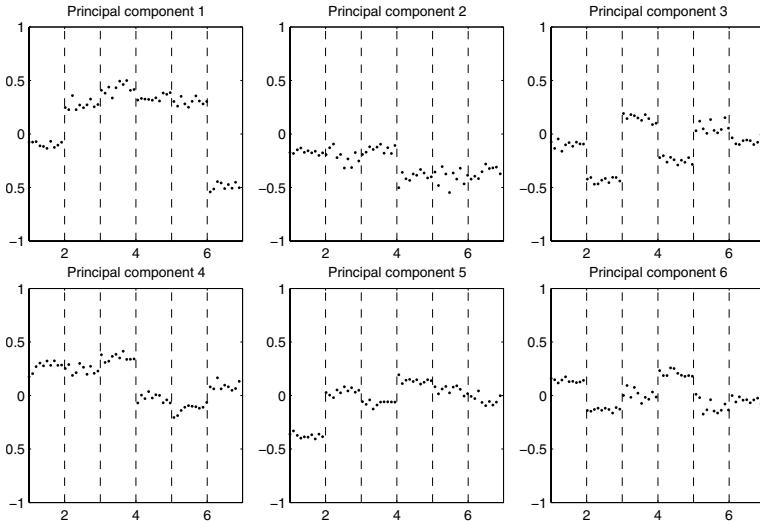


**Fig. 2.** The first two expression principal components. The components are shown at time steps  $t = 1$ ,  $t = 1/2t_f$  and  $t = t_f$ . The first component (row 1) is mainly related to opening of the mouth, while the second component (row 2) seems to be a smile

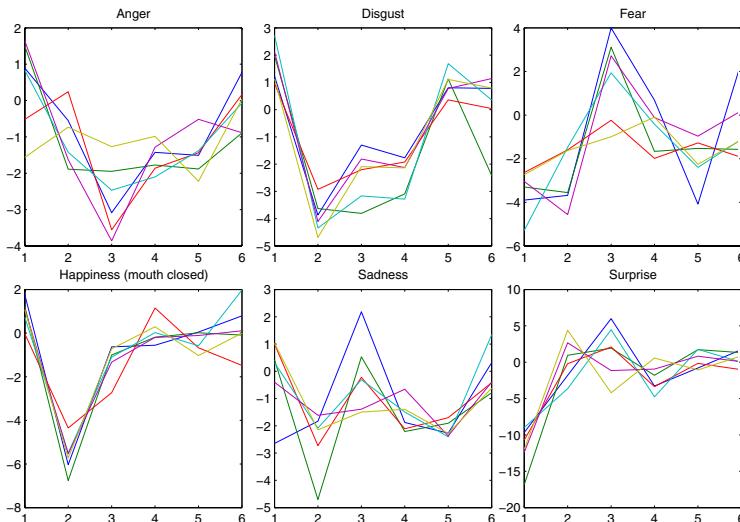
The actual reconstruction was done by projecting the measured tracks into the shape and expression spaces and then back to the original track space to obtain the reconstructed tracks  $\mathbf{X}^*$ ,

$$\mathbf{X}^* = \mathbf{1} \otimes (\mathbf{m} + \mathbf{SS}^T \mathbf{X}^1) + \mathbf{EE}^T \mathbf{X}. \quad (9)$$

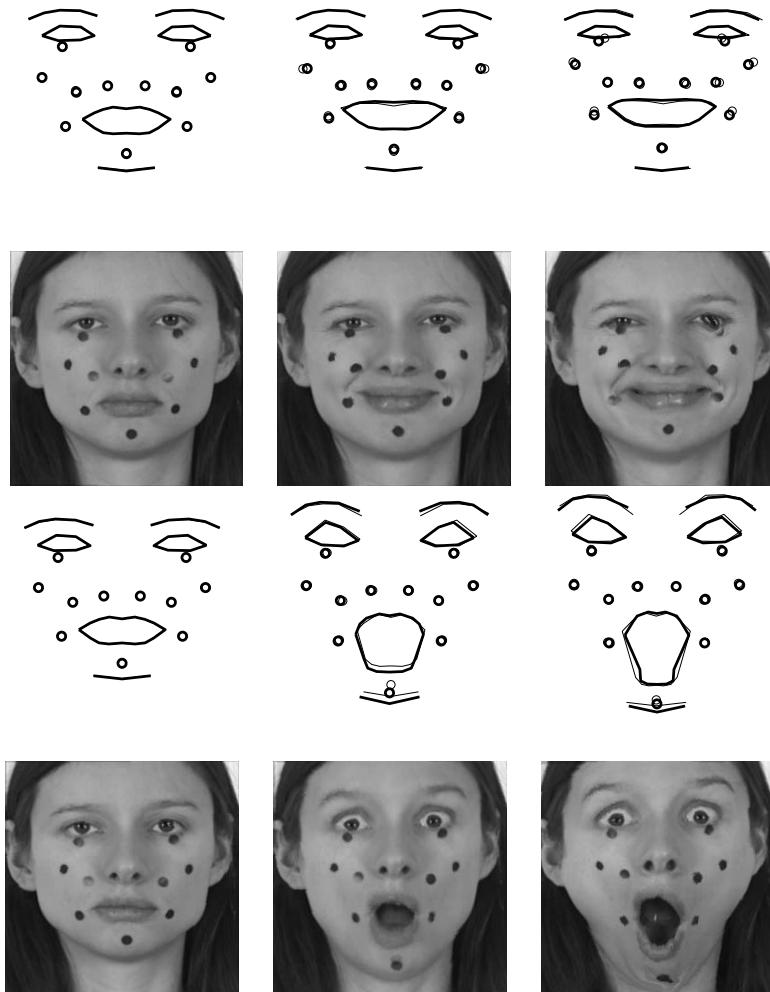
We used 15 principal components for the shape space and 6 components for the expression space, which in both cases amounted to ca. 99% of the total variance. The original and reconstructed tracks were compared both visually and numerically. Two sample reconstructions are shown in Fig. 5, and Table 2



**Fig. 3.** First six shape space coordinates for the 60 initial graphs  $\mathbf{X}^1$ . The x-axis is the person index from 1 to 6. Each image corresponds to a single principal component with 10 coordinate instances for each person. The dashed lines indicate change of person. In most cases, the persons are clearly distinct from one another, and the coordinates are similar for the same person



**Fig. 4.** First six expression space coordinates for the six basic expressions. The x-axis is principal component index. Each line corresponds to a single expression instance. The expressions are similar to each other across persons, although there are differences, too. For example, the coordinates for the expressions of happiness show more similarity than the expressions of fear. The similar situation is encountered in everyday life - expressions of happiness are much more alike than expressions of fear



**Fig. 5.** Reconstruction results for the “happiness (mouth closed)” (upper two rows) and “surprise” (lower two rows) expressions. The depicted time steps are  $t = 1$ ,  $t = 1/2t_f$  and  $t = t_f$ . The thinner graphs show the original data and the thicker graphs the reconstructed expressions, while the images show the results of morphing the video frame corresponding to the time step according to the reconstructed graph. The expressions are clearly recognizable, and there are few distortions

contains mean reconstruction errors per unit of scale as defined by the scale factor (8) (for the unscaled size  $256 \times 256$  training data the scale was around 50).

The reconstruction results are rather promising: visually, the reconstructed expressions are easily recognizable and contain little distortion, and the numerical errors are low (for the original data, the mean error is below 2 pixels for most cases).

**Table 2.** Mean Reconstruction Error per Unit of Scale

Expression	$t = 1$	$t = 1/2t_f$	$t = t_f$	$t = \{1...t_f\}$
Anger	0.0070	0.0267	0.0353	0.0214
Disgust	0.0071	0.0225	0.0296	0.0198
Fear	0.0082	0.0274	0.0353	0.0221
Happiness (mouth open)	0.0069	0.0250	0.0336	0.0208
Happiness (mouth closed)	0.0061	0.0246	0.0356	0.0212
Sadness	0.0073	0.0240	0.0311	0.0206
Surprise	0.0071	0.0265	0.0322	0.0229
Happiness + surprise	0.0072	0.0337	0.0411	0.0251
Happiness + disgust	0.0078	0.0282	0.0385	0.0246
Mouth opening	0.0063	0.0221	0.0258	0.0174
All expressions	0.0071	0.0261	0.0338	0.0216

## 5 Conclusion

We have presented a novel model for the representation of fiducial feature points on the human face. The model is a sum of two linear submodels: a person-dependent shape model and a person-independent expression model. The parameters of the model are learned from video data of trained actors making specified expressions. Our reconstruction results imply that the proposed separation of the facial graph as orthogonal shape and expression parts is feasible.

The model presented here is trained only on frontal facial images, and can not handle large pose variations. With 3D data it should be straightforward to extend the model to accommodate these. Also, there is considerable intrapersonal variation in facial expressions with regard to their strength and speed, whereas the current model assumes that expression durations and speeds are the same. This problem has to be addressed in further research.

The model has several practical applications. In its probabilistic form (7) the model can be used directly as a prior in expression-dependent Bayesian object matching [15]. Furthermore, in the future we will work on implementing the expressions on a Talking Head model [19]. The proposed model includes the dynamics of the expressions, and hence should be an improvement over the previously used expression model. Another interesting research topic is to compare the obtained expression principal components (Fig. 2) and the FACS action units to see whether there is any systematic correspondence.

## References

1. R. Adolphs, Recognizing emotion from facial expressions: psychological and neurological mechanisms, *Behavioral and Cognitive Neuroscience Reviews*, vol. 1, no. 1, 2002, pp. 21-62.
2. P. Ekman, W. Friesen, and J. Hager, *Facial Action Coding System*, Consulting Psychologists Press, 1978.

3. I.A. Essa and A.P. Pentland, Coding, analysis, interpretation and recognition of facial expressions, *IEEE TPAMI*, vol. 19, no. 7, 1997, pp. 757-763.
4. Y.-l. Tian, T. Kanade, and J.F. Cohn, Recognizing action units for facial expression analysis, *IEEE TPAMI*, vol. 23, no. 2, 2001, pp. 97-115.
5. G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, Classifying facial actions, *IEEE TPAMI*, vol. 21, no. 10, 1999, pp. 974-989.
6. P. Ekman, Expression and the nature of emotion, In: K. Scherer and P. Ekman, editors, *Approaches to Emotion*, Lawrence Erlbaum, 1984.
7. P. Ekman and W. Friesen, *Unmasking the Face. A Guide to Recognizing Emotions from Facial Expressions*, Consulting Psychologists Press, 1975.
8. B.D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, In: *Proc. Imaging Understanding Workshop*, 1981, pp. 121-130.
9. C. Tomasi and T. Kanade, Detection and tracking of feature points, *Carnegie Mellon University Technical Report CMU-CS-91-132*, 1991.
10. F. Bourel, C.C. Chibelushi, and A.A. Low, Robust Facial Feature Tracking, In: *Proc. BMVC 2000*, 2000.
11. J.G. Daugman, Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression, *IEEE TASSP*, vol. 36, no. 7, 1988, pp. 1169-1179.
12. A. Gelman, J.B. Carlin, H.S. Stern, and D.R. Rubin, *Bayesian Data Analysis*, 2nd edition, Chapman & Hall, 2004.
13. S.J. McKenna, S. Gong, R.P. Wurtz, J. Tanner, and D. Banin, Tracking facial feature points with Gabor wavelets and shape models, In: *Proc. 1st International Conference on Audio- and Video-based Biometric Person Authentication*, 1997.
14. L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, Face Recognition by Elastic Bunch Graph Matching, In: L.C. Jain, U. Halici, I. Hayashi, S.B. and Lee, editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, CRC Press, 1999.
15. T. Tamminen and J. Lampinen, Bayesian object matching with hierarchical priors and Markov chain Monte Carlo, In: J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West, editors, *Bayesian Statistics 7*, Oxford University Press, 2003.
16. B. Abboud and F. Davoine, Appearance factorization for facial expression analysis, In: *Proc. BMVC 2004*, 2004, pp. 507-516.
17. T.F. Cootes, G.J. Edwards, and C.J. Taylor, Active appearance models, *IEEE TPAMI*, vol. 23, no. 6, 2001, pp. 681-685.
18. C. Chatfield and A.J. Collins, *Introduction to Multivariate Analysis*, Chapman & Hall, 1995.
19. M. Frydrych, J. Kätsyri, M. Dobšík, and M. Sams, Toolkit for animation of Finnish talking head, In: *Proc. AVSP 2003*, 2003, pp. 199-204.

# Development of Direct Manipulation Interface for Collaborative VR/MR Workspace

Hiroshi Sasaki<sup>1</sup>, Tadayuki Takeda<sup>2</sup>, Masataka Imura<sup>2</sup>, Yoshihiro Yasumuro<sup>2</sup>,  
Yoshitsugu Manabe<sup>2</sup>, and Kunihiro Chihara<sup>2</sup>

<sup>1</sup> Information Science and Technology Center, Kobe University,  
1-1, Rokkodai, Nada, Kobe, Hyogo, 657-8501, Japan  
[sasaki@kobe-u.ac.jp](mailto:sasaki@kobe-u.ac.jp)  
<http://www.istc.kobe-u.ac.jp/>

<sup>2</sup> Graduate School of Information Science, Nara Institute of Science and Technology,  
8916-5, Takayama, Ikoma, Nara, 630-0192, Japan  
[{tadayu-t, imura, yasumuro, manabe, chihara}@is.naist.jp](mailto:{tadayu-t, imura, yasumuro, manabe, chihara}@is.naist.jp)  
<http://chihara.naist.jp/index.html>

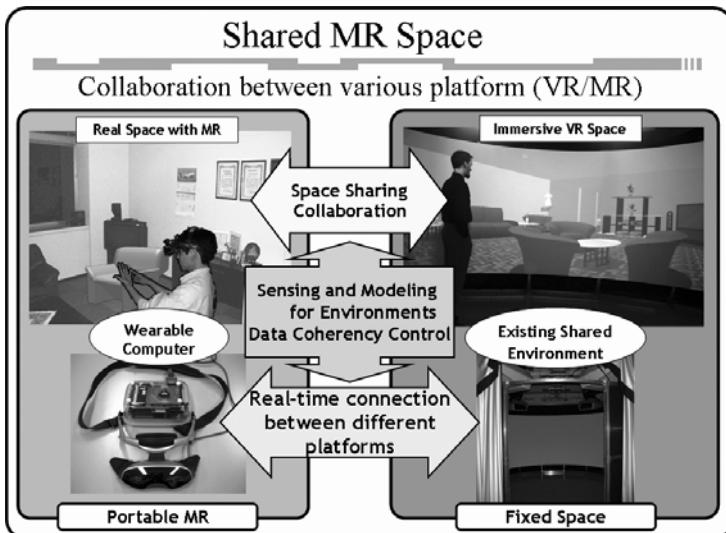
**Abstract.** Our research projects aim to connect various VR/MR platforms and to realize the seamless collaborative works with the intuitive operations whenever and wherever users like. In order to realize an ideal collaborative workspace, an effective handling scheme for both interactive virtual objects and system operation of VR/MR platform is needed. In this paper, we describe the three components, which are the direct manipulation interfaces for portable MR space and for fixed shared VR/MR space, and Interactive 3D Marker for displaying and manipulating virtual objects reflected lighting and geometry conditions of the real world. These components can give users seamless manipulation workspace by the natural operations such as they behave in their daily life. We also introduce Interactive Interior Design System using these components. This system realizes to coordinate the suitable interiors in the real space, by the free manipulation of virtual interior objects reflected the geometry and the lighting condition of the real space.

## 1 Introduction

There are many researches about the shared MR space for collaborative works such as conferencing, object modeling, layout planning and so on. However, most of the foregoing researches only connect two or few fixed sites with special multiple displays to provide immersive workspaces. Few user interfaces in the foregoing researches also allow users to manipulate MR space by the natural operations such as they behave in their daily life.

Our research projects aim to connect various VR/MR platforms as shown in Fig. 1 and to realize the seamless collaborative works with the intuitive operations whenever and wherever users like. In order to realize these spaces, it is need to develop two technologies: one is the portable MR workspace which users can carry out whenever they like, another is an effective handling scheme for both interactive virtual objects and operation of VR/MR platform.

In this paper, we describe the three components, which are the direct manipulation interfaces for portable MR space and for fixed shared VR/MR space, and Interactive 3D Marker for displaying and manipulating virtual objects reflected lighting and geometry conditions of the real world. These components can give users seamless manipulation environment. We also introduce Interactive Interior Design System using these components. This system realizes to coordinate the suitable interiors in the real space, by the free manipulation of virtual interior objects reflected the geometry and the lighting condition of the real space.



**Fig. 1.** Collaborative Works Between Various VR/MR Platforms

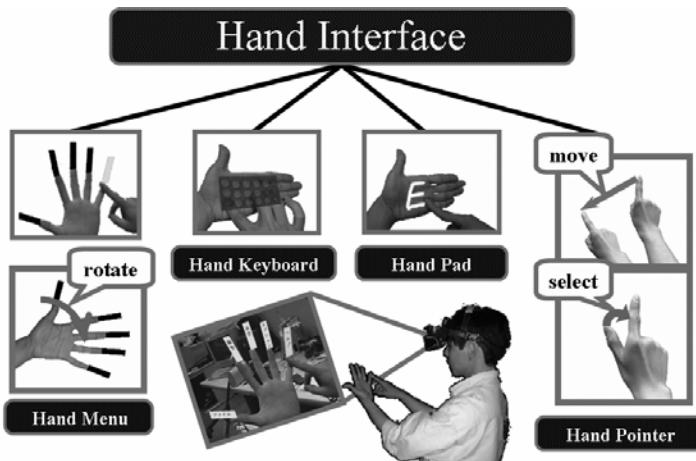
## 2 Direct Manipulative User Interface

### 2.1 Hand Menu for Hands-Free Operation in the Portable MR Space

In order to realize the portable MR spaces which gives users the MR workspace whenever and wherever they like, it is important to realize the easy and intuitive interface for MR space.

Hand Interface as shown in Fig. 2 is a non-contact and intuitive interface to realize an easy manipulation of a computer every time and everywhere. This interface only needs the user's hands and a camera attached with the head mounted display, without any additional device such as foregoing systems [1-4]. When the user performs some easy actions within the camera view, the user can operates the menu, selects some objects in the MR space, writes or draws, takes a picture, and so on.

Hand Menu is a menu system which is one module of Hand Interface. When the user opens his/her hand widely within the camera view, the system shows the user virtual menu items superimposed on his/her fingertips. The user can select a menu item



**Fig. 2.** Overview of Hand Interface

he/she wants to choose easily, by touching certain his/her fingertip superimposed a menu item with the index finger of another hand.

All of these processes are performed through the image recognition technique. Hand Menu recognizes the user operations by the following 6 steps:

1. hand-area extraction
2. fingertips detection
3. distinguishing the menu-hand superimposed menu items and the selection-hand to touch the menu
4. distinguishing the palm and the back of the menu-hand
5. displaying menu items
6. menu selection

At first, the system extracts hand-areas from obtained image through the camera using skin color information of the hand. In this hand-area, the system detects the fingertips utilizing a curvature through tracking the outline of the hand-area. The method to distinguish the menu-hand and the selection-hand is quite simple, using the number of detected fingertips. The system recognizes the region with five fingertips as the menu-hand and the region with one fingertip as the selection-hand. In order to distinguish the palm and the back, that is, whether the palm faces to the user's eye or not, the system examines the geometrical relation among the index, thumb and pinkie. As soon as the system recognizes that the menu-hand is exposed, the system superimposes the menu on each fingertip of the menu-hand in the user's sight through the see-through HMD. According to the orientation of the menu-hand, one of two menu sets is displayed. The system recognizes that the user selects a certain menu item, using each position and relation of the fingertips of the menu-hand and the selection-hand, and starts up the selected menu.

## 2.2 Direct Manipulation Interface for Fixed Immersive VR/MR Space

There are many researches how to manipulate the virtual objects and the VR/MR space [5-10]. However, it is difficult to allow users to manipulate them naturally just as if users manipulate the object of the real space in their daily life.

In order to realize the direct manipulation interface in the fixed immersive VR/MR environment, we are developing the free manipulation system using the infrared motion sensing system, as shown in Fig. 3.

This system allows users to move freely in the immersive environment and to manipulate the virtual objects naturally by the operations of touching, grasping, putting up, putting down, pointing and so on such as users behave in their daily life. The user has only to attach the Infrared Rays (IR) reflecting marker on the fixed points of his/her body and to enter this system.

When the user makes some action, the system obtains the 3D position of each IR maker from the obtained images through many IR cameras. Based on the 3D positions of the markers, the system assumes the 3D position of the user's body parts and recognizes how action he/she makes.

Here, we classify the user's action into 3 types:

1. whole action
2. local near action
3. local far action

Whole action is the action moving whole body such as going around in the immersive space, looking out and so on. In order to detect whole action, the user's position and direction in the immersive space is needed. So, we attach two IR markers to 3D eyeglass as shown in Fig. 4, and the system detects a middle point of the markers as the position of the user's head.

Local near action is the action when interacting virtual objects near the user, such as touching, grasping, putting up, putting down, pointing and so on. Local far action has similar aim as local near action. This is the action to manipulate virtual objects far from the user by combining pointing action and local near action. So, we attach IR markers to the hand for detecting local near action, elbow and shoulder for the pointing action as shown in Fig. 4.

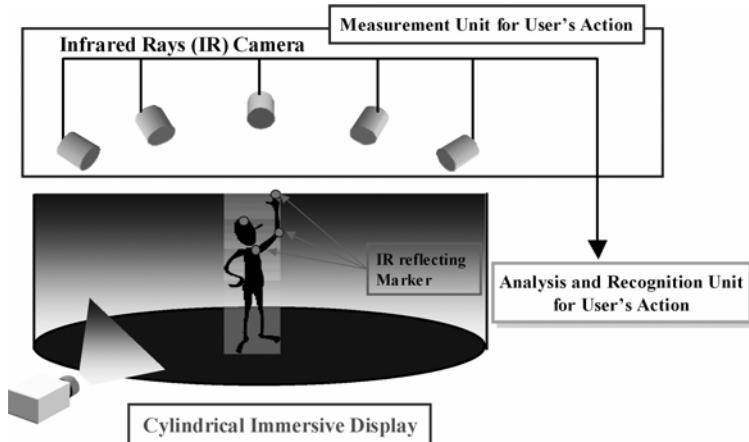
The system detects the detail hand's action from the marker with the hand, and the pointing point by assuming from the point of hand, elbow, shoulder and head. In order to estimate the pointing point, the system uses the reference point  $p$ ,  $q$  and  $r$ , as shown in Fig. 5. Reference point  $p$ ,  $q$  and  $r$  are the point on the line extended from head, elbow and wrist to the index fingertip. The system estimates the pointing point according to Eq. 1.

$$a\vec{p} + b\vec{q} + (1-a-b)\vec{r} = \vec{O} \quad (1)$$

Here, the vector  $O$  is the position vector of target. The  $a$  and  $b$  are weight parameters to define the pointing point, which are obtained in advance using known points.

When the user indicates some point, the system calculates the pointing point recursively as the following. First, the system defines the reference point  $p$  as initial value of interim point and obtains the weight in this point. The new interim point is calcu-

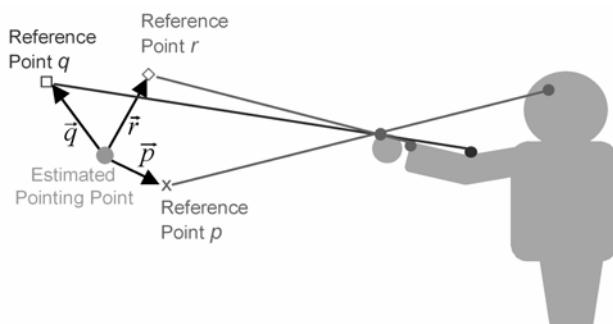
lated by the reference point  $p$ ,  $q$  and  $r$ , and the weight. This calculation is performed until the difference between the previous interim point and the new interim point becomes below a threshold value.



**Fig. 3.** System Overview



**Fig. 4.** Manipulation of the virtual objects by the natural motion



**Fig. 5.** Estimation of the Pointing Point

### 3 Interactive 3D Marker

Interactive 3D Marker realizes to acquire geometric and photometric information simultaneously with low processing cost. This marker consists of a cube with four 2D codes on the side and a spherical mirror on the top, as shown in Fig. 6.

Interactive 3D Marker can obtain its position and posture from 2D codes and lighting condition from a spherical mirror. So, when a user manipulates this marker directly such as moves or rotates it, a virtual object assigned to this marker can be displayed immediately according to the lighting and geometry conditions in the real space.

We assign a unique pattern to each 2D code to use it to detect which side is seen from the camera. In this paper, we used  $4 \times 4 = 16$  bits pattern, so that  $(2^{16} - 2^8) / (4 \times 4) = 4080$  kinds of virtual objects can be assigned for 16bits 2D code, considering excluding rotationally symmetric patterns.

Fig. 7 shows the process flow.

When a 2D code of Interactive 3D Marker is detected in the obtained image from the camera, the system calculates the geometrical information including its distance and posture from the appearance of the contour of the code region [11][12]. At the same time, the system acquires the embedded ID number by decoding the 2D code. Each ID number assigned to 2D code is associated with CG model. The CG models were defined as polygon models which contain thousands of vertices on which diffuse and specula reflection parameters were properly designed, in advance.

Continuously, the system calculates the lighting information from the spherical mirror of Interactive 3D Marker. The system assumes that lighting condition at the position of the 3D marker contains direct light from the light source, reflected light from specula surface in the environment and ambient light from distant view [13][14]. As shown in Fig. 8, when a spherical mirror is observed as a circle whose radius is  $r$  pixels and center position is  $(x_c, y_c)$ , normal vector  $N$  at the position  $(x, y)$  on the sphere surface observed in the spherical mirror image is expressed as Eq. 2

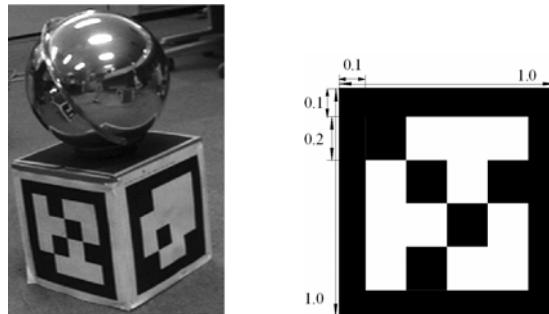
$$\begin{aligned}\vec{N} &= [e_x, e_y, e_z] \\ e_x &= \frac{(x - x_c)}{r}, \quad e_y = \frac{(y - y_c)}{r}, \quad e_z = \sqrt{1 - (e_x^2 + e_y^2)}\end{aligned}\tag{2}$$

Light coming through the focal point  $O$  into a pixel position  $(x, y)$  is emitted from a direction which is regular reflection line relative to normal vector  $N$  on the sphere. In the same way, the direction of the light source to illuminate each pixel on the image plane can be calculated.

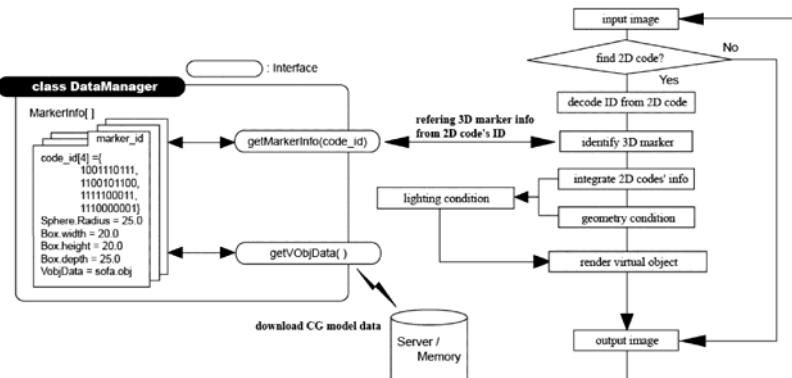
$$\vec{L} = \vec{V} - 2(\vec{N} \bullet \vec{V})\vec{N}\tag{3}$$

The image of the spherical mirror is divided into small regions and sampled the intensity of the light source which illuminates each region. The light intensity is calculated by averaging R, G and B value. The calculation of the direction of the light sources is used the centers of gravity of the regions.

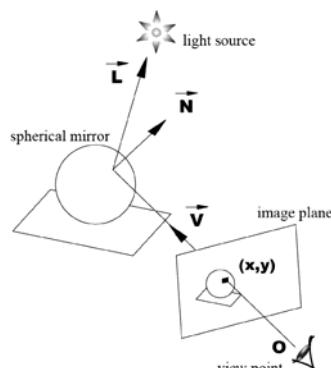
Finally, based on the obtained the geometry and lighting information, the system renders the CG models as fitting in the real space. The system assumes that the colors of the object surface are determined according to the model of Lambert and Phong for rendering virtual objects.



**Fig. 6.** Interactive 3D Marker (left) and 2D Code (right): digits stand for the size ratio



**Fig. 7.** Process Flow



**Fig. 8.** Schematic Geometry of Viewpoint and Light Source

## 4 Interactive Interior Design System

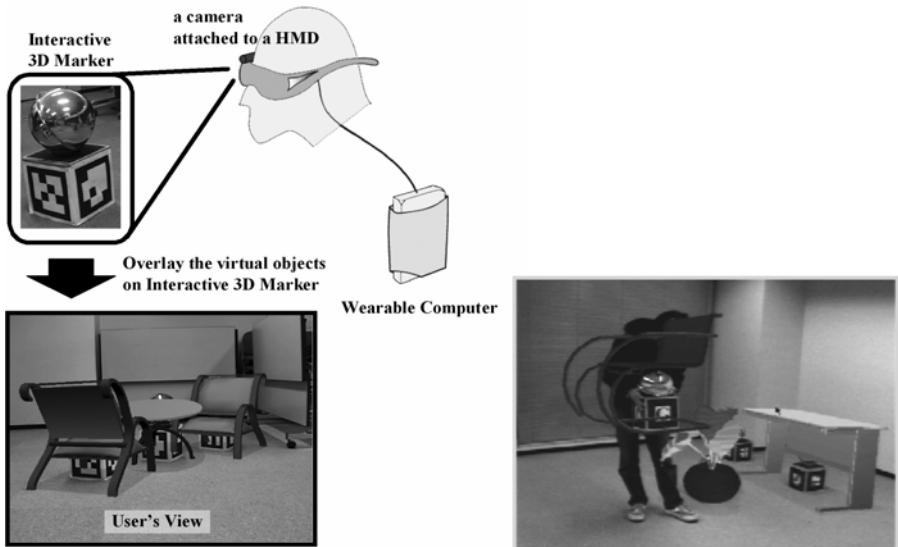
In this section, we describe the interior design system which is an application using the direct manipulative user interface and Interactive 3D Marker.

Fig. 9 shows the over view of the Interactive Interior Design System.

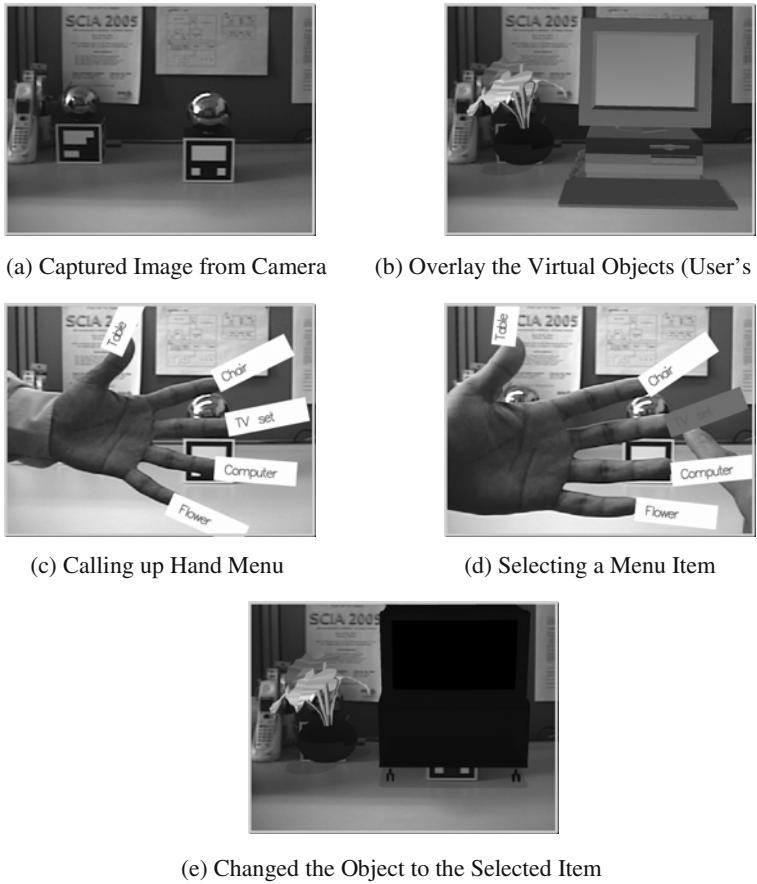
To use this system, the users have only to wear the head mounted display (HMD) attached a camera and a wearable computer as a processing unit. The users can make an interior design by putting and moving Interactive 3D Marker on the place they like freely. As each Interactive 3D Marker is assigned some interior object in advance, Interactive 3D Marker within the user's view through the HMD is displayed as the interior object which is a virtual object overlaid on Interactive 3D Marker according to the lighting and geometry condition of this marker in the real space.

The users can not only move Interactive 3D Marker but also reassigned Interactive 3D Marker from a certain interior object to other interior object. When the user opens his/her hand widely within the user's view, the system calls up Hand Menu, as shown in Fig. 10. Hand Menu has the menu items as the interior object's name or icons. As shown in Fig. 10, when the user selects a certain menu item by touching a menu item with his/her index finger, the system reassigned a selected object to Interactive 3D Marker on which his/her eyes were focused and redraw a new interior object immediately.

Therefore, this Interactive Interior Design System enables the user to try putting on and off the interiors reflected the lighting and geometry condition of the real world and to coordinate the suitable interiors in his/her space.



**Fig. 9.** Overview of Interactive Interior Design System



**Fig. 10.** Menu Operation with Hand Menu

## 5 Conclusions

In this paper, we described Direct Manipulation Interface and Interactive 3D Markers, as an effective handling scheme in MR space. Hand Menu, direct manipulation interface for the portable MR space, realized the hands-free operation by the intuitive actions. For the fixed immersive VR/MR environment, we developed the natural manipulative interface such as user manipulates in the real space. Interactive 3D Marker realized to obtain the geometric and photometric information of the place put it on and to manipulate virtual objects freely and directly.

Furthermore, we described Interactive Interior Design System which was an application using Hand Menu and Interactive 3D Marker. This system realized to coordinate the suitable interiors in the real space, by the free manipulation of virtual interior objects reflected the geometry and the lighting condition of the real space and the hands-free menu operations. This system allowed several users to collaborate in the interior design wherever they wanted to make the interior design. This system gave the user seamless operations in the MR Space.

## Acknowledgments

This research is partly supported by Core Research for Evolutional Science and Technology (CREST) Program “Advanced Media Technology for Everyday Living” of Japan Science and Technology Agency (JST).

## References

1. Masaaki Fukumoto, Yoshinobu Tonomura, Body Coupled FingeRing: Wireless Wearable Keyboard, Proceeding of the ACM Conference on Human Factors in Computing Systems, CHI97, pp. 147-154,1997.
2. Jun Rekimoto, GestureWrist and Gesture Pad: UnobtrusiveWearable Interaction Devices, Proceeding of the 5th International Symposium on Wearable Computers, pp. 21-27, 2001.
3. Koji Tsukada and Michiaki Yasumura: Ubi-Finger: Gesture Input Device for Mobile Use, Proceedings of APCHI 2002, Vol.1, pp.388-400, 2002.
4. Virtual Technologies, CyberGlove.
5. P.Hanrahan, L.D.Culter, B.Frolich: Two-handed Direct Manipulation on the Responsive Workbench, Symposium on Interactive 3D Graphics, pp.107-114, 1997.
6. T.Ilmonen: Immersive 3D User Interface for Computer Animation Control, International Conference of Computer Vision and Graphics, 2002.
7. I.Cohen and M.W.Lee: 3D Body Reconstruction for Immersive Interaction, 2nd International Workshop on Articulated Motion and Deformable Objects, pp.119-130, 2002.
8. S.Rougeaux, R.G.O'Hagan, A.Zelinsky: Visual Gesture Interface for Virtual Environments, User Interface Conference, pp.73-80, 2002.
9. D.Minnen, T.Westyn, A.Hurst, T.Starner, B.Leibe and J.Weeks: The Perceptive Workbench: Computer-vision-based Gesture Tracking, Object Tracking, and 3D Reconstruction for Augmented Desks, Machine Vision and Application, Vol.14, pp.59-71, 2003.
10. G.Wesche: The toolfinger: Supporting Complex Direct Manipulation in Virtual Environments, Proceedings of the workshop on Virtual Environments 2003 (EGVE '03), pp.39-45, 2003.
11. H.Kato and M.Billinghurst, Marker tracking and HMD calibration for a video based augmented reality conferencing system, IEEE International Workshop on Augmented Reality, pp. 125-133, 1999.
12. Diego Lopez de Ipi˜na, Paulo Mendonca, and Andy Hopper : “a low-cost visionbased location system for ubiquitous computing”, Personal and Ubiquitous Computing, Vol. 6, No. 3, pp. 206-219, 2002.
13. Paul Debevec, Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography, Proceedings of SIGGRAPH 98, pp. 189-198, 1998.
14. I.Sato, Y.Sato, and I.Ikeuchi, Acquiring a radiance distribution to superimpose virtual objects onto a real scene, IEEE Transactions on Visualization and Computer Graphics, Vol. 5, No. 1, pp. 1-12, 1999.

# Estimating Camera Position and Posture by Using Feature Landmark Database

Motoko Oe<sup>1</sup>, Tomokazu Sato<sup>2</sup>, and Naokazu Yokoya<sup>2</sup>

<sup>1</sup> IBM Japan

<sup>2</sup> Nara Institute of Science and Technology, Japan

**Abstract.** Estimating camera position and posture can be applied to the fields of augmented reality and robot navigation. In these fields, to obtain absolute position and posture of the camera, sensor-based methods using GPS and magnetic sensors and vision-based methods using input images from the camera have been investigated. However, sensor-based methods are difficult to synchronize the camera and sensors accurately, and usable environments are limited according to selection of sensors. On the other hand, vision-based methods need to allocate many artificial markers otherwise an estimation error will accumulate. Thus, it is difficult to use such methods in large and natural environments. This paper proposes a vision-based camera position and posture estimation method for large environments, which does not require sensors and artificial markers by detecting natural feature points from image sequences taken beforehand and using them as landmarks.

## 1 Introduction

The recovery of camera position and posture is required in a number of different fields such as augmented reality and robot navigation. In these fields, to obtain absolute position and posture of the camera, sensor-based methods using GPS and magnetic sensors[1, 2, 3, 4, 5] and vision-based methods using input images from the camera[6, 7, 8, 9, 10, 11, 12, 13] have been investigated. However, sensor-based methods are difficult to synchronize the camera and sensors accurately, and usable environments are limited according to selection of sensors. Vision-based methods can be classified in two groups: Methods using markers and methods without markers. Methods using markers need to allocate many artificial markers in the environment. Thus, it is difficult to use such methods in large and natural environments. On the other hand, marker-less methods are also proposed. Most of these methods track natural features and estimate camera position and posture by concatenating transformations between adjacent frames. Thus, these methods are inappropriate for long sequences because estimation errors accumulate and causes drift. Therefore, methods using prior knowledge of the environment are recently proposed[12, 13]. Lepetit et al.[12] have proposed a method using the 3-D model of the environment. It is robust to large camera displacements, extreme aspect changes and partial occlusions, but

their method is limited to an environment that can be modeled manually, so it is difficult to use in an outdoor environment. Gordon et al.[13] have proposed a method which constructs a sparse metric model of the environment, and performs model-based camera tracking. It does not require camera pre-calibration nor prior knowledge of scene geometry, but it is difficult to use the method in large environments because the error will accumulate when reconstructing the 3-D model.

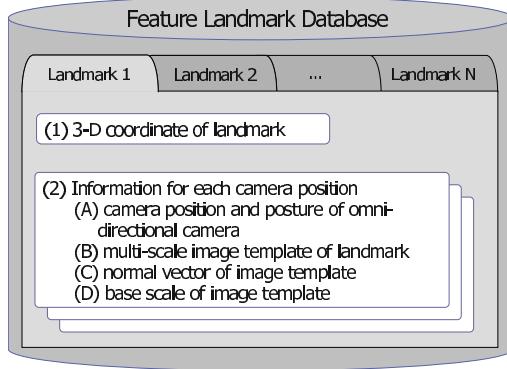
In this research, we propose a camera position and posture estimation method for large environments, which is based on detecting natural feature points from image sequence taken beforehand and using them as landmarks, and thus does not require sensors and artificial markers. Our method is composed of two stages. In the first offline stage, we reconstruct the environment from omni-directional image sequences. Then, a feature landmark database is created, and natural feature points extracted from the image sequences are registered as landmarks. The second stage is a sequential process, and camera position and posture which do not include significant cumulative errors are estimated by determining the correspondence between the input image and the landmarks. In Section 2, we describe the first stage of our method, which specifies the construction of the feature landmark database. Section 3 describes the position and posture estimation method using the feature landmark database created in Section 2. Section 4 shows the experiment result, and conclusion is shown in Section 5.

## 2 Constructing Feature Landmark Database

This section describes the first stage of our method, which specifies the construction of the feature landmark database. In our method, natural feature points detected from omni-directional image sequences are used as landmarks. We first take an omni-directional image sequence by walking through the environment with an omni-directional camera. Secondly, we obtain 3-D coordinates of landmarks and camera position and posture of the omni-directional camera from the image sequence. Lastly, the landmark database is created semi-automatically using the 3-D coordinates of the natural features, the omni-directional images, and its camera path. In the following sections, the elements of the feature landmark database are listed, and the way for constructing landmark database is detailed.

### 2.1 Landmark Information Acquisition by 3-D Reconstruction of Environment

Feature landmark database consists of a number of landmarks as shown in Figure 1. These landmarks are used to be matched to natural feature points from an input image in the second stage in order to estimate the camera position and posture of an input image. Each landmark retains the 3-D coordinate of itself(1), and several information for different camera positions(2). Information for different camera positions consists of four items: (A)camera position and posture of the omni-directional camera, (B)multi-scale image template of the



**Fig. 1.** Elements of feature landmark database

landmark, (C)normal vector of the image template, and (D)base scale of the image template.

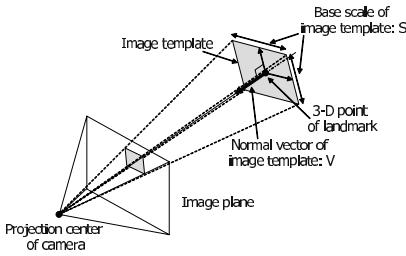
To obtain the landmark information listed above (Figure 1), 3-D reconstruction of the environment is required. First, we reconstruct the environment from omni-directional image sequence and obtain (1)3-D coordinate of the landmark and (A)camera position and posture of the omni-directional camera. Next, we generate (C)normal vector of the image template, (D)base scale of the image template, and (B)multi-scale image template of the landmark.

**3-D Reconstruction of the Environment from Omni-directional Image Sequence.** Our extrinsic camera parameter estimation is based on structure-from-motion[14]. In this method, first, markers and natural features in the image sequences captured by an omni-directional multi-camera system are automatically tracked and then the reprojection errors are minimized throughout the sequences. Thus, we can obtain extrinsic camera parameter of the camera system and 3-D coordinates of natural features in absolute coordinate system based on the markers without accumulative estimation errors, even in a large and complex environment. Note that, in our method, intrinsic camera parameters of the camera system are assumed to be known.

**Creating Landmarks.** Landmark database is automatically created using the result of 3-D reconstruction. The elements of landmarks are created by the following procedures.

### (1) 3-D coordinate of landmark

We use natural features detected by Harris operator[15] from the omni-directional image as feature landmarks. The 3-D coordinate of the landmark is estimated by the 3-D reconstruction of the environment, and is obtained by the world coordinate system. The X and Y axes of the world coordinate system are aligned to the ground and Z axis is vertical to the ground.



**Fig. 2.** Landmark and its image template

## (2) Information for each camera position

Landmarks are used to determine the correspondence between feature points in an input image and 3-D coordinates of the landmarks. In this research, information from several different camera positions is obtained and used for a robust matching of the landmarks, considering the aspect changes of image patterns depending on the shooting position.

**(A) Camera position and posture of omni-directional camera:** Camera position and posture are retained by the world coordinate system, and are used to select landmarks from the database to match with the input image. We use the extrinsic camera parameter estimated in Section 2.1.

**(B) Multi-scale image template of landmark:** Image template is created by projecting the omni-directional image to a plane which is vertical to the line through the landmark's 3-D coordinate and the projection center of the camera, as shown in Figure 2. The lens distortion is removed from the image template. First, the normal vector  $V$  and the base scale  $S$  shown in Figure 2 are precalculated. Then, to create an image template of a base scale, a square plane which implements the following assumptions is configured.

- Landmark is allocated on the center of the plane
- The plane is vertical to the normal vector  $V$
- The plane size is  $S \times S$  in the world coordinate system
- The plane's X axis is parallel to the X-Y axis of the world coordinate system

Next, the previously defined plane is divided into an  $N \times N$  grid where  $N \times N$  is the resolution of image templates. Each center of the grid is projected to the omni-directional images by its 3-D coordinate, and the color value of the projected pixel is set as the template's pixel color value. In the same way, double and quadruple scale image templates are created for each camera position. We define single, double, and quadruple scale image templates as a set of multi-scale image templates.

**(C) Normal vector of image template:** As shown in Figure 2, the normal vector of the image template is defined as the normal vector of the plane which is vertical to the line through the landmark's 3-D coordinate and

the omni-directional camera's position. It is used to select an image template for matching from several image templates taken by different camera positions. Normal vector of the image template is simply acquired as a normalized vector from the landmark's 3-D coordinate to the omni-directional camera's position.

- (D) **Base scale of image template:** As shown in Figure 2, the scale of the image template is the size of the plane used to create the image template. The scale size is retained in the world coordinate system, and the base scale is determined so that the resolution of the omni-directional image and the image template becomes nearly equal.

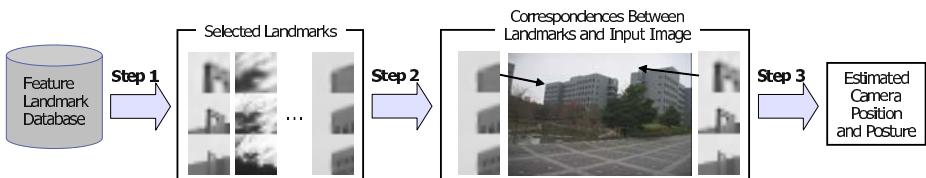
### 3 Camera Position and Posture Estimation Using Database

#### 3.1 An Overview of Proposing Method

This section describes a camera position and posture estimation method based on the feature landmark database. The initial camera position and posture are estimated in the first frame. Then, using the previous camera position and posture, landmarks are selected from the landmark database(step 1). Detecting natural features from the input image and matching them with the landmark image templates, the correspondence between landmark and input image is established(step 2). Lastly, camera position and posture are estimated from the correspondences between landmarks and input image(step 3). In this paper, we assume that initial camera position and posture are given. The following sections describe these steps. Figure 3 shows the data flow of the estimation process.

#### 3.2 Selecting Landmark from Landmark Database

To find a correspondence with the input image, several landmarks are selected from numerous landmarks in the landmark database. Furthermore, to handle partial occlusions and aspect changes, an image template with the nearest appearance to the input image is chosen from a number of image templates stored in the database. Considering the appearance, it is ideal if the image template and input image are taken in the same position. However, the camera position and posture of the input image are not yet estimated, so we use the camera



**Fig. 3.** Data Flow of Camera Position and Posture Estimation Process

position and posture of the previous frame as a replacement. Landmarks satisfying the following requirements are selected to make correspondence with the input image.

**(requirement 1)** Landmark has to be in the image when projecting its 3-D coordinate using the previous camera position and posture: We project the landmark's 3-D coordinate on the input image by using previous camera position and posture. Only the landmarks projected on the input image are selected.

**(requirement 2)** Distance between the camera position when the landmark was taken and the camera position when the input image was taken should be under a given threshold: We actually calculate the distance between the camera position when the landmark was taken and the camera position of the previous frame, and select landmarks under the threshold.

**(requirement 3)** Angle between the normal vector of the image template and the vector from landmark to camera position when the input image was taken should be under a given threshold and is the minimum for all the image templates of the landmark: We select the image template if angle  $\theta$  between the normal vector of the image template and the vector from landmark to previous camera position is the minimum for all the image templates of the same landmark. If the angle  $\theta$  of the selected image template is over the threshold, that landmark is not selected.

**(requirement 4)** Landmark must not be adjacent to already selected landmarks: First, the input image is divided into a grid. The landmarks on the input image are then projected to the image plane by using the previous camera position and posture, and only one landmark per each grid are selected.

Landmarks that implement the requirement 1 are selected first. Then, the selected landmarks are narrowed down to a fixed number of landmarks by the ascending order of the distance mentioned in the requirement 2. From the list of landmarks, landmarks with smaller angles in the requirement 3 are picked up one by one, and the selecting process is repeated until a fixed number of landmarks that implement the requirement 4 are chosen.

### 3.3 Determining Correspondence Between Landmark and Input Image Feature

In this step, the correspondence between selected landmarks and features in an input image are computed. First, natural features are detected from the input image using interest operator, and are then corresponded with the selected landmarks using template matching.

**Detecting Natural Feature from Input Image.** To find the correspondence between landmarks and input image, natural feature points are detected from the input image by Harris operator[15]. In this step, a landmark is projected to the input image, using previous camera position and posture. On the assumption that the corresponding point for the landmark exists near the projected point,

natural feature points are detected within a fixed window surrounding the projected point. The detected feature points are listed as correspondence candidates of the landmark.

**Matching Between Landmark Image Template and Input Image.** In this step, each landmark is compared with its correspondence candidates. First, an image pattern is created for each natural feature point listed as a correspondence candidate. Next, the landmark image template is compared with each image pattern by normalized cross correlation. Then, the feature point with the most correlative image pattern is selected, and its neighboring pixels are also compared with the landmark as correspondence candidates. Lastly, the most correlative feature point is corresponded with the landmark.

### 3.4 Camera Position and Posture Estimation Based on Established Correspondences

Camera position and posture are estimated from the list of 2-D and 3-D correspondences acquired from the matching between landmarks and input image. First, outliers are eliminated by RANSAC[16]. Next, camera position and posture are estimated using only the correspondences that are supposed to be correct. Finally, camera position and posture with the minimum reprojection error are computed by using non-linear least square minimization method.

## 4 Experiments

To verify the validity of the proposed method, we actually have created a landmark database of an outdoor environment and have carried out experiments of estimating camera position and posture from an outdoor image sequence.

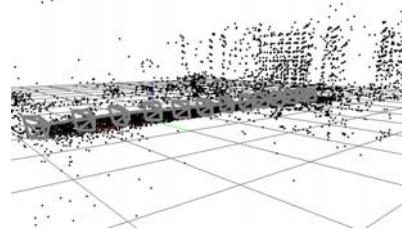
### 4.1 Experiments in an Outdoor Environment

First, an outdoor image sequence is captured by an omni-directional multi-camera system(Point Grey Research Ladybug) as shown in Figure 4 for constructing a landmark database. In this experiment, intrinsic parameters of the camera system was calibrated by Ikeda's method in advance[17]. Captured image sequence consists of 1,250 frames long with 6 images per each frame(totally 7,500 images). Then, landmark database is created by estimating camera path and 3-D coordinates of natural features[14]. For every landmark, multi-scale image template with three different scales of  $15 \times 15$  pixels each, is created per each camera position. The number of landmarks created in this experiment is about 12,400, and the number of image templates created per each landmark is 8 on average. Figure 5 shows a part of estimated camera path and 3-D coordinates of natural feature points in constructing the landmark database.

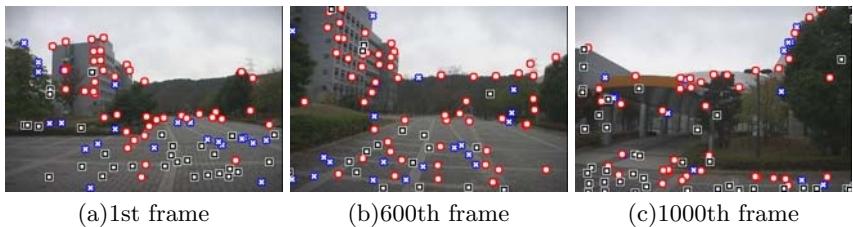
Next, we have captured a 1,000 frames long monocular video image sequence( $720 \times 480$  pixels, progressive scan, 15fps) with a video camera(SONY



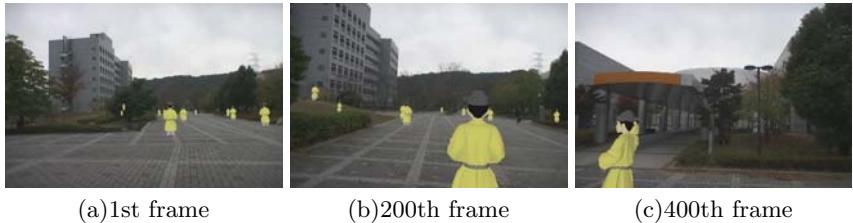
**Fig. 4.** Omni-directional camera system Ladybug and images taken by ladybug



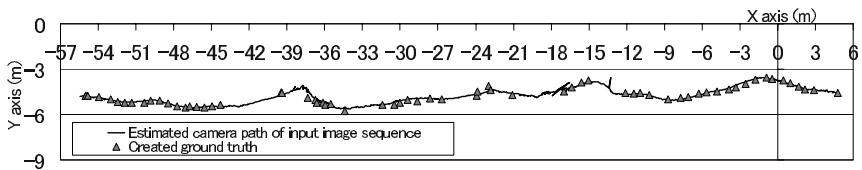
**Fig. 5.** Estimated camera path and 3-D coordinates of natural feature points



**Fig. 6.** Landmarks used for camera position and posture estimation



**Fig. 7.** Match move based on estimated camera position and posture  
 $(http://yokoya.naist.jp/pub/movie/oe/outdoor.mpg)$



**Fig. 8.** Estimated camera path and the ground truth

DSR-PD-150) and camera position and posture are sequentially estimated using the landmark constructed earlier. In this experiment, initial position and posture of the camera is manually specified in the first frame of the input sequence. The

maximum number of landmarks selected from the database to correspond with input image is 100 per frame, with the window size for detecting natural features from input image is  $120 \times 60$  pixels, the number of RANSAC iterations is 500. As a result, processing time for a frame was about 4 seconds with a PC(CPU Pentium4 3GHz, Memory 1.5GB).

Figure 6 shows the landmarks used for camera position and posture estimation. In this figure, squares indicate feature landmarks rejected by similarity measure, crosses are also rejected by RANSAC, and circles are inliers of feature landmarks. The inliers are used for camera position and posture estimation. Figure 7 shows the result of match move; matching virtual 3-D objects to the camera movements using the estimated camera position and posture. It can be observed that the CG person drawn in geometrically correct positions throughout the sequence(<http://yokoya.naist.jp/pub/movie/oe/outdoor.mpg>).

## 4.2 Quantitative Evaluation

We have evaluated the estimation accuracy by comparing the estimated camera position and posture with the ground truth. The ground truth is created by measuring 3-D position of feature points using a 3-D laser measure named "Total Station" and manually specifying their correspondence with each input image, and solving PnP(Perspective n-Point) problem from the correspondence. The ground truth is created for every 10 frames, except for the following frames: frames which could not obtain enough measured points because the scene is interspaced with natural objects, and frames in which the reprojection error of the obtained ground truth is over 1.5 pixels.

As a result, camera position estimation error was 220mm on average, and estimation error of the optical axis was approximately 0.37 degrees. Figure 8 shows the result of estimated camera parameter and the ground truth. Camera path is estimated from 1,000 frames long image sequence, and the X and Y axes of the figure corresponds to the X and Y axes of the world coordinate system. It shows that the estimated camera path is generally smooth and the estimation error does not accumulate during the whole sequence. However, there were some frames with larger estimation errors than other frames. In these frames, landmarks used for camera position and posture estimation are tended to be aligned lopsidedly in the input image, or only the landmarks far from the camera position are used. Therefore, it is necessary to investigate a method for selecting landmarks from the landmark database to raise the accuracy of our method.

## 5 Conclusion

In this paper, we have proposed a camera position and posture estimation method for large environments by detecting natural feature points from image sequence taken beforehand and using them as landmarks. The proposed method provides image-based localization. We create a feature landmark database by reconstructing the environment from image sequences in advance. Camera posi-

tion and posture are estimated by determining the correspondence between the input image and the landmarks. In experiments, we have successfully demonstrated camera position and posture estimation from an image sequence of an outdoor environment, and have confirmed that the estimation result does not include cumulative errors. As a future work, camera position and posture estimation needs to be performed in real-time for use in augmented reality applications and robot navigations. It is also desirable that the camera's initial position and posture are estimated automatically.

## References

1. T. Höllerer, S. Feiner and J. Pavlik: "Situated documentaries: Embedding multi-media presentations in the real world," Proc. Int. Symp. on Wearable Computers, pp. 79–86, 1999.
2. T. H. S. Feiner, B. MacIntyre and A. Webster: "A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment," Proc. Int. Symp. on Wearable Computers, pp. 74–81, 1997.
3. R. Tenmoku, M. Kanbara and N. Yokoya: "A wearable augmented reality system using positioning infrastructures and a pedometer," Proc. IEEE Int. Symp. on Wearable Computers, pp. 110–117, 2003.
4. D. Hallaway, T. Höllerer and S. Feiner: "Coarse, inexpensive, infrared tracking for wearable computing," Proc. IEEE Int. Symp. on Wearable Computers, pp. 69–78, 2003.
5. G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller and D. Colucci: "High-performance wide-area optical tracking -the hiball tracking system," Presence: Teleoperators and Virtual Environments, Vol. 10, No. 1, pp. 1–21, 2001.
6. H. Kato and H. Billinghurst: "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," Proc. IEEE/ACM Int. Workshop on Augmented Reality, pp. 85–94, 1999.
7. U. Neumann and S. You: "Natural feature tracking for augmented-reality," IEEE Transactions on Multimedia, Vol. 1, No. 1, pp. 53–64, 1999.
8. A. J. Davison, Y. G. Cid and N. Kita: "Real-time 3d slam with wide-angle vision," Proc. IFAC Symp. on Intelligent Autonomous Vehicles, 2004.
9. L. Naimark and E. Foxlin: "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," Proc. IEEE/ACM Int. Symp. on Mixed and Augmented Reality, pp. 27–36, 2002.
10. C. Tomasi and T. Kanade: "Shape and motion from image streams under orthography: A factorization method," Int. J. of Computer Vision, Vol. 9, No. 2, pp. 137–154, 1992.
11. P. Beardsley, A. Zisserman and D. Murray: "Sequential updating of projective and affine structure from motion," Int. J. of Computer Vision, Vol. 23, No. 3, pp. 235–259, 1997.
12. V. Lepetit, L. Vacchetti, D. Thalmann and P. Fua: "Fully automated and stable registration for augmented reality applications," Proc. Int. Symp. on Mixed and Augmented Reality, pp. 93–102, 2003.
13. I. Gordon and D. G. Lowe: "Scene modelling, recognition and tracking with invariant image features," Proc. Int. Symp. on Mixed and Augmented Reality , pp. 110–119, 2004.

14. T. Sato, S. Ikeda and N. Yokoya: "Extrinsic camera parameter recovery from multiple image sequences captured by an omni-directional multi-camera system," Proc. European Conf. on Computer Vision, Vol. 2, pp. 326–340, 2004.
15. C. Harris and M. Stephens: "A combined corner and edge detector," Proc. Alvey Vision Conf., pp. 147–151, 1988.
16. M. A. Fischler and R. C. Bolles: "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Comm. of the ACM, Vol. 24, pp. 381–395, 1981.
17. S. Ikeda, T. Sato and N. Yokoya: "A calibration method for an omnidirectional multi-camera system," Proc. SPIE Electronic Imaging, Vol. 5006, pp. 499–507, 2003.

# Geometrical Computer Vision from Chasles to Today

K. Åström

Centre For Mathematical Sciences, Lund University, Lund, Sweden  
`kalle@maths.lth.se`

**Abstract.** In this talk I will present geometrical computer vision from its early beginnings in projective geometry and photogrammetry to new research on methods in algebraic geometry. In the talk I will give examples of theory for minimal structure and motion problems (central and non-central cameras, 1D and 2D retina), critical configurations and geometry in general as well as practical results of using such theory in 3D reconstruction, navigation, modelling and image interpretation.

## References

1. K. Åström and F. Kahl. Motion estimation in image sequences using the deformation of apparent contours. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(2):114–127, 1999.
2. K. Åström and F. Kahl. Ambiguous configurations for the 1d structure and motion problem. *Journal of Mathematical Imaging and Vision*, 18(2):191–203, 2003.
3. K. Åström and M. Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. *Journal of Mathematical Imaging and Vision*, 12(2):121–135, 2000.
4. Rikard Berthilsson, Kalle Åström, and Anders Heyden. Reconstruction of curves in  $r^3$  using factorization and bundle adjustment. *Int. Journal of Computer Vision*, 41(3):171–182, 2001.
5. T. Buchanan. Photogrammetry and projective geometry - an historical survey. *SPIE*, 1944:82–91, 1993.
6. M. Chasles. Question 296. *Nouv. Ann. Math.*, 14(50), 1855.
7. A. Heyden. Reconstruction from image sequences by means of relative depths. *Int. Journal of Computer Vision*, 24(2):155–161, September 1997. also in Proc. of the 5th International Conference on Computer Vision, IEEE Computer Society Press, pp. 1058–1063.
8. E. Kruppa. Zur Ermittlung eines Objektes Zwei Perspektiven mit innerer Orientierung. *Sitz-Ber. Akad. Wiss., Wien, math. naturw. Kl. Abt, IIa*(122):1939–1948, 1913.
9. D. Nistér. An efficient solution to the five-point relative pose problem. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 195–202. IEEE Computer Society Press, 2003.
10. D. Nistér and F. Schaffalitzky. What do four points in two calibrated images tell us about the epipoles? In *Proc. 8th European Conf. on Computer Vision, Prague, Czech Republic*, 2004.

11. H Schubert. Die trilineare beziehung zwischen drei einstufigen grundgebilden. *Mathematische Annalen*, 17, 1880.
12. A. Shashua. Trilinearity in visual recognition by alignment. In *Proc. 3rd European Conf. on Computer Vision, Stockholm, Sweden*, pages 479–484, 1994.
13. P. Stefanovic. Relative orientation - a new approach. *ITC Journal*, 3:417–448, 1973.
14. H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, 2005.
15. H. Stewénius, F. Kahl, D. Nistér, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Proc. Conf. Computer Vision and Pattern Recognition, San Diego, USA*, 2005.
16. B. Triggs. Matching constraints and the joint image. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 338–343, 1995.
17. B. Vijayakumar, D. Kriegman, and J. Ponce. Structure and motion of curved 3D objects from monocular silhouettes. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 327–334, 1996.

# The S-Kernel and a Symmetry Measure Based on Correlation

Bertrand Zavidovique<sup>1</sup> and Vito Di Gesù<sup>1,2</sup>

<sup>1</sup> IEF, University of Paris XI, ORSAY, France

<sup>2</sup> DMA, Università di Palermo, Italy

[digesu@math.unipa.it](mailto:digesu@math.unipa.it), [zavido@ief.u-psud.fr](mailto:zavido@ief.u-psud.fr)

**Abstract.** Symmetry is an important feature in vision. Several detectors or transforms have been proposed. In this paper we concentrate on a measure of symmetry. Given a transform  $S$ , the kernel  $SK$  of a pattern is defined as the maximal included symmetric sub-set of this pattern. The maximum being taken over all directions, the problem arises to know which center to use. Then the optimal direction triggers the shift problem too. We prove that, in any direction, the optimal axis corresponds to the maximal correlation of a pattern with its flipped version. That leads to an efficient algorithm. As for the measure we compute a modified difference between respective surfaces of a pattern and its kernel. A series of experiments supports actual algorithm validation.

## 1 Introduction

This paper deals with measuring a degree of symmetry of  $2D$  subsets of pictures. It helps extracting objects. Symmetry is a prevalent feature in human perception. For instance the human face or body is approximately symmetric that is exploited to assist in face recognition and detection. Psychologists of the Gestalt school have assigned a relevant role to symmetry in attentive mechanism both in visual and auditory systems [1, 15]. From the survey by Zabrodsky [13], we stress upon results corroborating our own findings in the machine domain: - saliency of vertical symmetry provided mental rotation : detections are in the order vertical, horizontal, bent and then rotational symmetry; - parts near the axis contribute more to symmetry than further parts near edges, themselves more critical than regions in between.

The concept of symmetry is important in machine vision too as confirmed by an extensive literature: a recent quite interesting survey is [2]. Models of symmetry suffer three drawbacks: -  $\mathbf{d}_1$  - edges mainly support symmetry detection; -  $\mathbf{d}_2$  - perfect symmetry is targeted; -  $\mathbf{d}_3$  - the center of mass is assumed to be the focus of attention

Similar difficulties have been long solved for edges, regions or motion in actually measuring the phenomenon – edginess , uniformity , set-direction – to decide after the measure rather than using a strict distance. We addressed  $\mathbf{d}_1$  in [14] by defining iterative transforms as the *IOT* that better account for the

inner object. In the present paper we tackle all  $d_i$ -difficulties together to optimally solve the problem. The bibliography of section 2 suggests tools. In section 3, we introduce the notion of a “kernel” that stems logically from *IOT* through a classical gauge in functional analysis. In section 4, preliminary study of the main parameter - the optimal axis to spot - leads to an ultimate detection algorithm based on correlation, and a general enough symmetry measure is derived. A series of experiments in section 5, on both binary and grey scaled pictures, allow to evaluate the technique, to check its sensitivity to the center position and the validity of the degree of symmetry. Short discussion and further comments conclude the paper.

## 2 State of the Art

The use of gray level information was firstly investigated in [6], where the symmetry descriptor is based on a cross correlation of gray levels. In [9], the search for symmetries evaluates the ... of a body around its center of gravity. Applied at a local level this descriptor defines the ... (*DST*). In [7], local reflectional symmetry is computed in convolving with the first and second derivative of Gaussians. Each point gets both a symmetry “measure” and an axis orientation. Shen [17] or DuBuff [18] use complex moments with Fourier or Gabor image approximation. It implies axes to pass through the center of mass, and moments are not invariant to affine transforms.

In [8], authors introduce several descriptors from Marola’s one extended to finite supports and varying scales based on the Radon and Fourier transforms. Scale dependency is claimed to detect global symmetries without any prior segmentation. The ... is then implemented by a probabilistic genetic algorithm for speedup. Likewise, Shen and al. [12] detect symmetry in seeking out the lack of it. The asymmetric term of their measure (energy) is null for any pattern invariant through horizontal reflection, whence minimizing that term over the image. In [11], a multi-scale (see also [10]) vector potential is constructed from the gradient field of filtered images. Edge and symmetry lines are extracted through a vector field (i.e. curl of the vector potential): symmetry axes are where the curl of the vector vanishes and edges are where the divergence of the potential vanishes. Most described methods so far provide symmetry descriptors to compute measures from. Others aim at straight symmetry measures. Comparing for instance Cross’s and Yeshurun’s, Yeshurun and al. [22] build on the Blum-Asada vein [3], but in quantifying a potential for every pixel to be centre of symmetry based on pairs of edge points tentatively symmetric from their respective gradient vectors. A degree of symmetry is assigned to every pair within a given pixel neighborhood and a weighted combination of these makes the pixel potential, whose local maxima provide a measure depending on both intensity and shape. The technique further extends to textures [23]. The review of preceding works points out that: 1) comparing a pattern with its transformed version, for invariance, can prevent from imposing the centroid as the a priori focus of interest; 2) introducing true measures supports more abstract

versions of distances, founding approximate comparison; 3) sets which measures apply on may be “sets of pixels or vectors”(shapes) or “sets of patterns” (in-class transforms): in either case “set operations”, as Minkowski’s ones, are worth considered. They do not limit to contours and bridge logic with geometry.

Three more works fit very well the algorithmic line above and are the closest to ours, making clear the main contributions of this paper. In [19] the authors correlate the image with its transformed version to by-pass the centroid. But they do that on the inner product of (gaussian) gradients, hence on edges. R. Owens [20] searches explicitly for a measure of symmetry to indicate approximate bilateral symmetry of an isolated object. But she defines tentative symmetries from the principal axes of inertia, whence the centroid again, before to compute the sum of absolute differences of grey levels in symmetric pairs over the object, normalized by their maximum. Note that, although it is not mentioned, such a measure amounts to a slightly modified  $L_1$ -difference between the object and a maximal-for-inclusion symmetric version of it in the given direction. Kazhdan et al. [21] target true visualization of symmetry over every point of a pattern. They use explicitly the same idea of a difference ( $L_2$  in their case) between the image and its closest symmetric version (the average of the picture and its transform). But they need a measure that integrates all reflective invariance about a bundle of straight lines (or planes in 3-D). It is robust to noise and suitable for object matching, yet a center is necessary to this representation. The representation plots for every direction the measure of symmetry about the normal plane passing through the center of mass. Note that its local maxima point out potential pattern symmetries.

### 3 The New Symmetry Measure

#### 3.1 Symmetry Indicators (*IOT*)

In [14] we defined *IOT* that is a map product of iterated morphological erosion and symmetry detection.

**Definition 1.** The . . . . . ,  $S_\alpha$ , on a continuous object  $X \subset R^2$  is given by:

$$S_\alpha(X) = \int_X m(x) \times \rho^2(x, r(\alpha)) dx \quad \text{for } \alpha \in [0, \pi[ \quad (1)$$

where,  $r(\alpha)$  is the straight line with slope  $\alpha$  passing through the center of gravity of the object  $X$ ,  $m(x)$  is the mass of the object in  $x \in X$ , and  $\rho$  is a distance function of  $x$  from the straight line.  $\diamond$

**Definition 2.** The . . . . . , *IOT*, is given by:

$$IOT_{\alpha,1}(X) = S_\alpha(X)$$

$$IOT_{\alpha,n}(X) = S_\alpha \left[ (\mathbf{E})^{n-1}(X) \right] \quad \text{for } n > 1 \quad (8)$$

$(E)^n$  stands for the morphological erosion by the unit sphere (or any other suitable structuring element would any suitable a priori information be available), iterated  $n$  times.

The number of iterations depends on the image size and on the gray level distribution. The  $S$  transform is thus computed on progressively shrunk versions of the binary input image or on steadily intensity reduced versions of the gray level input image, until some predefined decrease or a minimum of intensity is reached. The iterated elongation,  $\eta_n(X)$ , is defined as:

$$\eta_n(X) = \frac{\min_{\alpha \in [0, \pi]} \{IOT_{\alpha, n}(X)\}}{\max_{\alpha \in [0, \pi]} \{IOT_{\alpha, n}(X)\}} \quad (8)$$

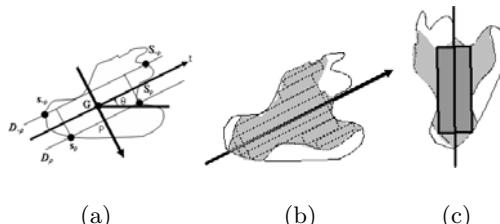
It represents dynamic changes of  $X$  shapes indicators versus  $n$ . Since in most cases,  $\eta$  curves become flat or show some other type of constancy after a certain number of erosions, it was conjectured that any pattern larger than the structuring element would have a symmetric kernel that  $IOT$  reveals (at least one pixel remains after erosion to meet the definition). Let us call  $IOTK$  this pattern. The intuitive idea here is that the closer the kernel to the pattern the more symmetric pattern, although it is easy to design examples (see Figure 1) where the  $IOTK$  is as “far” as wanted from the pattern.

**Remark 1:** when it proves necessary, this included version of the kernel could be balanced by the including version obtained by dilation. Then the question arises to define the kernel more formally.

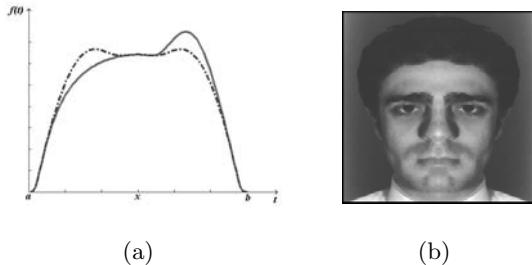
### 3.2 Definition of the Kernel

Following commonly used gauges in functional analysis, a possible first answer with a flavor of optimality would be. . . . . resp.

**Definition 3.** The  $S$ -kernel of the pattern  $P$  -  $SK(P)$  - is the maximal for inclusion symmetric subset of  $P$ . Let us assume we applied the  $IOT$  and found a stable pattern after multiple erosion, like the dark rectangle in the Figure 1c



**Fig. 1.** (a) Sketch of the kernel detection algorithm; (b) the kernel of the pattern in (a); (c) expanding the  $IOTK$  of the pattern in (a) into the kernel



**Fig. 2.** (a) Searching for the symmetry axis of a function  $y = f(t)$  (plain line): its symmetric version around  $x$ ,  $S_f^x(t)$  (dotted line), and (b) the best symmetric version of the face Figure 3-1b

**Remark 2:** the center of mass likely varies from the kernel to the pattern. This introduces an additional question: how to define the likely symmetry axis where to compute the kernel from?

For instance, let be  $\mu = \arg\max_{\mu} \text{Symmetry}(ptrn)$ . How does  $K_\mu(ptrn)$  compare with  $K(ptrn)$ ? How do their respective Symmetry relate? In most cases  $K_\mu(ptrn)$  should be a good enough gauge of  $K(ptrn)$ , or the difference between them will be most indicative. The following section is entirely devoted to answering that. All complementary tests will result into or from experiments described in section 5.

## 4 Formal Setting and an Algorithm

The frame of the present study is not mandatorily that of an isolated pattern any more. Considering operators to be used in the end, correlation, our conjecture is rather of an attention focusing process based on symmetry. Introducing the latter notion of interest implies that all non interesting regions around be faded to zero, and that the pattern scan be started for instance at the very left of it. See below the end of , and results in for an illustration. Hence, the pattern  $f(x)$  gets a bounded support and the origin of the x-axis can be where the interest starts.

#### 4.1 Optimization of the Center Position

Let be (Figure 2):  $S_f^x(t) = \frac{f(x+t) + f(x-t)}{2}$  the symmetric version of  $f$  with respect to  $x$ . For the  $L_2$ -norm, the best axis  $x^*$  corresponds to:

$$S_{x*}(f) = \min_x \int_a^b [f(x+t) - f(x-t)]^2 dt$$

Considering the above general frame,  $a$  can be set to 0 without any loss in generality and support of  $f$  is included in that of  $S_f^x$ .  $f$  is assumed derivable then bounded integrable. It comes:

$$\frac{d}{dx} S_x(f) = \int_0^b \frac{d}{dx} [f(x+t) - f(x-t)]^2 dt$$

As  $f(x) = 0$  for  $x < 0$  and  $x > b$ ,

$$\frac{d}{dx} S_x(f) = 2 \left( \int_0^b f(x+t) \times f'(x-t) dt - \int_0^b f'(x+t) \times f(x-t) dt \right)$$

with  $x+t = u$  (resp.  $x-t = u$ ) then  $x-t = 2x-u$  (resp.  $x+t = 2x-u$ )

$$\int_0^b f'(x+t) \times f(x-t) dt = \int_x^{b+x} f(2x-u) \times f'(u) du$$

(resp.  $\int_0^b f(x+t) \times f'(x-t) dt = \int_{x-b}^x f(2x-u) \times f'(u) du$ )  
 $f(t)$  and  $f'(t)$  being null for  $t < 0$ , it comes in all cases:

$$\int_{x-b}^{x+b} f(2x-u) \times f'(u) du = \int_0^{2x} f(2x-u) \times f'(u) du = f \otimes f'(2x) = \frac{d}{dx} (f \otimes f)$$

with  $\otimes$  the convolution product.

## 4.2 Correlation and Algorithm

Eventually  $S_{x*}(f)$  corresponds to:  $(f \otimes f)$  maximal or equivalently to:

$$\int_{\sup(0,2x-b)}^{\inf(2x,b)} f(2x-u) \times f'(u) du = 0$$

whichever is easier to compute.

One gets yet another algorithm: find the maximum correlation of the picture in a given direction (i.e. over all translations in this direction) with its mirror symmetric in that direction (i.e. scanned right to left). Considering existing efficient algorithms for image geometric transforms (eg. *cordic*), rotations to span directions can then be performed on the image before scans and correlation: approximations need to be checked for the error introduced remain acceptable (comparable to the one from sampled angles and discrete straight lines).

**Remark 2:** Note that if the image is tiled adapted to the predicted size of potential symmetric objects, one can derive an efficient focusing process.

It is now obvious that considering the center of mass  $G$  for all potential symmetry axes to go through amounts to an approximation of  $f$  by the square term of its Taylor expansion, since it is defined by:

$$X_G / \int_0^b (X-u)f(u) du = \frac{1}{2} \int_0^b u^2 f'(X-u) du = 0$$

that is different in general from:

$$x^* / \int_{sup(0,2x-b)}^{inf(2x,b)} f(u) \times f'(2x-u) du = 0$$

As for extending the result in 2 or 3-D, formulas are identical for  $f(\underline{t})$  with  $\underline{t} = (t, s)$  since both derivation and convolution are linear. Actually, one considers  $S_{\underline{x}*}(f) = \min_{\underline{x}} \iint_D [f(\underline{x} + \underline{t}) - f(\underline{x} - \underline{t})]^2 dt.ds$  with  $\underline{x} = (x, 0)$  and  $D$  bounded by  $\phi(t, s) = 0$  determined otherwise or the rectangle frontier of the picture or part of it. It leads to the same formal result since derivation deals with the single variable  $x$  and that makes the case for grey scaled images.

**Remark 3:** The latter formal setting confirms that using the projection (Radon transform) to extract symmetries from provides necessary conditions only since the result of permuting integrals in  $\frac{d}{dx} S_{\underline{x}*}(f)$  is not guaranteed. In the discrete case, it is obvious how to build examples in shuffling column's pixels to get a non symmetric pattern from a symmetric one, still conserving the projection. Even if nature and noise make this type of ambiguities quite rare, in case of multiple symmetries one can be transformed into another by such permutation.

**Remark 4:** In cases where rotational symmetry would be explicitly looked for, polar coordinates can be used. The same formal result holds too:

$$S_\psi(f) = \int_0^{2\pi} \int_0^{\varphi(\theta)} [f(\rho, \theta) - f(\rho, \theta + \psi)]^2 \rho d\rho d\theta$$

since  $\rho$  is positive:

$$f(\rho, \theta) = g(\rho^2, \theta) = g(u, \theta) \implies S_\psi(f) = \int_0^{2\pi} \int_0^{\sqrt{\varphi(\theta)}} [g(u, \theta) - g(u, \theta + \psi)]^2 du d\theta$$

It leads again to the same computation except the pole (center of rotation) is assumed to be known, limiting the interest of the result.

### 4.3 Symmetry Measure

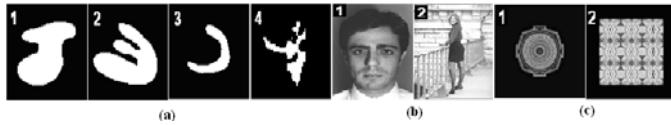
A preliminary simplifying assumption is made here: while the kernel was exhibited, the picture is considered to have been binarized or at least the pattern was cropped if it was not before. So, in order to test the proposed algorithm we compute a measure of symmetry classically defined as:

$$\lambda_1 = 1 - \frac{Area(D)}{Area(A)}$$

with  $A$ , the pattern or a binding sub-picture,  $B$ , its kernel, and  $Area(D) = Area(A - B)$ . Provided suitable binarization or windowing, it remains a robust first approximation where  $\lambda_1 = 1$  if  $Area(B) = Area(A)$ . See results in .

## 5 Experimental Results and Further Comments

In this section we show some results of the application of the S-kernel algorithm (*SKA*) to synthetic and real images. The purpose of experiments can be summarized as follows: 1) validate the ability of the proposed method in measuring the degree of symmetry, by comparing  $|\eta(SK(ptrn)) - \eta(ptrn)|$  with  $\lambda$ , and  $\eta(IOTK)$  with  $\eta(SK(ptrn))$ ; 2) compute the kernel by correlation and compare with *IOTK* (from the algorithm in [14]); 3) compare the axis position obtained by correlation with the best center position obtained after *IOTK*; 4) check ability of the algorithm to support attention focusing from symmetry. All



**Fig. 3.** Sample gallery of images used for experiments: (a) binary; (b) gray level; (c) textured

experiments are parameterized by direction and run on both binary and gray level images (see Figure 3).

### 5.1 Evaluating the Correlation Algorithm

Figures 4 show examples of the similarity measures computed after correlation,  $\rho$  and the similarity  $\lambda$  against the direction  $\alpha$ . It is interesting that *SKA* is able to grasp the circularity of the images 1c and the four axes of symmetry of the image 2c.

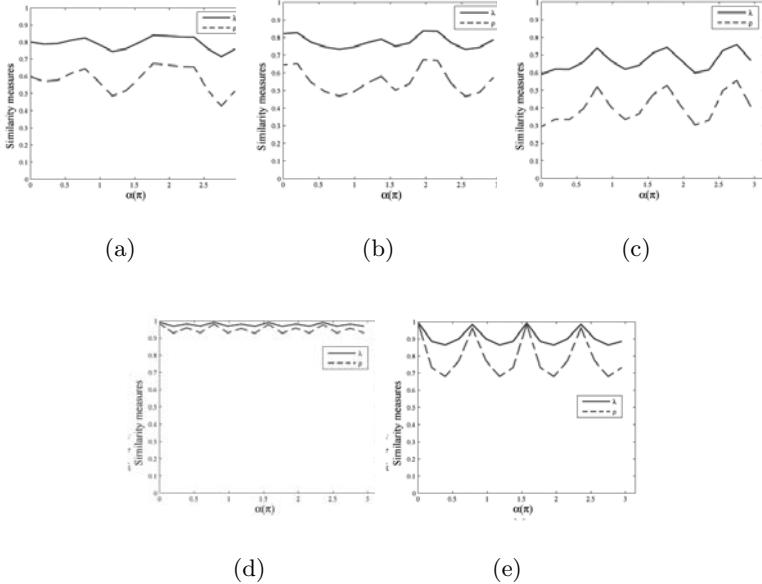
Table 1 reports the results for all images in Figure 3.  $\rho$  ( $\rho_{max}$ ) indicates the object direction  $\alpha$ .

We tested the robustness of *SKA* by rotating the images of  $45^\circ$  and the results of the computation were  $\rho = 0.89$ ,  $\alpha = 45.00^\circ$  for the image 1b, and  $\rho = 0.88$ ,  $\alpha = 135.00^\circ$  for the image 2b.

The direction of images 1a, 2a, and 3a is close to human perceive, the long nose of the sketched profile in image 4a forces the algorithm to an "horizontal" direction. A perfect agreement with *IOTK* exists for images 1b, 2b, 1c, and 2c.

**Table 1.** The correlation algorithm applied to images in Figure 3

Image	1a	2a	3a	4a	1b	2b	1c	2c
$\rho$	0.67	0.67	0.58	0.55	0.80	0.94	0.99	0.98
$\alpha$	$101.25^\circ$	$112.50^\circ$	$112.50^\circ$	$157.50^\circ$	$90.00^\circ$	$90.00^\circ$	$90.00^\circ$	$0.00^\circ$
$OST$	0.86	0.93	0.87	0.80	0.72	0.92	0.90	0.96
$\alpha_{OST}$	$112.50^\circ$	$101.00^\circ$	$56.25^\circ$	$0.00^\circ$	$90.00^\circ$	$45.00^\circ$	$90.00^\circ$	$0.00^\circ$



**Fig. 4.** Examples of correlation and similarity measures for images in Figure 3: (a) image 1a; (b) image 2a; (c) image 4a; (d) image 1c; (e) image 2c

We tested the ability of the correlation operator to exhibit circularity on the image 1c by computing the mean value and the variance of  $\rho$  for all  $\alpha$  and the results was  $(0.96, 0.02)$ .

## 5.2 Application to Attention Focusing

We tested the possibility of using kernel based operators to detect points of interest in complex images. Examples of such images are shown in Figures 5a,b; they represent a famous painting by Tintoretto and a group photo under natural illuminating conditions. In both images the goal was to detect the directions of the most symmetric objects of a given size. For example in the group photo the direction of people faces.

The procedure consists in raster scanning the input image with a window, size of which is set on the basis of the human face dimensions scaled to the input frame. Inside each window kernel-based operators are computed. The algorithm returns all windows for which the value of  $\lambda$  ( $\rho$ ) is greater than a given threshold  $\phi \in [0, 1]$ . Here, the threshold was set to the mean value of  $\lambda$  ( $\rho$ ) in all experiments. A great value of  $\lambda$  ( $\rho$ ) in a given direction indicates a bilateral symmetry typical of face like objects. The Figure 5 shows the results from SKA. Not all objects with high bilateral symmetry are faces. Nevertheless the method was able to extract all face positions, introducing an error of 17% in the evaluation of the face direction; over all experiments the percentage of not faces was 21%.



**Fig. 5.** Attention focusing by symmetry from (*SKA*): (a) Group of women (Tintoretto 1545-1588); (b) group photo

## 6 Concluding Remarks

This paper describes a new measure of axial symmetry derived from a new object feature named the “symmetry-kernel”. The symmetry kernel of an object is its maximal subpart symmetric respective to a given direction. A new algorithm is derived from, based on the computation of the cross-correlation of an object with its flipped version. It is fast and not sensitive to numerical factors because computations are inner products. The algorithm was tested on both synthetic and real data. Experiments show the ability of the symmetry-kernel to detect the main directionality of an object. It has been also implemented as a local operator to detect the presence of objects in a scene and their direction. The evaluation of the distance between an object and its kernel is a crucial point and needs further investigation.

## References

1. W.Khöler and H.Wallach, “Figural after-effects: an investigation of visual processes”, *Proc. Amer. phil. Soc.*, Vol.88, 269-357, 1944.
2. David O’Mara, “Automated facial metrology, chapter 4: Symmetry detection and measurement” PhD thesis, Feb. 2002
3. H.Blum and R.N.Nagel, “Shape description using weighted symmetric axis features”, *Pattern recognition*, Vol.10, pp.167-180, 1978.
4. M.Brady, H.Asada, “Smoothed Local Symmetries and their implementation”, *The International Journal of Robotics Research*, Vol.3, No.3, pp.36-61, 1984.
5. A. Sewisy, F. Lebert, “Detection of ellipses by finding lines of symmetry in the images via an Hough transform applied to staright lines”, *Image and Vision Computing*, Vol. 19 - 12, Oct. 2001, pp. 857-866
6. G.Marola, “On the detection of the axes of symmetry of symmetric and almost symmetric planar images”, *IEEE Trans.of PAMI*, Vol.11, pp.104-108, 1989.
7. R. Mammatha, H. Sawhney, “Finding symmetry in Intensity Images”, *Technical Report*, 1997

8. N.Kiryati, Y.Gofman, "Detecting symmetry in grey level images (the global optimization approach)", *preprint*, 1997.
9. V.Di Gesù, C.Valenti, "Symmetry operators in computer vision", in *Vistas in Astronomy*, Pergamon, Vol.40, No.4, pp.461-468,1996.
10. V.Di Gesù, C.Valenti, "Detection of regions of interest via the Pyramid Discrete Symmetry Transform", in *Advances in Computer Vision* (Solina, Kropatsch, Klette and Bajcsy editors), Springer-Verlag, 1997.
11. A. D. J. Cross and E. R. Hancock, "Scale space vector fields for symmetry detection", *Image and Vision Computing*, Volume 17, 5-6, pp. 337-345, 1999.
12. D. Shen, H. Ip, E.K. Teoh, "An energy of assymmetry for accurate detection of global reflexion axes, *Image Vision and Computing* 19 (2001), pp. 283-297.
13. H. Zabrodsky, "Symmetry - A review", *Technical Report 90-16*, CS Dep. The Hebrew University of Jerusalem
14. V. DiGesu and B. Zavidovique, "A note on the Iterative Object Symmetry Transform", *Pattern Recognition Letters, Pattern Recognition Letters*, Vol. 25, pp. 1533-1545, 2004.
15. R.M. Boyton, C.L. Elworth, J. Onley, C.L. Klingberg, "Form discrimination as predicted by overlap and area", *RADC-TR-60-158*, 1960
16. S. Fukushima, "Division-based analysis of symmetry and its applications", *IEEE PAMI*, Vol. 19-2, 1997
17. D. Shen, H. Ip, K.T. Cheung, E.K. Teoh, "Symmetry detection by Generalized complex moments: a close-form solution", *IEEE PAMI*, Vol. 21-5, 1999
18. J. Bigun, J.M.H. DuBuf, "N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation" *IEEE PAMI*, Vol. 16-1, 1994.
19. T. Masuda, K. Yamamoto, H. Yamada, "Detection of partial symmetyr using correlation with rotated-reflected images", *Pattern Recognition*, Vol. 26-8, 1993
20. D. O'Maraa, R. Owens, "Measuring bilateral symmetry in digital images". *IEEE TENCON*, Digital signal processing aplications, 1996
21. M. Kazhdan, B. Chazelle, D. Dobkin , A. Finkelstein, T. Funkhouser, "A reflective symmetry descriptor", *7<sup>th</sup> ECCV*, pp. 642-656, 2002
22. D.Reisfeld, H.Wolfson, Y.Yeshurun, "Detection of interest points using symmetry", *3rd IEEE ICCV* Osaka, Dec. 1990.
23. Y. Bonneh, D.Reisfeld, Y.Yeshurun, "Texture discrimination by local generalized symmetry", *4th IEEE ICCV* Berlin, May 1993
24. A. Merigot, B. Zavidovique, "Image analysis on massively parallel computers : An architectural point of view".*J. of pattern recognition and artificial intelligence*, Vol. 6 n°2-3, pp. 387-393, World Scientific, 1992

# Training Cellular Automata for Image Processing

Paul L. Rosin

Cardiff School of Computer Science,  
Cardiff University, Cardiff, CF24 3AA, UK  
[Paul.Rosin@cs.cf.ac.uk](mailto:Paul.Rosin@cs.cf.ac.uk),  
<http://users.cs.cf.ac.uk/Paul.Rosin>

**Abstract.** Experiments were carried out to investigate the possibility of training cellular automata to perform processing. Currently, only binary images are considered, but the space of rule sets is still very large. Various objective functions were considered, and sequential floating forward search used to select good rule sets for a range of tasks, namely: noise filtering, thinning, and convex hulls. Several modifications to the standard CA formulation were made (the B-rule and 2-cycle CAs) which were found to improve performance.

## 1 Introduction

Cellular automata (CA) consist of a regular grid of cells, each of which can be in only one of a finite number of possible states. The state of a cell is determined by the previous states of a surrounding neighbourhood of cells and is updated synchronously in discrete time steps. The identical rule contained in each cell is essentially a finite state machine, usually specified in the form of a rule table with an entry for every possible neighbourhood configuration of states.

Cellular automata are discrete dynamical systems, and they have been found useful for simulating and studying phenomena such as ordering, turbulence, chaos, symmetry-breaking, etc, and have had wide application in modelling systems in areas such as physics, biology, and sociology.

Over the last fifty years a variety of researchers (including well known names such as John von Neumann [15], Stephen Wolfram [16], and John Conway [7] have investigated the properties of cellular automata. Particularly in the 1960's and 1970's considerable effort was expended in developing special purpose hardware (e.g. CLIP) alongside developing rules for the application of the CAs to image analysis tasks [11]. More recently there has been a resurgence in interest in the properties of CAs without focusing on massively parallel hardware implementations. By the 1990's CAs could be applied to perform a range of computer vision tasks, such as: calculating properties of binary regions such as area, perimeter, convexity [5], gap filling and template matching [4], and noise filtering and sharpening [8],

One of the advantages of CAs is that, although each cell generally only contains a few simple rules, the combination of a matrix of cells with their local

interaction leads to more sophisticated emergent global behaviour. That is, although each cell has an extremely limited view of the system (just its immediate neighbours), localised information is propagated at each time step, enabling more global characteristics of the overall CA system.

A disadvantage with the CA systems described above is that the rules had to be carefully and laboriously generated by hand [14]. Not only is this tedious, but it does not scale well to larger problems. More recently there has been a start to automating rule generation using evolutionary algorithms. For instance, Sipper [13] shows results of evolving rules to perform thinning, and gap filling in isothetic rectangles. Although the tasks were fairly simple, and the results were only mediocre, his work demonstrates that the approach is feasible.

## 2 Design and Training of the CA

In the current experiments all input images are binary, and cells have two states (i.e. they represent white or black). Each cell's eight-way connected immediate neighbours are considered (i.e. the Moore neighbourhood). Fixed value boundary conditions are applied in which transition rules are only applied to non-boundary cells. The input image is provided as the initial cell values.

### 2.1 The Rule Set

Working with binary images means that all combinations of neighbour values gives  $2^8$  possible patterns or rules. Taking into account  $90^\circ$  rotational symmetry and bilateral reflection provides about a five-fold decrease in the number of rules, yielding 51 in total.

The 51 neighbourhood patterns are defined for a central black pixel, and the same patterns are inverted (i.e. black and white colours are swapped) for the equivalent rule corresponding to a central white pixel. According to the application there are several possibilities:

- both of these two central black and white rule sets can be learnt and applied separately,
- the two rule sets are considered equivalent, and each corresponding rule pair is either retained or rejected for use together, leading to a smaller search space of possible rule sets,
- just one of the black and white rule sets is appropriate, the other is ignored in training and application.

Examples of the latter two approaches will be shown in the following sections.

### 2.2 Training Strategy

Most of the literature on cellular automata studies the effect of applying manually specified transition rules. The inverse problem of determining appropriate rules to produce a desired effect is hard [6]. In our case there are  $2^{102}$  or  $2^{51}$

combinations of rules to be considered! Evolutionary methods appear to be preferred; for instance to solve the density classification task researchers have used genetic algorithms [10] and genetic programming [1]. Instead, we currently use a deterministic approach, which is essentially modified hill-climbing: the sequential floating forward search (SFFS) [12].

### 2.3 Objective Functions

An objective function is required to direct the SFFS, and various error measures have been considered in this paper. The first is root mean square (RMS) error between the input and target image.

In some applications there will be many more black pixels than white (or vice versa) and it may be preferable to quantify the errors of the black pixels separately from the white. This is done by computing the proportion  $B$  of black target pixels incorrectly coloured in the output image, and likewise  $W$  is computed for white target pixels. The combined error is taken as  $B + W$ .

The above measures do not consider the positions of pixels. In an attempt to incorporate spatial information, the distance at each incorrectly coloured pixel in the output image to the closest correctly coloured pixel in the target image is calculated. The final error is the summed distances. The distances can be determined efficiently using the distance transform of the target image.

A modification of the above is the Hausdorff distance. Rather than summing the distances only the maximum distance (error) is returned.

### 2.4 Extensions

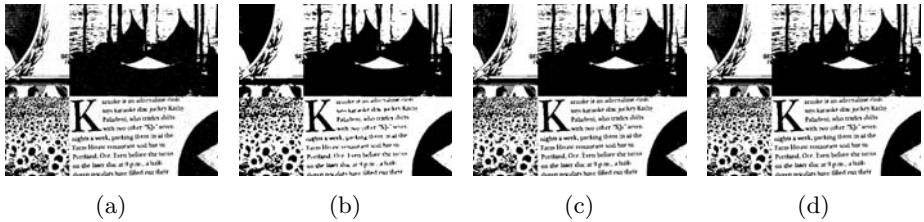
There are many possible extensions to the basic CA mechanism described above. In this paper two modifications were implemented and tested. The first is based on Yu . . 's [17] B-rule class of one dimensional CA. Each rule tests the value of the central pixel of the . . . . iteration in addition to the usual pixel and its neighbour's values at the . . . . iteration. The second variation is to split up the application of the rules into two interleaved cycles. In the even numbered iterations one rule set is applied, and in the odd numbered iterations the other rule set is applied. The two rule sets are learnt using SFFS as before, and are not restricted to be disjoint.

## 3 Noise Filtering

The first experiment is on filtering to overcome salt and pepper noise. Figure 1 shows a portion of the large training and test images. Varying amounts of noise were added, and for each level the CA rules were learnt using the various strategies and evaluation criteria described above. In all instances the rules were run for 100 iterations. It was found that using the SFFS method with the RMS error criterion provided the best results, and unless otherwise stated all the results shown used this setup.

**Table 1.** RMS errors of filtered versions of single pixel salt and pepper noise

S & P prob.	orig.	median			CA	CA B-rule	CA 2 cycle
		1 iteration	100 iterations	optimal iterations			
0.01	320	1807	2925	1807 (1)	199	147	199
0.1	3247	2027	3065	2027 (1)	1048	1009	1044
0.3	9795	3113	3521	2836 (2)	2268	2272	2263

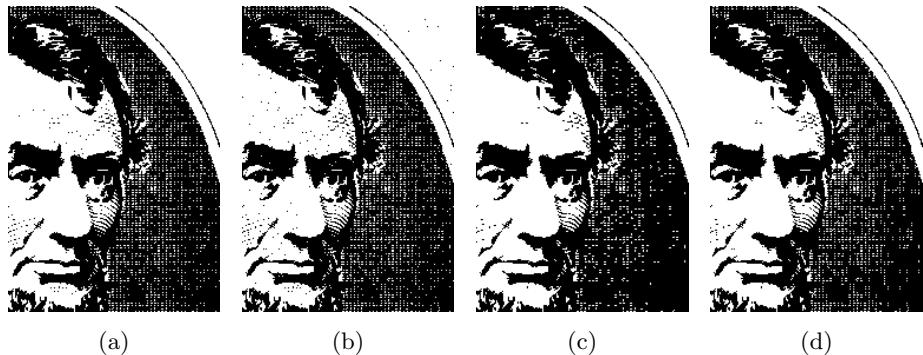
**Fig. 1.**  $585 \times 475$  sections of the  $1536 \times 1024$  original images before noise was added; (a), (b) training/test images**Fig. 2.** Salt and pepper noise affecting single pixels occurring with a probability of 0.01; (a) unfiltered, (b) 1 iteration of median, (c) CA, (d) B-rule CA

For comparison, results of filtering with a  $3 \times 3$  median filter are provided. While there are more sophisticated filters in the literature [3] this still provides a useful benchmark. Moreover, the optimal number of iterations of the median was determined for the . . . image, giving a favourable bias to the median results.

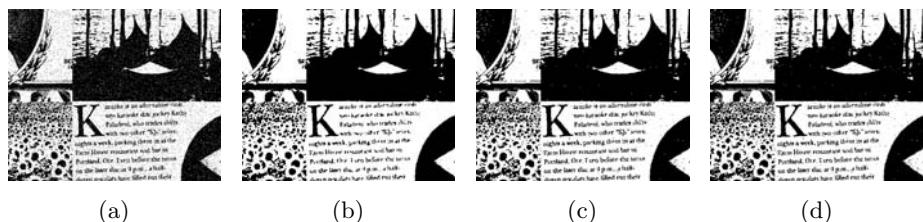
At low noise levels ( $p = 0.01$ ) the CA learns to use a single rule<sup>1</sup> (☒) to remove isolated pixels. As the RMS values show (table 1) this is considerably better than median filtering which in these conditions has its noise reduction overshadowed by the loss of detail, see figure 2. The B-rule CA produces even better results than the basic CA. 50 rules were learnt, although this is probably

<sup>1</sup> The rules are shown with a black central pixel – which is flipped after application of the rule. The neighbourhood pattern of eight white and/or black (displayed as gray) pixels which must be matched is shown. The rule set is shown (left to right) in the order that the rules were added to the set by the SFFS process.

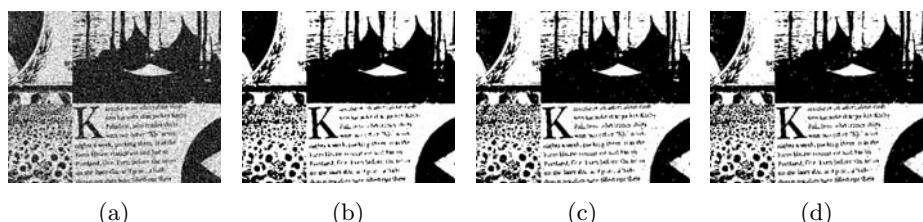
far from a minimal set since most of them have little effect on the evaluation function during training. As before, the first rule is  $\square\square$ , applied when the central pixel is a different colour in the previous iteration. In contrast, most of the remaining rules are applied when the central pixel is the same colour in the previous iteration. The difference in the outputs of the basic and B-rule CAs is most apparent on the finely patterned background to Lincoln (figure 3), which



**Fig. 3.** An enlarged portion from the original and processed images with 0.01 probability salt and pepper noise. (a) original, (b) original with added noise, (c) filtered with CA, (d) filtered with CA B-rule



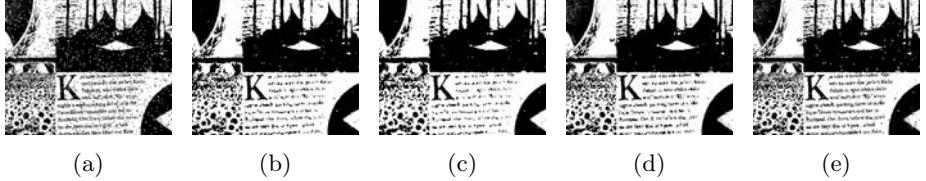
**Fig. 4.** Salt and pepper noise affecting single pixels occurring with a probability of 0.1; (a) unfiltered, (b) 1 iteration of median, (c) CA, (d) B-rule CA



**Fig. 5.** Salt and pepper noise affecting single pixels occurring with a probability of 0.3; (a) unfiltered, (b) 2 iterations of median, (c) CA, (d) B-rule CA

**Table 2.** RMS errors of filtered versions of  $3 \times 3$  pixel salt and pepper noise

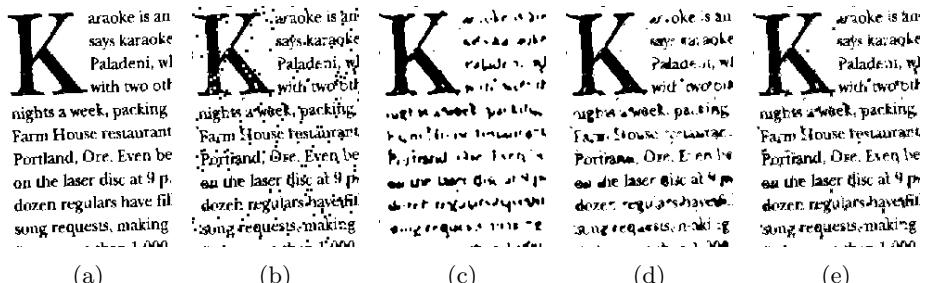
S & P prob.	orig.	3 $\times$ 3 median			5 $\times$ 5 median			CA	CA B-rule	CA 2 cycle
		1 iter.	100 iter.	opt. iter.	1 iter.	100 iter.	opt. iter.			
0.01	2819	3704	3465	3145 (3)	3923	6287	3923 (1)	2011	1425	1988
0.1	19886	18250	13583	13583 (39)	15621	9041	8930 (25)	8530	8090	8622

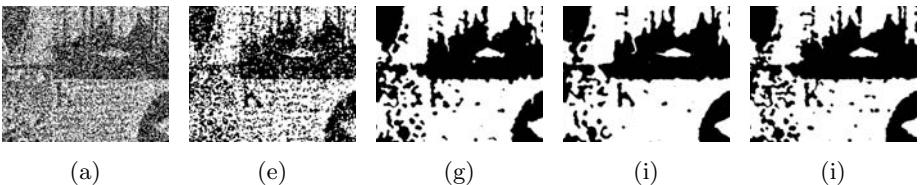
**Fig. 6.** Salt and pepper noise affecting  $3 \times 3$  blocks occurring with a probability of 0.01; (a) unfiltered, (b) 3 iterations of median, (c) 1 iteration of  $5 \times 5$  median, (d) CA, (e) B-rule CA

has been preserved while the noise on the face has still been removed. The 2-cycle CA produces identical results to the basic CA.

At greater noise levels the CA continues to perform consistently better than the median filter (figures 4 & 5). At  $p = 0.1$  the learnt CA rule set is  $\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$  and required 31 iterations for convergence. At  $p = 0.3$  the learnt CA rule set is  $\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$  and required 21 iterations for convergence. Again the 2-cycle CA produced little improvement over the basic CA, while the B-rule CA does at  $p = 0.1$  but not  $p = 0.3$ . The B-rule rule sets are reasonably compact, and the one for  $p = 0.1$  is shown: the rule set applied when the central pixel is a different colour in the previous iteration is  $\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$  while for the same coloured central pixel at the previous iteration the rule set is  $\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$ .

Increasing levels of noise obviously requires more filtering to restore the image. It is interesting to note that not only have more rules been selected as the

**Fig. 7.** An enlarged portion from the original and processed images with 0.01 probability salt and pepper noise. (a) original, (b) original with added noise, (c) filtered with 3 iterations of median filter, (d) filtered with CA, (e) filtered with CA B-rule



**Fig. 8.** Salt and pepper noise affecting  $3 \times 3$  blocks occurring with a probability of 0.1; (a) unfiltered, (b) 39 iterations of median, (c) 25 iterations of  $5 \times 5$  median, (d) CA, (e) B-rule CA

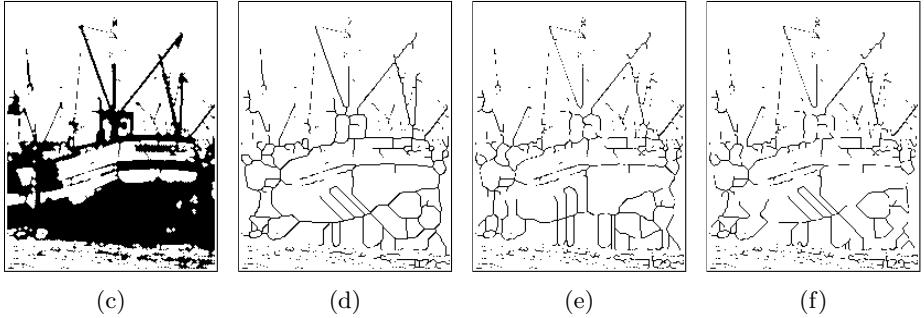
noise level increases, but also that, for the basic CA, they are strictly supersets of each other.

The second experiment makes the noise filtering more challenging by setting  $3 \times 3$  blocks, rather than individual pixels, to black or white. However, the CA still operates on a  $3 \times 3$  neighbourhood. Given the larger structure of the noise larger ( $5 \times 5$ ) median filters were used. However, at low noise levels ( $p = 0.01$ ) the  $3 \times 3$  median gave a lower RMS than the  $5 \times 5$  although the later was better at high noise levels ( $p = 0.1$ ). Nevertheless, the basic CA outperformed both (table 2). At  $p = 0.01$  (figure 6) the learnt rule set was  $\begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array}$  and required 42 iterations for convergence. The B-rule CA further improved the result, and this can most clearly be seen in the fragment of text shown in figure 7. At  $p = 0.1$  (figure 8) the learnt rule set was  $\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$  and even after 100 iterations the CA had not converged. The 2-cycle CA did not show any consistent improvement over the basic CA.

## 4 Thinning

Training data was generated in two ways. First, some one pixel wide curves were taken as the target output, and were dilated by varying amounts to provide the pre-thinned input. In addition, some binary images were thinned by the thinning algorithm by Zhang and Suen [18]. Both sets of data were combined to form a composite training input and output image pair. Contrary to the image processing tasks in the previous sections the RMS criterion did not produce the best results, and instead the summed proportions of black pixel errors and white pixel errors was used. Surprisingly the summed distance and Hausdorff distance error measures gave very poor results. It had seemed likely that they would be more appropriate for this task given the sparse nature of skeletons which would lead to high error estimates for even small mislocations if spatial information were not incorporated. However, it was noted that they did not lead the SFPS procedure to a good solution. Both of them produced rule sets with higher errors than the rule set learnt using RMS, even according to their own measures.

The test image and target obtained by Zhang and Suen's thinning algorithm are shown in figures 9a&b. The basic CA does a reasonable job (figure 9c), and



**Fig. 9.** Image thinning; (a) test input, (b) test image thinned using Zhang and Suen’s algorithm, (c) test image thinned with CA, (d) test image thinned with 2 cycle CA

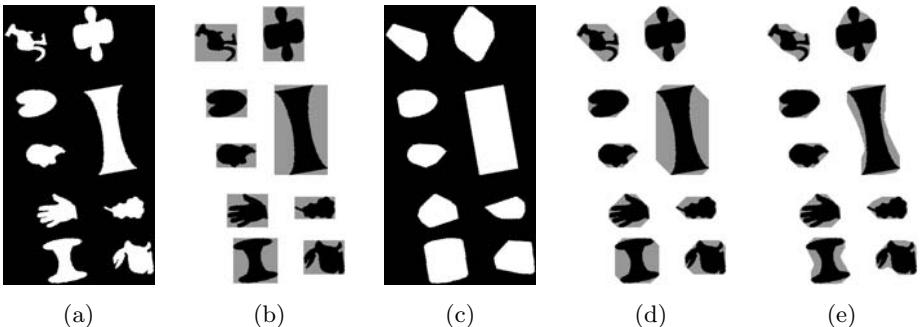
the rule set is  $\boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}}$ . The last rule has little effect, only changing three pixels in the image. Some differences with respect to Zhang and Suen’s output can be seen. In the wide black regions horizontal rather than diagonal skeletons are extracted, although it is not obvious which is more correct. Also, a more significant problem is that some lines were fragmented. This is not surprising since there are limitations when using parallel algorithms for thinning, as summarised by Lam . . [9]. They state that to ensure connectedness either the neighbourhood needs to be larger than  $3 \times 3$ . Alternatively,  $3 \times 3$  neighbourhoods can be used, but each iteration of application of the rules is divided into a series of subcycles in which different rules are applied.

This suggests that the two cycle CA should perform better. The rule set learnt for the first cycle is  $\boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}}$  and the second cycle rule set is a subset of the first:  $\boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}}$ . Again the last and least important rule from the first cycle has little effect (only changing 6 pixels) and so the basic CA and the first cycle of the B-rule have effectively the same rule set. As figure 9d shows, the output is a closer match to Zhang and Suen’s, as the previously vertical skeleton segments are now diagonal, but connectivity is not improved.

## 5 Convex Hulls

The next experiment tackles finding the convex hulls of all regions in the image. If the regions are white then rules need only be applied at black pixels since white pixels should not be inverted. As for the thinning task, the summed proportions of black pixel errors and white pixel errors was used. After training the learnt rule set was applied to a separate test image (figure 10a). Starting with a simple approximation as the output target, a four-sided hull, i.e. the axis aligned minimum bounding rectangle (MBR), the CA is able to produce the correct result as shown in figure 10b. The rule set learnt is  $\boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}} \quad \boxed{\text{■}}$ .

Setting as target the digitised true convex hull (see figure 10c) the CA learns to generate an eight-sided approximation to the convex hull (figure 10d) using



**Fig. 10.** Results of learning rules for the convex hull. (a) test input; (b) CA result with MBR as target overlaid on input; (c) target convex hull output; (d) CA result with (c) as target overlaid on input; (e) 2 cycle CA result with (c) as target overlaid on input

the rule set  $\boxed{\text{■}} \quad \boxed{\text{■■}} \quad \boxed{\text{■■■}} \quad \boxed{\text{■■■■}} \quad \boxed{\text{■■■■■}}$ . Interestingly, in comparison to the eight-sided output the only difference to the rules for the four-sided output is the removal of the single rule  $\boxed{\text{■■}}$ . The limitations of the output convex hull are to be expected given the limitations of the current CA. Borgefors and Sanniti di Baja [2] describe parallel algorithms for approximating the convex hull of a pattern. Their  $3 \times 3$  neighbourhood algorithm produces similar results to figure 10d. To produce better results they had to use larger neighbourhoods, and more complicated rules.

Therefore, extending the basic CAs capability by applying the 2-cycle version should enable the quality of the convex hull to be improved. As figure 10e shows the result is no longer convex although it is a closer match to the target in terms of its RMS error. This highlights the importance of the evaluation function. In this instance simply counting pixels is not sufficient, and a penalty function that avoids non-convex solutions would be preferable, although computationally more demanding.

## 6 Conclusions

The initial experiments with CAs are encouraging. It was shown that it is possible to learn good rule sets to perform common image processing tasks. Moreover, the modifications to the standard CA formulation (the B-rule and 2-cycle CAs) were found to improve performance. In particular, for filtering salt and pepper noise, the CA performed better than standard median filtering.

To further improve performance there are several areas to investigate. The first is alternative neighbourhood definitions (e.g. larger neighbourhoods, circular and other shaped neighbourhoods, different connectivity). Second, although several objective functions were evaluated, there may be better ones available – particularly if they are tuned to the specific image processing task. Third, can additional constraints be included to prune the search space, improving effi-

ciency and possibly effectiveness? Fourth, alternative training strategies to SFFS should be considered, such as evolutionary programming.

Most CAs use identical rules for each cell. An extension would be to use non-uniform CA, in which different rules could be applied at different locations, and possibly also at different time steps depending on local conditions.

## References

1. D. Andre, F.H. Bennett III, and J.R. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In *Proc. Genetic Prog.*, pages 3–11. MIT Press, 1996.
2. G. Borgefors and G. Sanniti di Baja. Analysing nonconvex 2D and 3D patterns. *Computer Vision and Image Understanding*, 63(1):145–157, 1996.
3. T. Chen and H.R. Wu. Application of partition-based median type filters for suppressing noise in images. *IEEE Trans. Image Proc.*, 10(6):829–836, 2001.
4. T. de Saint Pierre and M. Milgram. New and efficient cellular algorithms for image processing. *CVGIP: Image Understanding*, 55(3):261–274, 1992.
5. C.R. Dyer and A. Rosenfeld. Parallel image processing by memory-augmented cellular automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):29–41, 1981.
6. N. Ganguly, B.K. Sikdar, A. Deutsch, G. Canright, and P.P. Chaudhuri. A survey on cellular automata. Technical Report 9, Centre for High Performance Computing, Dresden University of Technology, 2003.
7. M. Gardner. The fantastic combinations of john conway’s new solitaire game “life”. *Scientific American*, pages 120–123, 1970.
8. G. Hernandez and H.J. Herrmann. Cellular automata for elementary image enhancement. *Graphical Models and Image Processing*, 58(1):82–89, 1996.
9. L. Lam and C.Y. Suen. An evaluation of parallel thinning algorithms for character-recognition. *IEEE Trans. PAMI*, 17(9):914–919, 1995.
10. F. Jiménez Morales, J.P. Crutchfield, and M. Mitchell. Evolving 2-d cellular automata to perform density classification. *Parallel Comp.*, 27:571–585, 2001.
11. K. Preston and M.J.B. Duff. *Modern Cellular Automata-Theory and Applications*. Plenum Press, 1984.
12. P. Pudil, J. Novovicova, and J.V. Kittler. Floating search methods in feature-selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.
13. M. Sipper. The evolution of parallel cellular machines toward evolware. *BioSystems*, 42:29–43, 1997.
14. B. Viher, A. Dobnikar, and D. Zazula. Cellular automata and follicle recognition problem and possibilities of using cellular automata for image recognition purposes. *Int. J. of Medical Informatics*, 49(2):231–241, 1998.
15. J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
16. S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley, 1994.
17. D. Yu, C. Ho, X. Yu, and S. Mori. On the application of cellular automata to image thinning with cellular neural network. In *Cellular Neural Networks and their Applications*, pages 210–215. 1992.
18. T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns. *Comm. ACM*, 27(3):236–240, 1984.

# Functional 2D Procrustes Shape Analysis

Rasmus Larsen

Informatics and Mathematical Modelling, Technical University of Denmark  
Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Denmark  
[rl@imm.dtu.dk](mailto:rl@imm.dtu.dk), <http://www.imm.dtu.dk/image>

**Abstract.** Using a landmark based approach to Procrustes alignment neglects the functional nature of outlines and surfaces. In order to re-introduce this functional nature into the analysis we will consider alignment of shapes with functional representations. First functional Procrustes analysis of curve shapes is treated. Following this we will address the analysis of surface shapes.

## 1 Introduction

In this paper we consider the representation and alignment of two dimensional points sets, curves and surfaces. The curves and surfaces may arise as outlines and areas delineated by outlines of two dimensional objects or cross sections or projections of three dimensional objects. Intuitively and formalized in the definition by [1] an object's shape is invariant under a Euclidean similarity transformation. Often a set of curves delineates an area of interest. In some of these cases it may then be appropriate to consider alignment with respect to the interior of these objects instead of their outline or landmarks on their outline. Prior to any modelling of shape we need to filter out these nuisance parameters of a Euclidean similarity transformation from each object in our data set. We define functional Procrustes analysis based on spline based representations of outlines as well as spline based representations of regions. Thus generalizing the method of generalized Procrustes analysis (GPA) based on sets of labelled landmarks [2, 3, 4].

## 2 Functional Generalized Procrustes Analysis

In the following two representations of shapes are considered, namely functional curve and surface representations. A **functional 2D curve shape** consisting of an open or closed continuous curve is represented by a continuous complex function  $y(s) : [0, l] \rightarrow \mathbb{C}$ . For closed curves  $y(0) = y(l)$ . In our applications we consider all such curves with the exception of curves for which  $y(s) = \text{const.}$  for all  $s \in [0, l]$ . The **centroid** of a functional curve shape is given by  $\bar{y} = \frac{1}{l} \int_0^l y(s) ds$ . A curve with centroid 0 is said to be centered. The centroid size is  $S(y) =$

$\left\{ \int_0^l |y(s) - \bar{y}|^2 ds \right\}^{1/2}$  We may choose to use the natural parameterization of the curves. However, generally the correspondences between outlines of biological and other objects are not given by the normalized distance traversed along the outline. More often we assume that the correspondence between such outlines are given at sets of manually or (semi)-automatically identified anatomical and/or geometrical landmarks along the outlines, and natural or other parameterizations are used to define correspondences between landmarks. In other situations the correspondences have to be estimated directly from data.

A **functional 2D surface shape** consisting of a surface patch is represented by a continuous complex function  $y(s) : \Omega \rightarrow \mathbb{C}$ . In our application we will consider such surface patches that are constrained to have non-zero area. The centroid of a functional surface shape is given by  $\bar{y} = \frac{1}{|\Omega|} \int_{\Omega} y(s) ds$ , where  $|\Omega|$  is the size(area) of the parameter space. A surface with centroid 0 is said to be centered. The centroid size is  $S(y) = \left\{ \int_{\Omega} |y(s) - \bar{y}|^2 ds \right\}^{1/2}$

## 2.1 Functional Curve Shapes

Let us consider two curves  $y(s), w(s) : [0, l] \rightarrow \mathbb{C}$ . Without loss of generality we assume that the curves have been centered, i.e.  $\int_0^l y(s) ds = \int_0^l w(s) ds = 0$ .

**Definition 1.**  $w^P(s) = \hat{a} + \hat{b}w(s)$   
 $(\hat{a}, \hat{b})$

$$D^2(y, w) = \int_0^l |y(s) - bw(s) - a|^2 ds.$$

$a \in \mathbb{C}$ ,  $\beta = \text{mod}(b) \in \mathbb{R}_+$ ,  
 $0 \leq \theta = \arg(b) < 2\pi$

**Result 1.**

$$\hat{a} = 0 \quad \hat{b} = \frac{\int_0^l w(s)^* y(s) ds}{\int_0^l w(s)^* w(s) ds}$$

Omitted, follows from differentiation of the objective function.

To obtain a symmetric measure of shape distance we standardize the curve shapes to unit size. The objective function then becomes

**Definition 2.** functional curve shape Procrustes distance

$$d_F^C(y, w) = \left\{ 1 - \frac{\int_0^l y(s)^* w(s) ds \int_0^l w(s)^* y(s) ds}{\int_0^l y(s)^* y(s) ds \int_0^l w(s)^* w(s) ds} \right\}^{1/2}.$$

## 2.2 Functional Procrustes Mean Curve Shape

Let a sample of  $n$  two dimensional curves given by  $w_i(s) : [0, l] \rightarrow (C)$  be available from the perturbation model

$$w_i(s) = a_i + b_i(\mu(s) + \epsilon_i(s)), \quad i = 1, \dots, n,$$

where  $a_i \in \mathbb{C}$  are translation vectors,  $\beta_i = \text{mod}(b_i) \in \mathbb{R}_+$  are scale parameters,  $0 \leq \theta_i = \arg(b_i) < 2\pi$  are rotations,  $\epsilon_i(s) : [0, l] \in \mathbb{C}$  are independent zero mean complex random error functions, and  $\mu$  is the population mean curve. Under this model it is possible to estimate the shape of  $\mu$ ,  $[\mu]$ .

**Definition 3.**

$$[\hat{\mu}] = \arg \inf_{\mu} \sum_{i=1}^n (d_F^C)^2(w_i, \mu)$$

$$[\hat{\mu}] = \arg \inf_{\mu} \sum_{i=1}^n (d_F^C)^2(w_i, \mu) \quad (1)$$

## 2.3 Functional Curve Shape Representation

A convenient representation of curves is based on linear basis expansions in  $s$

$$y(s) = \sum_{m=1}^M c_m h_m(s), \quad s \in [0, l], \quad (2)$$

where  $c_m \in \mathbb{C}$ ,  $h(s) : [0, l] \rightarrow \mathbb{R}$ . For closed curves  $y(0) = y(l)$ . The parameter  $s$  provides the correspondence between curves. A centered curve shape is obtained by translating the linear basis function coefficients by  $\mathbf{w}^T \mathbf{c} / \mathbf{w}^T \mathbf{1}_M$ , where  $\mathbf{w} = \int_0^l \mathbf{h}(s) ds$ . Let two centered curves be given by linear combinations of the same set of basis functions, i.e.

$$y(s) = \sum_{m=1}^M c_m h_m(s) = \mathbf{h}(s)^T \mathbf{c}, \quad w(s) = \sum_{m=1}^M d_m h_m(s) = \mathbf{h}(s)^T \mathbf{d},$$

where we have introduced a vector notation for the coefficients and basis functions:  $\mathbf{c} = (c_1, \dots, c_M)^T$ ,  $\mathbf{d} = (d_1, \dots, d_M)^T$ , and  $\mathbf{h}(s) = (h_1(s), \dots, h_M(s))^T$ .

**Result 2.**

$$\hat{a} = 0 \quad \hat{b} = \frac{\mathbf{d}^* \mathbf{A} \mathbf{c}}{\mathbf{d}^* \mathbf{A} \mathbf{d}} = \frac{(\mathbf{L} \mathbf{d})^* (\mathbf{L} \mathbf{c})}{(\mathbf{L} \mathbf{d})^* (\mathbf{L} \mathbf{d})}$$

$$\mathbf{A} = \mathbf{L} \mathbf{L}^T = \int_0^l \mathbf{h}(s) \mathbf{h}(s)^T ds$$

$$\mathbf{L} \mathbf{A} = \mathbf{A} \mathbf{L}$$

Follows directly from Result 1.  $\square$

Let a sample of  $n$  centered curves be given by

$$w_i(s) = \sum_{m=1}^M d_{im} h_m(s) = \mathbf{h}(s)^T \mathbf{d}_i, \quad i = 1, \dots, n, \quad \mathbf{d}_i = (d_{i1}, \dots, d_{iM})^T$$

**Result 3.**

$$w_i = \mathbf{L}^{-1} \mathbf{v} \quad \mathbf{v} \in \mathbb{C}^M$$

$$\mathbf{C} = \sum_{i=1}^n \frac{(\mathbf{L}\mathbf{d}_i)(\mathbf{L}\mathbf{d}_i)^*}{(\mathbf{L}\mathbf{d}_i)^*(\mathbf{L}\mathbf{d}_i)}.$$

Obviously, the minimizing  $\mu(s)$  must belong to the same linear subspace as the sample of curves. Subject to  $S(\mu) = 1$  we seek the minimizer of

$$\begin{aligned} \sum_{i=1}^n d_F^2(\mu, w_i) &= \sum_{i=1}^n \left\{ 1 - \frac{\int_0^l \mu(s)^* w_i(s) ds \int_0^l w_i(s)^* \mu(s) ds}{\int_0^l \mu(s)^* \mu(s) ds \int_0^l w_i(s)^* w_i(s) ds} \right\} \\ &= n - \sum_{i=1}^n \left\{ \frac{\mathbf{e}^* \mathbf{A} \mathbf{d}_i \mathbf{d}_i^* \mathbf{A} \mathbf{e}}{\mathbf{e}^* \mathbf{A} \mathbf{e} \mathbf{d}_i^* \mathbf{A} \mathbf{d}_i^*} \right\} = n - \frac{\mathbf{v}^* \mathbf{C} \mathbf{v}}{\mathbf{v}^* \mathbf{v}}, \end{aligned}$$

where  $\mathbf{A} = \mathbf{L}\mathbf{L}^T = \int_0^l \mathbf{h}(s) \mathbf{h}(s)^T ds$  is a positive definite matrix,  $\mathbf{L}$  is the lower triangular matrix resulting from a Cholesky decomposition of  $\mathbf{A}$ , and  $\mathbf{v} = \mathbf{L}\mathbf{e}$ . Since

$$S(\mu) = \sqrt{\int_0^l \mu(s)^* \mu(s) ds} = \sqrt{\mathbf{e}^* \int_0^l \mathbf{h}(s) \mathbf{h}(s)^T ds \mathbf{e}} = \sqrt{\mathbf{e}^* \mathbf{A} \mathbf{e}} = \|\mathbf{L}\mathbf{e}\|$$

we have  $\mathbf{L}\hat{\mathbf{e}} = \arg \sup_{\|\mathbf{v}\|=1} \mathbf{v}^* \mathbf{C} \mathbf{v}$  and therefore  $\hat{\mathbf{e}} = \mathbf{L}^{-1} \hat{\mathbf{v}}$   $\square$

Hence,  $\hat{\mu}$  is given by the coefficients  $\hat{\mathbf{e}}$  obtained as the complex eigenvector corresponding to the largest eigenvalue of  $\mathbf{C}$ . Again rotations of  $\hat{\mathbf{e}}$  also yield solutions, but all corresponding to the same curve shape.

## 2.4 Functional Curve Shape Parameterization

We have not yet discussed the choice of curve parameterization. In some situations the parameterization may be given by the nature of the data at hand. In other situations the natural parameterization of curve length is appropriate. In many situations the correspondence between curves are given at finite series of landmarks. These may be manually or automatically identified geometrical and anatomical landmarks.

Let there be given  $n$  closed curves,  $y_i$ ,  $i = 1, \dots, n$  with the constraints  $y_i(\xi_0) = y_{ik}$ ,  $y_i(\xi_j) = y_{ij}$  for  $j = 1, \dots, k$ , where  $0 = \xi_0 \leq \xi_1 \leq \dots \leq \xi_k = l$ . Let the curve lengths between landmarks and the total length of the  $i$ th curve be

$$l_{ij} = \int_{\xi_{j-1}}^{\xi_j} |y'(s)| ds, \quad L_i = \int_{\xi_0}^{\xi_k} |y'(s)| ds = \sum_{j=1}^k l_{ij},$$

then a parameterization  $s \in [0, l]$  based on normalized average curve length between landmarks is given by

$$\xi_0 = 0, \quad \xi_j = \xi_{j-1} + \frac{l}{n} \sum_{i=1}^n l_{ij} / L_i,$$

This parameterization is based on normalizing the curve segments  $l_{ij}$  with respect to the length of each curve. Instead of these normalized curve segment lengths we could use the average curve segment length from the Procrustes aligned curves,  $l_{ij}^P$ , i.e.

$$\xi_0^P = 0, \quad \xi_j^P = \xi_{j-1}^P + l \sum_{i=1}^n l_{ij}^P / \sum_{i=1}^n L_i^P = \xi_{j-1}^P + l l_{ij}^P / L_i^P,$$

$l_{ij}^P$  is the curve segment lengths for the Procrustes mean curve. However, because parameterization precedes Procrustes alignment this has to be done iteratively.

Let us consider at set of outlines of hands, 10 images each of 4 individuals were taken<sup>1</sup>. 56 landmarks are chosen as is shown in Figure 1(a). We will compare Procrustes alignment based on landmark and functional approaches.

Given landmarks  $\mathbf{y} = (y_1, \dots, y_k)^T$ ,  $k = 56$ , we employ a periodic linear spline,  $y(s)$ ,  $\xi_0 \leq s \leq \xi_k$  – i.e. a second order B-spline – to interpolate the points  $(\xi_0, y_k), (\xi_1, y_1), \dots, (\xi_k, y_k)$ . The parameterization is based on the average curve segment length in the Procrustes aligned curves, and is determined iteratively.

Let  $B_{i,2}(s)$  be the  $k+1$  second order B-spline basis functions corresponding to knots  $\xi$  (cf. [5]). Then we have

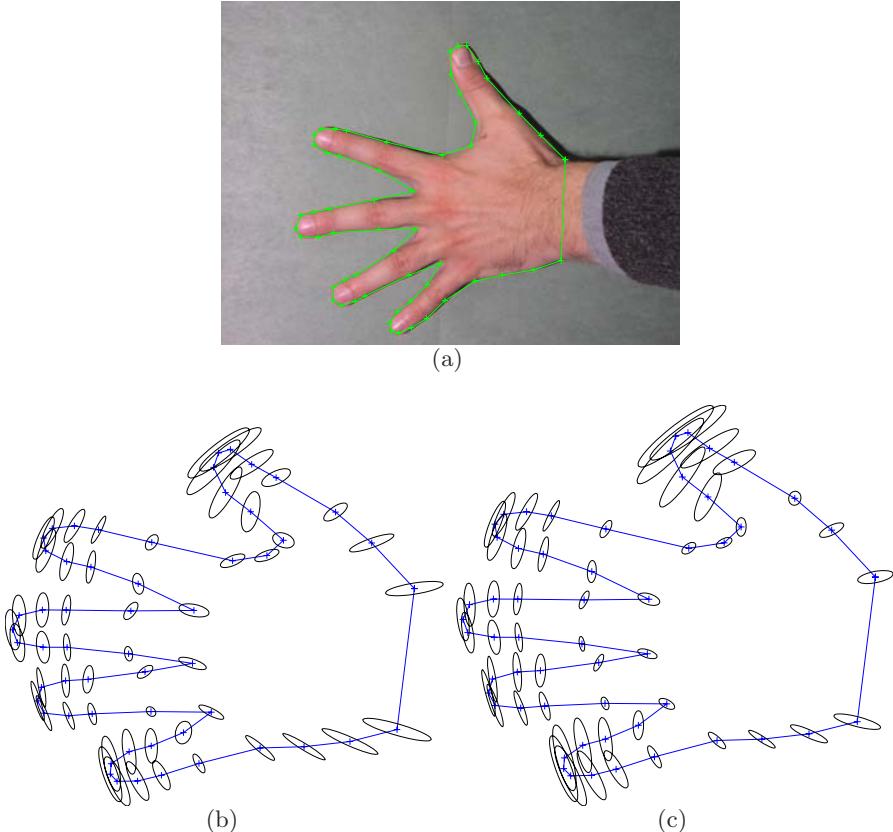
$$h_i(s) = B_{2,i+1}(s) \quad i = 1, \dots, k-1; \quad h_k(s) = B_{2,1}(s) + B_{2,k+1}(s)$$

Fitting the data  $y(\xi_0) = y_k$ ,  $y(\xi_j) = y_j$  for  $j = 1, \dots, k$  yields a linear system of equations to which the solution trivially is  $\hat{c}_i = y_i$ , or in vector notation  $\hat{\mathbf{c}} = \mathbf{y}$ . Now  $\mathbf{w} = \int_{\xi_0}^{\xi_k} \mathbf{h}(s) ds$  and  $\mathbf{A} = \int_{\xi_0}^{\xi_k} \mathbf{h}(s) \mathbf{h}(s)^T ds$  can be determined. For arbitrary knot sequences  $\xi = \{\xi_0, \dots, \xi_k\}$  with knot spacing  $d_i = \xi_i - \xi_{i-1}$ .

$$\mathbf{w} = \frac{1}{2} \begin{bmatrix} d_1+d_2 \\ d_2+d_3 \\ \vdots \\ d_{k-1}+d_k \\ d_k+d_1 \end{bmatrix} \quad \mathbf{A} = \frac{1}{6} \begin{bmatrix} 4(d_1+d_2) & d_2 & & & d_1 \\ d_2 & 4(d_2+d_3) & d_3 & & \\ & \ddots & \ddots & \ddots & \\ & & d_{k-1}4(d_{k-1}+d_k) & d_k & \\ & & & d_k & 4(d_k+d_1) \end{bmatrix}$$

---

<sup>1</sup> The data are available from [www.imm.dtu.dk/~aam](http://www.imm.dtu.dk/~aam)



**Fig. 1.** (a) 56 landmarks on the outline of a hand. Landmark based (b) and functional curve (c) Generalized Procrustes Alignment of hand images. The scatter of the full Procrustes fits of the curves at each landmark is shown by a contour ellipse at 1 standard deviation

In Figure 1 the results of landmark based and functional generalized Procrustes analysis of the hand data are shown. Segments where landmarks are relatively more dispersed receive larger weight in the functional analysis, hence the scatter becomes relatively smaller than compared with the landmark based approach.

## 2.5 Surface Generalized Procrustes Analysis

For the alignment of surface patches,  $y(s), w(s) : \Omega \rightarrow \mathbb{C}$  similar results as for curve shapes exist. The only difference being that all integrals are over  $\Omega$  instead of  $[0, l]$ . As for the curve shapes a crucial element is the parameterization of the surface shapes. Again we will discuss a parameterization derived from a finite series of landmarks. Let there be given  $n$  surface shapes,  $y_i$ ,  $i = 1, \dots, n$  with

landmarks  $y_i(\xi_j) = y_{ij}$  for  $j = 1, \dots, k$ , where  $\xi_j \in \Omega$  is the surface parameter for the  $j$ th landmark. Obtaining a parameterization is closely related to defining a warp function between the shapes under consideration. A warp function is a spatial transformation from one spatial configuration into another. [6] give a survey of warping methods. We will consider the simplest construction of a warp by assuming a piece-wise affine function. We will base this function on a mesh constructed by a Delaunay triangulation.

We begin by choosing a reference shape. This reference shape may be the landmark based Procrustes mean shape estimated from the set of landmarks on all shapes. We approximate the outline of the surface area of the reference shape by linear splines of (some of) the landmarks. Having done this we can partition the surface area by the set of triangles of a Delaunay triangulation that reside inside the outline of the surface area. The Delaunay triangulation partitions the convex hull of the landmarks. However, we only retain those triangles inside the outline of the surface area. Knowing the order of vertices of the surface area outline it is easily determined by inspection of the triangle vertices order if a triangle is inside the surface area.

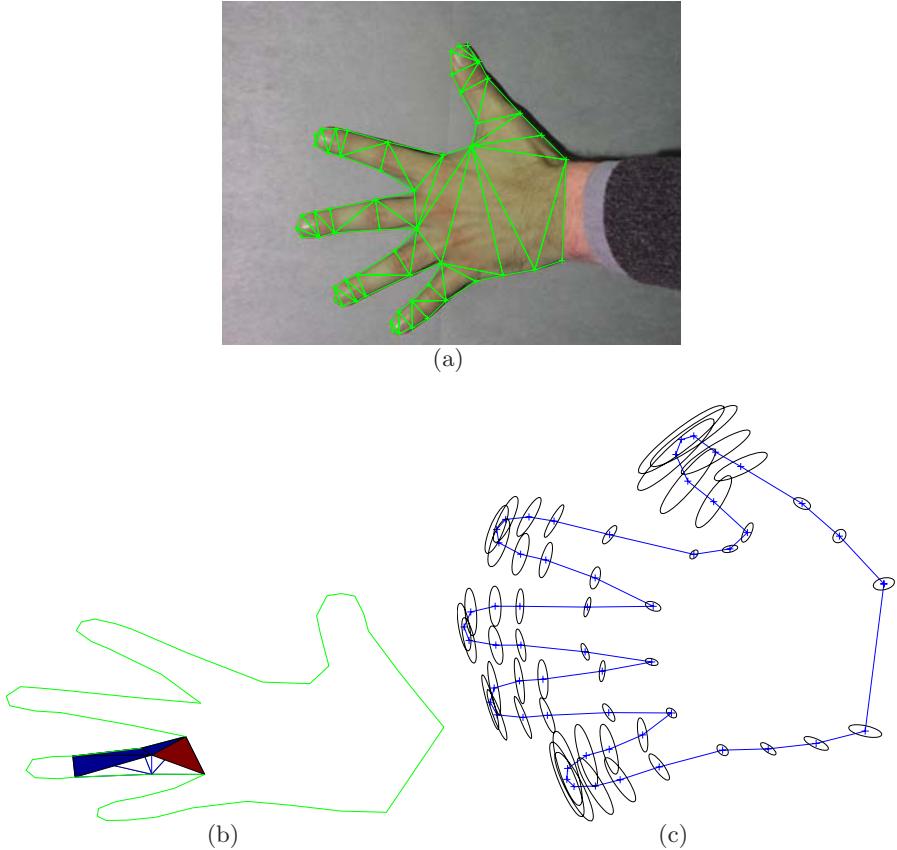
Now, a crucial assumption is that by applying the reference shape Delaunay triangulation to each of the  $n$  shapes of the data set we will obtain a one-to-one mapping. This will generally not be true. However, for shape sets with low variability this is not an unreasonable condition and it can easily be tested by examining triangle normals. Let  $\Omega$  be the surface patch of the complex plane corresponding to the reference surface shape, and let the parameterization of the shape data set be given by affinely warping the Delaunay triangles of the reference shape to the corresponding triangles of each of the surface shapes.

This procedure is realized by choosing a representation in based on  $M = k$  pyramidal basis functions. Each function is centered at a landmark where it has value 1; it is only non-zero in the Delaunay triangles in which that landmark is a vertex; it varies linearly within each triangle, and has value 0 at the 2 other vertices. In Figure 2(b) one basis function spanning in this case 4 Delaunay triangles is shown. The optimal coefficients in Equation (2) are trivially equal to the landmark coordinates. In order to determine the weight matrix,  $\mathbf{A}$  we partition the surface shape,  $\Omega$  into mutually exclusive patches given by the Delaunay triangles,  $\Omega_t$ ,  $t = 1, \dots, T$  i.e.

$$\mathbf{A} = \int_{\Omega} \mathbf{h}(s) \mathbf{h}^T(s) ds = \sum_{i=1}^T \int_{\Omega_i} \mathbf{h}(s) \mathbf{h}^T(s) ds$$

Let  $\phi_i : B \rightarrow \mathbb{C}$  be an affine function that maps  $B : 0 \leq u \leq 1, 0 \leq v \leq u$  to the  $i$ 'th Delaunay triangle. Then with this Delaunay triangle the basis functions that are non-zero here are equal to one of these element functions

$$\left. \begin{array}{l} f_1(\phi_i(u, v)) = u \\ f_2(\phi_i(u, v)) = v \\ f_3(\phi_i(u, v)) = 1 - u - v \end{array} \right\} \quad \text{for } 0 \leq u \leq 1, 0 \leq v \leq u$$



**Fig. 2.** (a) Annotated hand with Delaunay triangulation of landmarks; (c)  $m$ th basis function; (c) Surface based functional Procrustes alignment of the hand images. he scatter of the full Procrustes fits of the curves at each landmark is shown by a contour ellipse at 1 standard deviation. Compare Figure 1

For those pairs of basis functions  $(j, k)$  that are non-zero within the  $i$ th Delaunay triangle, let  $\tau_{ij}, \tau_{ik} \in \{1, 2, 3\}$  identify which elementar function the  $j$ th and  $k$ th basis functions consist of. Then we have

$$\int_{\Omega_i} h_j(s) h_k(s) ds = \begin{cases} \text{Ar}(\Omega_i)/6 & \text{for } j = k \\ \text{Ar}(\Omega_i)/12 & \text{for } j \neq k \end{cases}$$

Now within each Delaunay triangle 3 basis functions are non-zero. We can compute their contributions from Equation (3) and update  $\mathbf{A}$  accordingly.

In Figure 2(a) based on the landmarks shown in Figure 1(a) the triangles of the Delaunay triangulation that belong to the interior of the hand are shown. The Delaunay triangulation is determined from a landmark based Procrustes mean of the hand data set. Following the procedure described above we arrive

at the surface shape Procrustes alignment illustrated in Figure 2(c). Compared to the landmark based and curve shape based alignments shown in Figure 1 we obtain an even better alignment of the bulk of the hand. The major part of the variation is transferred to the fingers/fingertips.

### 3 Conclusion

We have derived procrustes methods for the alignment of curve and surface shapes based on functional representations using arbitrary parameterizations. In particular we have examined natural curve parameterizations. We have demonstrated that functional representations based on natural parameterizations and functional procrustes methods result in more intuitive alignment of sets of shapes.

### References

1. Kendall, D.G.: The diffusion of shape. *Advances in Applied Probability* **9** (1977) 428–430
2. Gower, J.C.: Generalized Procrustes analysis. *Psychometrika* **40** (1975) 33–50
3. ten Berge, J.M.F.: Orthogonal Procrustes rotation for two or more matrices. *Psychometrika* **42** (1977) 267–276
4. Goodall, C.: Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society, Series B* **53** (1991) 285–339
5. Nielsen, H.B.: Cubic splines (1998)
6. Glasbey, C.A., Mardia, K.V.: A review of image warping methods. *Journal of Applied Statistics* **25** (1998) 155–171

# The Descriptive Approach to Image Analysis Current State and Prospects

I. Gurevich

Department of Mathematical and Applied Techniques for Image Analysis and  
Nonlinear Problems,  
Dorodnicyn Computing Centre of the Russian Academy of Sciences,  
40, Vavilov str., Moscow GSP-1 119991. The Russian Federation  
[igourevi@ccas.ru](mailto:igourevi@ccas.ru)

**Abstract.** The presentation is devoted to the research of mathematical fundamentals for image analysis and recognition procedures. The final goal of this research is automated image mining: a) automated design, test and adaptation of techniques and algorithms for image recognition, estimation and understanding; b) automated selection of techniques and algorithms for image recognition, estimation and understanding; c) automated testing of the raw data quality and suitability for solving the image recognition problem. The main instrument is the Descriptive Approach to Image Analysis, which provides: 1) standardization of image analysis and recognition problems representation; 2) standardization of a descriptive language for image analysis and recognition procedures; 3) means to apply common mathematical apparatus for operations over image analysis and recognition algorithms, and over image models. It is shown also how and where to link theoretical results in the foundations of image analysis with the techniques used to solve application problems.

## 1 Introduction

Automation of image processing, analysis, estimating and understanding is one of the crucial points of theoretical computer science having decisive importance for applications, in particular, for diversification of solvable problem types and for increasing the efficiency of its solving.

The presentation is devoted to the research of mathematical fundamentals for image analysis and recognition procedures being conducted previously in the Scientific Council “Cybernetics” of the Russian Academy of Sciences, Moscow, Russian Federation, and currently in the Dorodnicyn Computing Centre of the Russian Academy of Sciences, Moscow, Russian Federation.

The final goal of this research is automated image mining. The main instrument is the Descriptive Approach to Image Analysis [1,2], which provides: 1) specialization of Zhuravlev’s Algebra [8] for an image recognition case; 2) standardization of image analysis and recognition problems representations; 3) standardization of a descriptive language for image analysis and recognition procedures; 4) means to apply common mathematical apparatus for operations over image analysis and recognition algorithms, and over image models [3].

Taking as a strategic goal the automated image mining it is necessary to provide image analysis professionals and final users with the following opportunities:

- automated design, test and adaptation of techniques and algorithms for image recognition, estimation and understanding;
- automated selection of techniques and algorithms for image recognition, estimation and understanding;
- automated testing of the raw data quality and suitability for solving the image recognition problem;
- standard technological schemes for image recognition, estimation, understanding and retrieval.

We shall outline the goals of theoretical development in the framework of the Descriptive Approach (and image analysis algebraization) (“What for”), the tool to achieve this goal (“How”), state of the art in the field (prospective trends), necessary steps to finalize the Descriptive Approach (“What to Do or What to be Done”) and the global problem of an image reduction to a recognizable form. It will be shown also how and where to link theoretical results in the foundations of image analysis with the techniques used to solve application problems.

The structure of the paper is as follows:

1. What for
2. How
  - The Tool - Descriptive Approach
3. State of the Art:
  - Plurality and Fusion
  - Multialgorithmic Classifiers
  - Multimodel Representations
4. What to Do or What to be Done. Basic Steps:
  - Step 1. Mathematical Settings of an Image Recognition Problem
  - Step 2. Image Models
  - Step 3. Multimodel Representation of Images
  - Step 4. Image Equivalence
  - Step 5. Image Metrics
  - Step 6. Descriptive Image Algebras.
5. Conclusion
- Image Reduction to a Recognizable Form.

## 2 What for?

The image analysis and recognition techniques and tools are destined for solving of the following basic classes of applied image analysis problems:

1. Image matching for classification with an image, a set of images and a series of images.
2. Searching an image for some regularity / irregularity / object / token / fragment / primitive of arbitrary or prescribed type/form.

3. Clusterization of an image set.
4. Image segmentation (for homogeneous regions, groups of objects, selection of features).
5. Automatic selection of image primitives, specific objects, feature objects, logical and spatial relations.
6. Image reduction to a recognizable form.
7. Reconstruction and Restoration of missed frames in an image series and of images by fragments, primitives, generative procedures and context.
8. Image analysis problem decomposition and synthesis.

The most important – critical points of an applied image analysis problem solving are as follows:

- CPI. Precise setting of a problem (Step 1).
- CPII. Correct and “computable” representation of raw and processed data for each algorithm at each stage of processing (Step 2, Step 5).
- CPIII. Automated selection of an algorithm (Step 1, Step 3, Step 6):
  - CPIII-1. Decomposition of the solution process for main stages (Step 1, Step 6);
  - CPIII-2. Indication of the points of potential improvement of the solution (“branching points”) (Step 1, Step 6);
  - CPIII-3. Collection and application of problem solving experience (Step 3, Knowledge Base);
  - CPIII-4. Selection for each problem solution stage of basic algorithms, basic operations and basic models (operands) (Step 6);
  - CPIII-5. Classification of the basic elements (Step 3, thesaurus).
- CPIV. Performance evaluation at each step of processing and of the solution (Step 2, Step 4, Step 6).
  - CPIV-1. Analysis, estimation and utilization of the raw data specificity (Step 2);
  - CPIV-2. Diversification of mathematical tools used for performance evaluation (Step 6);
  - CPIV-3. Reduction of raw data to the real requirements of the selected algorithms (Step 4).

The further development of the Descriptive Approach should provide necessary means for implementing of these steps. After each Critical Points in the brackets are indicated the corresponding “next steps” (the description of the steps see below).

So, the success of image mining in a whole is connected with overcoming of the Critical Points in a following way:

- automated design, test and adaptation of techniques and algorithms for image recognition, estimation and understanding (CPIV);
- automated selection of techniques and algorithms for image recognition, estimation and understanding (CPIII);
- automated testing of the raw data quality and suitability for solving the image recognition problem (CPII);
- standard technological schemes for image recognition, estimation, understanding and retrieval (CPI).

### 3 How?

Mathematical fundamentals for image processing and image analysis procedures are constructed in the framework of the Descriptive Approach to Image Analysis, which provides:

- specialization of Zhuravlev's Algebra for an image recognition case (CP1);
- standardization of image analysis and recognition problems representation (CP1);
- standardization of a descriptive language for image analysis and recognition procedures (CP2);
- means to apply common mathematical apparatus for operations over image analysis and recognition algorithms, and over image models (CPIV).

The Descriptive Approach is based on:

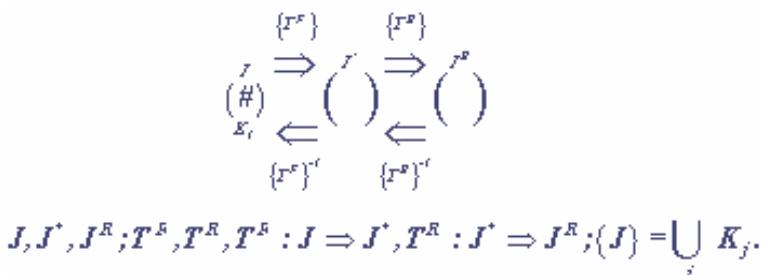
- descriptive model of image recognition procedures (CPI);
- image reduction to a recognizable form (CPII);
- image models (CPII);
- algebraization of image mining (CPIII, CPIV);
- generative principle and bases of transforms and models.

The preliminary condition of algebraization of image mining is development of formal systems for image representation and transformation satisfying to the following conditions: a) each object is a hierarchical structure constructed by a set of operations of image algebra [5] applied to the set of elements of images; b) the objects are points, sets, models, operations, morphisms; c) each transform is a hierarchical structure constructed by a set of operations of image algebra on the set of basic transforms.

The Descriptive Approach provides construction and application of two types of such formal systems - special versions of algebras - image algebras (CPIII, CPIV) [7] and descriptive image algebras (CPIII, CPIV) [4-6].

Exploitation of the Generative principle and bases of transforms and models provides for decomposition of a problem into primitive tasks, establishing of the correspondence between basic primitive tasks and basic primitive transforms and combining of basic algorithms and models.

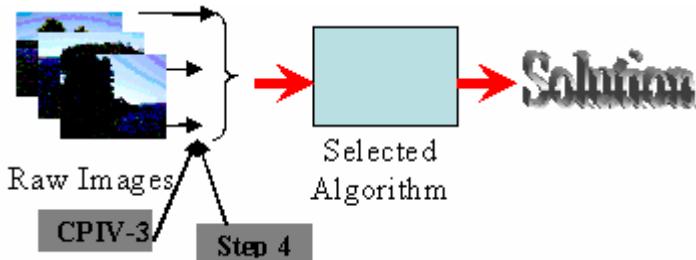
The corner-stone of the Descriptive Approach is a model of image recognition procedures (Fig. 1).



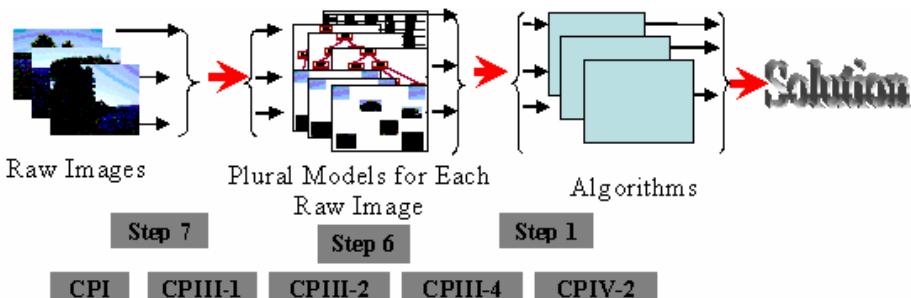
**Fig. 1.** Descriptive model of image recognition procedures

## 4 State of the Art

The current trends in image recognition are connected with plurality and fusion of image recognition and image data, use of multiple classifiers and of multiple model representations of the images under processing. The classical and modern versions of image recognition schema are represented in Fig. 2 and Fig. 3.



**Fig. 2.** Image recognition. Classical case



**Fig. 3.** Image recognition. General case (multiple classifiers and model plurality)

## 5 What to Do or What to Be Done

In this section we shall outline the basic “next steps” necessary to finalize the Descriptive Approach and indicate what is done and what to be done for each of the steps. These steps are as follows:

- Step 1. Mathematical Settings of an Image Recognition Problem (CPI, CPIII-1, CPIII-2);
- Step 2. Image Models (CPII, CPIV-1);
- Step 3. Multiple Model Representation of Images CPIII-3, CPIII-5);
- Step 4. Image Equivalence (CPIV-3);
- Step 5. Image Metrics (CPII);
- Step 6. Descriptive Image Algebras (CPIII-1, CPIII-2, CPIII-4, CPIV-2).

## 5.1 Step 1. Mathematical Settings of an Image Recognition Problem

Done:

- CPIII-1 - Descriptive Model of Image Recognition Procedures;
- CPIII-1 - Mathematical Setting of an Image Recognition Problem. Image Equivalence Case.

To Be Done:

- CPIII-2 - Establishing of interrelations and mutual correspondence between image recognition problem classes and image equivalence classes;
- CPI - New mathematical settings of an image recognition problem connected with image equivalency;
- CPI - New mathematical settings of an image recognition problem connected with an image multiple model representation and image data fusion.

## 5.2 Step 2. Image Models

Done:

- CPII - Main types of image models were introduced and defined;
- CPII - It was shown which types of image models are generated by the main versions of descriptive image algebras with one ring.

To Be Done:

- CPIV-1 - Creation of image models catalogue;
- CPIV-1 - Selection and study of basic operations on image models for different types of image models (including construction of bases of operations);
- CPIV-1 - Use of information properties of images in image models;
- CPIV-1 - Study of multiple model representations of images.

## 5.3 Step 3. Multiple Model Representation of Images

Done:

- CPIII-3 - Generating Descriptive Tree (GDT) - a new data structure for generation plural models of an image is introduced.

To Be Done:

- CPIII-5 - to define and to specify GDT;
- CPIII-5 - to set up image recognition problem using GDT;
- CPIII-5 - to define descriptive image algebra using GDT;
- CPIII-5 - to construct a descriptive model of image recognition procedures based on GDT using;
- CPIII-5 - to select image feature sets for construction of P-GDT;
- CPIII-5 - to select image transform sets for construction of T-GDT;
- CPIII-5 - to define and study of criteria for selection of GDT-primitives.

An example of GDT is shown in Fig. 4.

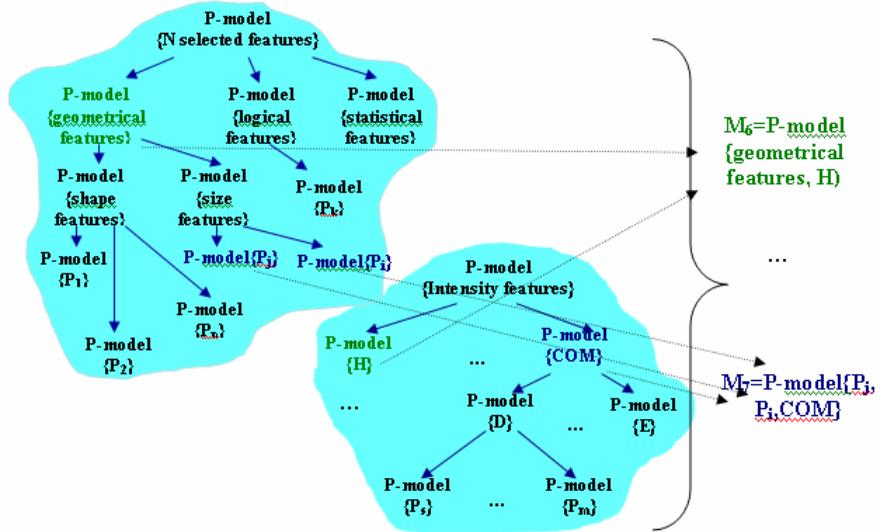


Fig. 4. Generating Descriptive Trees

#### 5.4 Step 4. Image Equivalence

Done:

There were introduced several types of image equivalence:

- image equivalence based on the groups of transformations;
- image equivalence directed at the image recognition task;
- image equivalence with respect to a metric.

To Be Done:

- CPIV-3 - to study image equivalence based on information properties of an image;
- CPIV-3 - to define and construct image equivalence classes using template (generative) images and transform groups;
- CPIV-3 - to establish and to study links between image equivalence and image invariance;
- CPIV-3 - to establish and to study links between image equivalence and appropriate types of image models;
- CPIV-3 - to establish and to study links between image equivalence classes and sets of basic image transforms.

#### 5.5 Step 5. Image Metrics

To Be Done:

- CPII - to study, to classify, to define competence domains of pattern recognition and image analysis metrics;

- CPII - to select workable pattern recognition and image analysis metrics;
- CPII - to construct and to study new image analysis-oriented metrics;
- CPII - to define an optimal image recognition-oriented metric;
- CPII - to construct new image recognition algorithms on the base of metrics generating specific image equivalence classes.

## 5.6 Step 6. Descriptive Image Algebras

Done:

- CPIII-1 - Descriptive Image Algebras (DIA) with a single ring were defined and studied (basic DIA);
- CPIII-2 - it was shown which types of image models are generated by main versions of DIA with a single ring (see Fig. 5);
- CPIII-4 - the technique for defining and testing of the necessary and sufficient conditions for generating DIA with a single ring by a set of image processing operations were suggested;
- the necessary and sufficient conditions for generating basic DIA with a single ring were formulated;
- CPIV-2 - the hierarchical classification of image algebras was suggested (see Fig. 6);
- it was proved that the Ritter's algebra could be used for construction DIA without a "template object".

To Be Done:

- CPIII-1 - to study DIA with a single ring, whose elements are image models;
- CPIII-2 - to study DIA with several rings (super algebras);
- CPIII-2 - to define and study of DIA operation bases;
- CPIII-4 - to construct standardized algebraic schemes for solving image analysis and estimation problems on the DIA base;
- CPIV-2 - to generate DIA using equivalence and invariance properties in an explicit form;
- to demonstrate efficiency of using DIA in applied problems.

	Elements of the Ring	Operations of the Ring	Result (type of a model)
1	Operations for computation of numerical features	Standard algebraic operations	P-models
2	Images	Standard algebraic operations	Images
3	Image algebra operations	Standard algebraic operations	G-, T-models, images, image fragments
4	Standard algebraic operations with parameters	Image algebra operations	G-, T-models, images, image fragments
5	Images and image representations	Image algebra operations	G-, T-models, images, image fragments

Fig. 5. Generation of image models by DIA

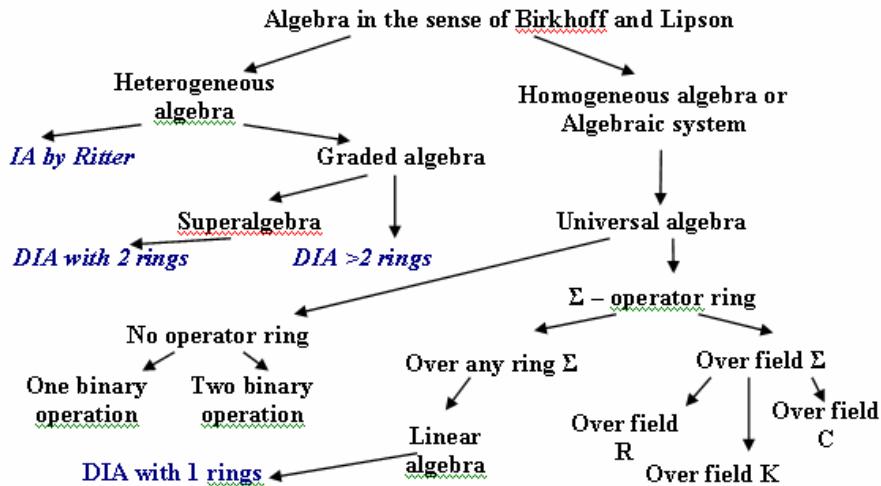


Fig. 6. Hierarchy of algebras

## 6 Conclusion

In principle, the success of image analysis and recognition problem solution depends mainly on the success of image reduction to a recognizable form, which could be accepted by an appropriate image analysis/recognition algorithm. All above mentioned steps contribute to the development techniques for this kind of image reduction/image modeling. It appeared that an image reduction to a recognizable form is a critical issue for image analysis applications, in particular for qualified decision making on the base of image mining. The main tasks and problems of an image reduction to a recognizable form are listed below:

1. Formal Description of Images: 1) study and construction of image models (Step 2); 2) study and construction of multiple model image representations (Step 3); 3) study and construction of metrics (Step 5).
2. Description of Image Classes Reducible to a Recognizable Form: 1) introduction of new mathematical settings of an image recognition problem (Step 1); 2) establishing and study of links between multiple model representation of images and image metrics (Steps 3, 5); 3) study and use of image equivalencies (Step 4).
3. Development, Study and Application of an Algebraic Language for Description of the Procedures of an Image Reduction to a Recognizable Form (Step 6).

After passing through the above mentioned steps it would be possible to formulate the axiomatics of the descriptive (mathematical) theory of image analysis.

## Acknowledgments

The paper was partially supported by the Russian Foundation for Basic Research, Grant No. 05-01-00784 and by the grant “Descriptive Image Algebras” (Program of Basic Research of the Department of Mathematical Sciences of the RAS).

## References

1. Gurevich, I.B.: The Descriptive Framework for an Image Recognition Problem. Proceedings of The 6th Scandinavian Conference on Image Analysis (Oulu, June 19 - 22, 1989): in 2 volumes. - Pattern Recognition Society of Finland, vol. 1 (1989), 220 - 227.
2. Gurevich, I.B.: Descriptive Technique for Image Description, Representation and Recognition. Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications in the USSR, vol.1, no. 1 (1991), 50 - 53.
3. Gurevich, I.B., Yashina, V.V.: Application of algebraic language in image analysis. Illustrative example. Proceedings of the 7th International conference "Pattern Recognition and Image Analysis: New Information Technologies" (PRIA-7-2004), St. Petersburg, Russian Federation, vol. 1 (2004), 240-243.
4. Gurevich, I.B., Yashina, V.V.: Conditions of Generating Descriptive Image Algebras by a Set of Image Processing Operations. Progress in Pattern Recognition, Speech and Image Analysis. Proceedings of the 8th Iberoamerican Congress on Pattern Recognition, November 26-29, 2003, Havana, Cuba, CIARP'2003, LNCS 2905, Springer-Verlag Berlin Heidelberg (2003), 498-505.
5. Gurevich, I.B., Yashina, V.V.: Descriptive Image Algebras with One Ring. Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Application, vol. 13, no.4 (2003), 579-599.
6. Gurevich, I.B., Zhuravlev, Y.I.: An Image Algebra Accepting Image Models and Image Transforms. Proceedings of the 7th International Workshop "Vision, Modeling, and Visualization 2002" (VMV2002), November 20 – 22, 2002, Erlangen, Germany, IOS Press, B.V.Amsterdam, Infix, Akademische Verlagsgesellschaft, Aka GMBH, Berlin (2002), 21 – 26.
7. Ritter, G.X., Wilson, J.N.: Handbook of Computer Vision Algorithms in Image Algebra, 2-d Edition. CRC Press Inc. (2001).
8. Zhuravlev, Yu.I.: An Algebraic Approach to Recognition or Classification Problems. Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, vol.8, no. 1 (1998), 59 – 100.

# A New Method for Affine Registration of Images and Point Sets

Juho Kannala<sup>1</sup>, Esa Rahtu<sup>1</sup>, Janne Heikkilä<sup>1</sup>, and Mikko Salo<sup>2</sup>

<sup>1</sup> Machine Vision Group,

Department of Electrical and Information Engineering,

University of Oulu, P.O.Box 4500,

90014 University of Oulu, Finland

{jkannala, erahtu, jth}@ee.oulu.fi

<sup>2</sup> Rolf Nevanlinna Institute,

Department of Mathematics and Statistics,

University of Helsinki, P.O.Box 68,

00014 University of Helsinki, Finland

msa@rni.helsinki.fi

**Abstract.** In this paper we propose a novel method for affine registration of images and point patterns. The method is non-iterative and it directly utilizes the intensity distribution of the images or the spatial distribution of points in the patterns. The method can be used to align images of isolated objects or sets of 2D and 3D points. For Euclidean and similarity transformations the additional constraints can be easily embedded in the algorithm. The main advantage of the proposed method is its efficiency since the computational complexity is only linearly proportional to the number of pixels in the images (or to the number of points in the sets). In the experiments we have compared our method with some other non-feature-based registration methods and investigated its robustness. The experiments show that the proposed method is relatively robust so that it can be applied in practical circumstances.

## 1 Introduction

The traditional approach to image registration is to first extract some salient features from both images and then find correspondence between these features. Thereafter the feature correspondences are used to recover the geometric transformation that registers the images [1]. The problem with this approach is that it is not always possible to find a sufficient number of features which can be localized accurately from both images and matched reliably between them. In addition, when registering point patterns an essential part of the problem is that correspondences between the point sets are unknown [2].

A different approach to registration is represented by methods that directly utilize the intensity information of the images. Unfortunately, there are quite a few direct methods for affine registration. The maximization of mutual information [3, 4] is an intensity-based technique that can be applied also in the case of affine transformations. Though being quite a robust method it is not com-

putationally very simple and it requires iterative optimization techniques. Two other possible methods for affine registration are the cross-weighted moments [5] and affine moment descriptors [6]. However, the cross-weighted moment method is somewhat cumbersome for large images due to its computational complexity. All these three methods, the maximization of mutual information, cross-weighted moments and affine moment descriptors can be used to register both images and point patterns [4, 5, 6].

In this paper we introduce an entirely new intensity-based method for affine registration. The method is based on a novel image transform which was recently developed to produce global affine invariant features directly from gray-scale images [7]. A slight modification of the transform makes it possible to compute such descriptor values that allow to recover the affine transformation parameters between the images. The method can be additionally extended to registration of point sets. The structure of the paper is as follows. In Section 2 we give background for the proposed registration method by first describing the affine invariant image transform behind it. The actual registration method is then described in Section 3 and the implementation is covered in Section 4. The experiments and results are presented and discussed in Sections 5 and 6.

## 2 Background

Recently a new affine invariant image transform was proposed for efficient computation of global affine invariant features from grayscale images [7]. This transform is the basis for our affine registration method, and hence, it is described in the following.

Let  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  be a compactly supported image function, whose centroid is  $\mu(f) = \int_{\mathbb{R}^2} \mathbf{x} f(\mathbf{x}) d\mathbf{x} / \|f\|_{L^1}$ . We define the normalized function  $\tilde{f}(\mathbf{x}) = f(\mathbf{x} + \mu(f))$  by shifting the coordinate origin to the centroid of the image. Clearly, the normalized representation is invariant to translations of the image.

Then the image transform [7] is defined as follows and its affine invariance is proven below.

**Definition 1.**  $f \in L^\infty(\mathbb{R}^2)$ ,  $\alpha, \beta \in \mathbb{R}$

$$If(\alpha, \beta) = \frac{1}{\|f\|_{L^1}} \int_{\mathbb{R}^2} \tilde{f}(\mathbf{x}) \tilde{f}(\alpha\mathbf{x}) \tilde{f}(\beta\mathbf{x}) d\mathbf{x}. \quad (1)$$

**Proposition 1.**  $If(f \circ \mathcal{A}^{-1}) = If$

We first show that  $I$  is translation invariant. If  $g(\mathbf{x}) = f(\mathbf{x} - \mathbf{x}_0)$  then  $\|g\|_{L^1} = \|f\|_{L^1}$  and  $\mu(g) = \mu(f) + \mathbf{x}_0$ . Thus  $\tilde{g}(\mathbf{x}) = g(\mathbf{x} + \mu(g)) = \tilde{f}(\mathbf{x})$ , and consequently  $Ig = If$ . Also, if  $\mathbf{A}$  is a nonsingular matrix let  $g(\mathbf{x}) = f(\mathbf{A}^{-1}\mathbf{x})$ . Since  $\mu(g) = \mathbf{A}\mu(f)$  one has  $\tilde{g}(\mathbf{x}) = f(\mathbf{A}^{-1}(\mathbf{x} + \mathbf{A}\mu(f))) = \tilde{f}(\mathbf{A}^{-1}\mathbf{x})$  and

$$Ig(\alpha, \beta) = \frac{1}{|\det \mathbf{A}| \|f\|_{L^1}} \int_{\mathbf{R}^2} \tilde{f}(\mathbf{A}^{-1}\mathbf{x}) \tilde{f}(\alpha\mathbf{A}^{-1}\mathbf{x}) \tilde{f}(\beta\mathbf{A}^{-1}\mathbf{x}) d\mathbf{x}.$$

The change of variables  $\mathbf{x} \mapsto \mathbf{Ax}$  gives that  $Ig = If$ .

One obtains more general global affine invariants in the following way. Let  $H : \mathbb{R}^k \rightarrow \mathbb{R}$  be a measurable function, and define

$$\hat{I}f(\alpha_1, \dots, \alpha_k) = \frac{1}{\|f\|_{L^1}} \int_{\mathbb{R}^2} H(\tilde{f}(\alpha_1 \mathbf{x}), \dots, \tilde{f}(\alpha_k \mathbf{x})) d\mathbf{x}. \quad (2)$$

If  $H$  satisfies some conditions (e.g.  $H$  continuous and  $H(\mathbf{0}) = 0$ ), then the expression is well defined for compactly supported  $f \in L^\infty(\mathbf{R}^2)$ . The proof of Proposition 1 shows that  $\hat{I}$  is also affine invariant. By changing variables  $\mathbf{x} \mapsto \alpha_1^{-1} \mathbf{x}$  we see that no information is lost if we normalize  $\alpha_1 = 1$ . In (1) we chose  $k = 3$  and  $H(x, y, z) = xyz$  and made the normalization  $\alpha_1 = 1$ .

The formula (1) provides a method for producing affine invariant features from a grayscale image and in [7] these features were used to classify affine transformed images which were additionally corrupted by noise and occlusion. It was observed that already a small subset of the features was enough for successful classification.

### 3 Registration Method

The above approach may also be utilized in affine registration after a slight modification. In the following we propose a transform which produces such descriptors in  $\mathbb{R}^2$  that allow to recover the transformation between an image and its affine transformed version.

**Definition 2.**  $f \in L^\infty(\mathbb{R}^2)$ ,  $\alpha, \beta \in \mathbb{R}$

$$\mathbf{J}f(\alpha, \beta) = \frac{1}{\|f\|_{L^1}} \int_{\mathbb{R}^2} \mathbf{x} \tilde{f}(\mathbf{x}) \tilde{f}(\alpha \mathbf{x}) \tilde{f}(\beta \mathbf{x}) d\mathbf{x}. \quad (3)$$

**Proposition 2.**  $\mathcal{A}(\mathbf{x}) = \mathbf{Ax} + \mathbf{t}$

$$f'(\mathbf{x}) = f(\mathcal{A}^{-1}(\mathbf{x})) \quad \mathbf{J}f' = \mathbf{A} \mathbf{J}f$$

We get  $f'(\mathbf{x}) = (f \circ \mathcal{A}^{-1})(\mathbf{x}) = f(\mathbf{A}^{-1}\mathbf{x} - \mathbf{A}^{-1}\mathbf{t})$ . Then, since  $\mu(f') = \mathbf{A}\mu(f) + \mathbf{t}$  and  $\|f'\|_{L^1} = |\det \mathbf{A}| \|f\|_{L^1}$ , we have  $\tilde{f}'(\mathbf{x}) = f'(\mathbf{x} + \mu(f')) = f(\mathbf{A}^{-1}(\mathbf{x} + \mu(f')) - \mathbf{A}^{-1}\mathbf{t}) = f(\mathbf{A}^{-1}\mathbf{x} + \mu(f)) = \tilde{f}(\mathbf{A}^{-1}\mathbf{x})$  and

$$\mathbf{J}f'(\alpha, \beta) = \frac{1}{|\det \mathbf{A}| \|f\|_{L^1}} \int_{\mathbb{R}^2} \mathbf{x} \tilde{f}(\mathbf{A}^{-1}\mathbf{x}) \tilde{f}(\alpha \mathbf{A}^{-1}\mathbf{x}) \tilde{f}(\beta \mathbf{A}^{-1}\mathbf{x}) d\mathbf{x}.$$

The change of variables  $\mathbf{x} \mapsto \mathbf{Ax}$  gives that  $\mathbf{J}f' = \mathbf{A}(\mathbf{J}f)$ .

The above proposition implies that by computing the transform values  $\mathbf{J}f'(\alpha_i, \beta_i)$  and  $\mathbf{J}f(\alpha_i, \beta_i)$  for at least two different pairs  $(\alpha_i, \beta_i)$  one obtains a

set of linear equations from which the transformation matrix  $\mathbf{A}$  may be solved. If more than two pairs are used, a linear least-squares solution of

$$\min_{\mathbf{A}} \sum_i \|\mathbf{J}f'(\alpha_i, \beta_i) - \mathbf{A}\mathbf{J}f(\alpha_i, \beta_i)\|^2 \quad (4)$$

can be computed. Thereafter the translation may be solved by  $\mathbf{t} = \boldsymbol{\mu}(f') - \mathbf{A}\boldsymbol{\mu}(f)$ .

In practice, with real images, it is better to use the normalized features  $\mathbf{J}f'/If'$  and  $\mathbf{J}f/If$  which are still related by the matrix  $\mathbf{A}$  since  $If' = If$ . Then it is sufficient that  $f'(\mathbf{x}) = sf(\mathcal{A}^{-1}(\mathbf{x}))$ , i.e., the intensity values in the affine transformed image may have different scale than in the original image. This leads to a better robustness against different illumination conditions.

The transform  $f \mapsto \mathbf{J}f$  has the following symmetries, which are obtained from (3) by changes of variables.

- Proposition 3.** (a)  $\mathbf{J}f(\alpha, \beta) = \mathbf{J}f(\beta, \alpha)$   
 (b)  $\mathbf{J}f(\alpha, \beta) = \alpha^{-3}\mathbf{J}f(1/\alpha, \beta/\alpha) \dots \alpha \neq 0$   
 (c)  $\mathbf{J}f(\alpha, \beta) = \beta^{-3}\mathbf{J}f(1/\beta, \alpha/\beta) \dots \beta \neq 0$

The symmetries may be used to show that it is enough to compute  $\mathbf{J}f(\alpha, \beta)$  in the triangle  $T$  defined by the vertices  $\{(-1, -1), (-1, 1), (1, 1)\}$ . The values  $\mathbf{J}f(\alpha, \beta)$  outside  $T$  may be computed from the values in  $T$  using the symmetries. The symmetries should be taken into account when choosing the points  $(\alpha_i, \beta_i)$  in order to avoid redundancy.

Analogously to Remark 1 also the transform (3) may be generalized so that the transformation property stated in Proposition 3 still holds. Moreover, we may generalize it for functions defined in  $\mathbb{R}^d$ . Especially interesting is the case  $d = 3$  since this allows to extend the above registration method to volume images. This is summarized in the following.

**Proposition 4.**  $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  . . . . .  $\mathcal{A}(\mathbf{x}) = \mathbf{Ax} + \mathbf{t}$  . . . . .  $\mathbf{A} \in \mathbb{R}^{d \times d}$  . . . . .  $f \in L^\infty(\mathbb{R}^d)$  . . . . .  $f \circ \boldsymbol{\mu}(f) = \int_{\mathbb{R}^2} \mathbf{x} f(\mathbf{x}) d\mathbf{x} / \|f\|_{L^1}$  . . . . .  $\tilde{f}(\mathbf{x}) = f(\mathbf{x} + \boldsymbol{\mu}(f))$  . . . . .  $H : \mathbb{R}^k \rightarrow \mathbb{R}$  . . . . .  $H(\mathbf{0}) = 0$  . . . . .

$$\hat{\mathbf{J}}f(\alpha_1, \dots, \alpha_k) = \frac{1}{\|f\|_{L^1}} \int_{\mathbb{R}^2} \mathbf{x} H(\tilde{f}(\alpha_1 \mathbf{x}), \dots, \tilde{f}(\alpha_k \mathbf{x})) d\mathbf{x}. \quad (5)$$

$$\hat{\mathbf{J}}(f \circ \mathcal{A}^{-1}) = \mathbf{A}(\hat{\mathbf{J}}f)$$

### 3.1 Euclidean and Similarity Transformations

Euclidean and similarity transformations are subgroups of affine transformations. In 2D, a general affine transformation has six degrees of freedom while a similarity transformation has four and a Euclidean transformation only three degrees of freedom. Hence, in such cases the matrix  $\mathbf{A}$  has to satisfy additional

constraints and instead of a general nonsingular matrix we should estimate a transformation of the form

$$\mathbf{A} = s\mathbf{R}, \quad (6)$$

where  $\mathbf{R}$  is a rotation matrix and  $s$  is a scaling factor, which is equal to 1 for Euclidean transformations.

Nevertheless, the constraint (6) may be easily embedded to the registration method described above. There we estimated the affine transformation matrix  $\mathbf{A}$  by first computing a set of correspondences  $\mathbf{J}f'(\alpha_i, \beta_i) \leftrightarrow \mathbf{J}f(\alpha_i, \beta_i)$  and then solving the resulting linear least-squares problem (4). However, the least-squares solution of (4) is possible in closed form also when the constraint (6) is forced. An algorithm for finding the least-squares solution when  $\mathbf{A}$  is forced to be a rotation matrix, i.e.  $s = 1$ , was described in [8] and it is based on the singular value decomposition. In [9] Umeyama made a correction to this method and presented a complete algorithm for the least-squares solution of similarity transformation parameters between two sets of  $d$ -dimensional points when the correspondences between the sets are known. Though we assume in (4) that there is no translation Umeyama's method can still be straightforwardly applied. Hence, we obtain the solution  $\mathbf{A} = s\mathbf{R}$  and the translation is then again solved by  $\mathbf{t} = \boldsymbol{\mu}(f') - \mathbf{A}\boldsymbol{\mu}(f)$ .

### 3.2 Registration of Point Sets

Assume that instead of an image we have a set of points,  $\mathbf{x}_i$ , and an affine transformed version of it,  $\mathbf{x}'_i = \mathcal{A}(\mathbf{x}_i) = \mathbf{A}\mathbf{x}_i + \mathbf{t}$ , and we would like to solve the transformation between the point patterns without knowing the point correspondences.

We consider that the points  $\mathbf{x}_i$  are random samples of a random variable  $\mathbf{X}$  with some probability density  $p$ . Then the points  $\mathbf{x}'_i$  are samples of  $\mathbf{X}'$  which has density  $p' = (p \circ \mathcal{A}^{-1})/\|p \circ \mathcal{A}^{-1}\|_{L^1}$ . We denote the mean and covariance of  $\mathbf{X}$  by  $\boldsymbol{\mu}$  and  $\mathbf{C}$ , and those of  $\mathbf{X}'$  by  $\boldsymbol{\mu}'$  and  $\mathbf{C}'$ . We define the normalized random variables by  $\tilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}$  and  $\tilde{\mathbf{X}}' = \mathbf{X}' - \boldsymbol{\mu}'$  and the corresponding densities are denoted by  $\tilde{p}$  and  $\tilde{p}'$ . Then it holds that  $\tilde{p}'(\mathbf{x}) = \tilde{p}(\mathbf{A}^{-1}\mathbf{x})/|\det \mathbf{A}|$ .

Furthermore, assume for the present that two functions,  $g$  and  $g'$ , are given so that  $g'(\mathbf{x}) = sg(\mathbf{A}^{-1}\mathbf{x})$  where  $s$  is an arbitrary nonzero scale. Now, instead of (3) we consider descriptors defined for  $\gamma \in \mathbb{R}$  as follows

$$\mathbf{H}(\gamma) = \frac{\int_{\mathbb{R}^2} \mathbf{x}g(\gamma\mathbf{x})\tilde{p}(\mathbf{x})d\mathbf{x}}{\int_{\mathbb{R}^2} g(\gamma\mathbf{x})\tilde{p}(\mathbf{x})d\mathbf{x}} = \frac{E[\tilde{\mathbf{X}}g(\gamma\tilde{\mathbf{X}})]}{E[g(\gamma\tilde{\mathbf{X}})]}, \quad (7)$$

where the integrals are interpreted as expectation values. In practice, as we do not know the function  $\tilde{p}$  we must compute (7) by using the sample means computed from the samples  $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ , where  $\bar{\mathbf{x}}$  is the centroid of the original points  $\mathbf{x}_i$ . The corresponding descriptors for the transformed pattern are  $\mathbf{H}'(\gamma) = E[\tilde{\mathbf{X}}'g'(\tilde{\mathbf{X}}')]/E[g(\tilde{\mathbf{X}}')]$  and it holds that  $\mathbf{H}'(\gamma) = \mathbf{A}\mathbf{H}(\gamma)$ . Thus, as above, a set of correspondences  $\mathbf{H}'(\gamma_i) \leftrightarrow \mathbf{H}(\gamma_i)$  could be used to solve  $\mathbf{A}$  if we only knew suitable functions  $g$  and  $g'$ . Thereafter, the translation can be solved from  $\mathbf{t} = \bar{\mathbf{x}}' - \mathbf{A}\bar{\mathbf{x}}$ .

The problem in choosing  $g$  and  $g'$  is that we do not know  $\mathbf{A}$  beforehand. However, we may estimate the covariances  $\mathbf{C}$  and  $\mathbf{C}'$  as sample covariances from the point sets  $\{\mathbf{x}_i\}$  and  $\{\mathbf{x}'_i\}$  and set

$$g(\mathbf{x}) = N(\mathbf{0}, \mathbf{C}), \quad g'(\mathbf{x}) = N(\mathbf{0}, \mathbf{C}'), \quad (8)$$

where  $N(\mathbf{0}, \mathbf{C})$  is the zero mean Gaussian distribution with covariance  $\mathbf{C}$ . Then we have  $g'(\mathbf{x}) = g(\mathbf{A}^{-1}\mathbf{x})/|\det \mathbf{A}|$  as required.

Thus, we have sketched an algorithm for solving the affine transformation between two point patterns. It should be noticed that in the noiseless case,  $\mathbf{x}'_i = \mathbf{A}\mathbf{x}_i + \mathbf{t}$ , the affine transformation is recovered exactly, up to a numeric round off error, but the above probabilistic framework provides a justification of the method also when there is noise in the point coordinates or some points in the other set have no counterparts in the other.

## 4 Implementation

### 4.1 Intensity-Based Method

In order to apply the proposed registration method for digital images we have to discretize the integral (3). This results in

$$\mathbf{J}f(\alpha, \beta) = \frac{1}{\sum_k f(\mathbf{x}_k)} \sum_k \mathbf{x}_k \tilde{f}(\mathbf{x}_k) \tilde{f}(\alpha \mathbf{x}_k) \tilde{f}(\beta \mathbf{x}_k), \quad (9)$$

where we need samples of  $\tilde{f}$  on three grids centered at the origin and having sample intervals  $\{1, \alpha, \beta\}$ . The sample values are interpolated from the original image  $f$  since we recall that  $\tilde{f}$  is a shifted version of  $f$ . In the experiments we used bilinear interpolation.

The descriptor values (9) are computed for some set of pairs  $(\alpha_i, \beta_i)$ . It should be noted that with a suitable choice of these pairs one may select a single interpolation grid so that it will give samples needed for all used  $(\alpha_i, \beta_i)$ . Hence, only one interpolation is needed to compute several descriptor values. The symmetries in Proposition 4 indicate that we would only need to take  $(\alpha, \beta)$ s from the triangle  $\{(-1, -1), (-1, 1), (1, 1)\}$ , and we chose to take uniform sampling with sample interval 0.25 resulting in 45 different pairs  $(\alpha_i, \beta_i)$  inside the triangle. In this case, the interpolation grid should also have the sample interval 0.25.

From (9) one can directly see that the computational complexity of the registration method is  $O(N^2)$  for an  $N \times N$  image. Finding samples of  $\tilde{f}$  requires interpolation to  $N \times N$  points and the summation in (9) has  $N^2$  terms which results in overall complexity  $O(N^2)$ .

### 4.2 Point-Based Method

Implementation of the point-based registration method of Section 3.2 is quite straightforward. We choose a set of values  $\gamma_i$  and compute the correspondences

$\mathbf{H}'(\gamma_i) \leftrightarrow \mathbf{H}(\gamma_i)$  from which the affine transformation matrix is solved by least-squares. In the experiments we used values  $\gamma_i = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ . The algorithm is summarized in the following:

- Given two sets of points  $\{\mathbf{x}_j\}$  and  $\{\mathbf{x}'_j\}$  compute the corresponding sample means,  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{x}}'$ , and sample covariances,  $\mathbf{C}$  and  $\mathbf{C}'$ .
- Compute the normalized points  $\tilde{\mathbf{x}}_j = \mathbf{x}_j - \bar{\mathbf{x}}$  and  $\tilde{\mathbf{x}}'_j = \mathbf{x}'_j - \bar{\mathbf{x}}'$ . Set  $g = N(\mathbf{0}, \mathbf{C})$  and  $g' = N(\mathbf{0}, \mathbf{C}')$ .
- Compute the points  $\mathbf{H}(\gamma_i)$  and  $\mathbf{H}'(\gamma_i)$ . The expectation values in (7) are computed as sample means by using the samples  $\tilde{\mathbf{x}}_j$  and  $\tilde{\mathbf{x}}'_j$ .
- Solve the affine transformation matrix  $\mathbf{A}$  from point correspondences  $\mathbf{H}'(\gamma_i) \leftrightarrow \mathbf{H}(\gamma_i)$  by least-squares. If a Euclidean or a similarity transformation is wanted the additional constraints may be forced as described in Section 3.1. Finally, solve  $\mathbf{t} = \bar{\mathbf{x}}' - \mathbf{A}\bar{\mathbf{x}}$ .

It can be seen that the method is computationally quite simple. The computational complexity is only  $O(n)$ , where  $n$  is the number of points in the sets. For example, this is significantly better than the complexity of the cross-weighted moment method [5] which is  $O(n^2)$ .

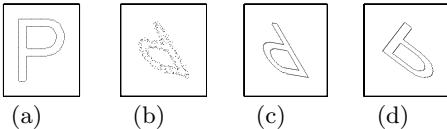
## 5 Experiments

### 5.1 Point Patterns

The first experiment was performed with the point pattern shown in Fig. 1(a). The points were transformed with random affine transformations and isotropic Gaussian noise was added to the coordinates before matching, cf. Fig. 1(b). The random transformation matrices were chosen according to

$$\mathbf{A} = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \kappa \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix},$$

where  $\omega, \phi \in [0, 2\pi]$  and  $\kappa \in [0.3, 1]$  are uniformly distributed random variables. The standard deviation  $\sigma$  of the Gaussian noise was chosen to be proportional to the standard deviation of the  $x$ -coordinates of the original data points, i.e.,  $\sigma = \lambda\sigma_x$ , where values  $\lambda \in [0, 0.1]$  were used.



**Table 1.** Registration results for the point set in Fig. 1. The average values of the matching error  $\epsilon$  at different levels of noise.

**Fig. 1.** Pattern matching: (a) original, (b) transformed and noise added ( $\lambda = 0.06$ ), (c) recovered transformation with the new method ( $\epsilon = 0.12$ ), (d) with MD ( $\epsilon = 1.59$ )

$\lambda$	0	0.02	0.04	0.06	0.08	0.10
MD	0.00	0.01	0.07	0.14	0.24	0.31
CW	0.00	0.05	0.10	0.16	0.23	0.30
H	0.00	0.04	0.09	0.13	0.18	0.23

Patterns were matched with three different methods: affine moment descriptors (MD) [6], cross-weighted moments (CW) [5] and the new method (H). The cross-weighted moment method was implemented according to [5] with the parameter values  $s_1 = 0.95$  and  $s_2 = 0.99$  used by the authors in their experiments.

For each estimated transformation matrix  $\hat{\mathbf{A}}$  we evaluated the distance  $\epsilon$  to the true matrix  $\mathbf{A}$  by defining the points  $\mathbf{p}_1 = (1, 0)^\top$  and  $\mathbf{p}_2 = (0, 1)^\top$  and computing

$$\epsilon = \frac{1}{2} \sum_{i=1}^2 \frac{\|(\mathbf{A} - \hat{\mathbf{A}})\mathbf{p}_i\|}{\|\mathbf{A}\mathbf{p}_i\|}. \quad (10)$$

This is the measure we used to assess the matching result.

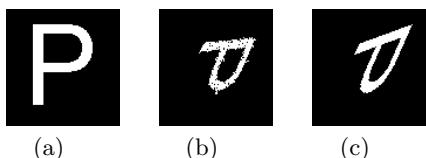
In Table 1 we have tabulated the average values of (10) among 1000 estimated transformations at different levels of noise. The results show that the new method seems to be most tolerant to noise. Although the moment descriptor method often gives a good result it sometimes badly fails as illustrated in Fig. 1(d). The new method and the cross-weighted moment method seem to behave more steadily.

## 5.2 Binary Images

The third experiment was quite similar to the previous one but here we used the binary image shown in Fig. 2(a). The random transformations were obtained as above. The noise added to the transformed images was uniformly distributed binary noise with the noise level  $P$  indicating the probability of a single pixel to change its value. After adding the noise we removed the separated noise pixels from the background, cf. Fig. 2(b).

We did 500 random affine transformations and the average errors  $\epsilon$  at different noise levels are shown in Table 2. Here the point-based variant of the new method (H) was applied so that the white pixels of the images were considered as 2D points. The method (J) is based on (9) and was implemented as described in Section 4.1.

The results show that the moment descriptor method works badly with this data. It works fine for some transformations but now it fails so often that also the average error is quite high. The intensity-based implementation of the new method does not work very well either. Perhaps one reason for this is that the intensity function of a binary image does not contain very much information that



**Fig. 2.** Binary images: (a) original, (b) transformed and noise added ( $P = 0.08$ ), (c) recovered transformation ( $\epsilon = 0.26$ )

**Table 2.** Results for the binary image with different levels of noise.

$P$	0	0.02	0.04	0.06	0.08	0.10
MD	0.47	0.62	0.67	0.74	0.74	0.79
CW	0.11	0.17	0.21	0.28	0.29	0.34
J	0.33	0.37	0.39	0.42	0.43	0.45
H	0.09	0.14	0.18	0.22	0.26	0.29

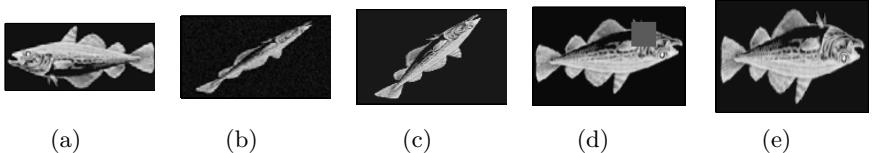
could be utilized in registration. Interestingly, the simple point-based implementation works best. In most cases the estimated transformation is reasonably close to the true one, although the errors are still notable.

### 5.3 Grayscale Images

Next we tested our method by registering the grayscale image in Fig. 3(a) with its affine transformed versions, which were additionally corrupted by Gaussian noise (Fig. 3(b)) or square-shaped occlusion (Fig. 3(b)). The standard deviation of the Gaussian noise was chosen to be proportional to the maximum intensity of the image, i.e.,  $\sigma = \delta I_{\max}/100$ , where values  $\delta \in [0, 6]$  were used. The noise was added also to the black background of the transformed images, which may distort the intensity distribution and hence also the centroid. In the occlusion experiment gray squares with varying size were randomly placed in the image.

We did again 500 random affine transformations and the results are shown in Tables 3 and 4. Here we did not use the cross-weighted moment method since the number of pixels is large and the computation would have been very time consuming. It can be seen from Table 3 that the noise immediately causes the moment descriptor method (MD) to fail. This does not come as a surprise since the second- and third-order statistics are very sensitive to noise in the background [6]. The new method (J) seems to perform much better. Table 4 shows that in the occlusion experiment both methods manage to recover a transformation reasonably close to the true one when the size of the occlusion is not too large.

Finally, we experimented also real images. In Figs. 4(a) and 4(b) we have two views of a book. They were obtained by taking two digital photographs of the book on a dark background so that the book was rotated and translated between the exposures. The original images were decreased in size and segmented so that



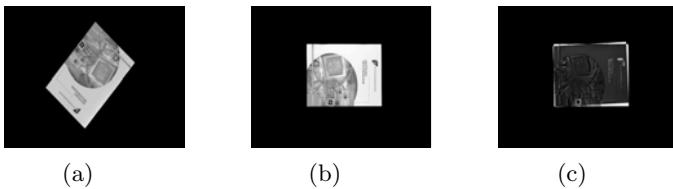
**Fig. 3.** Image registration: (a) original, (b) transformed and noise added (6%), (c) re-covered transformation ( $\epsilon = 0.19$ ), (d) transformed and occluded ( $L = 25$ ), (e) recovered transformation ( $\epsilon = 0.13$ )

**Table 3.** Registration of noisy grayscale images. The average values of the matching error  $\epsilon$  among 500 estimated transformations at six different levels of noise

$\delta$	0	0.5	1	2	4	6
MD	0.03	2.40	2.40	2.40	2.43	2.45
J	0.02	0.04	0.06	0.11	0.20	0.29

**Table 4.** Registration of occluded grayscale images. Parameter  $L$  denotes the side length of the occluding square

$L$	5	9	15	19	25	31
MD	0.04	0.08	0.21	0.32	0.52	0.77
J	0.04	0.11	0.25	0.39	0.65	0.97



**Fig. 4.** Real images: (a) Image 1, (b) Image 2, notice the different illumination, (c) registration result

the background was set to have zero intensity value. The resulting  $400 \times 600$  images are shown in Figs. 4(a) and 4(b). We estimated the similarity transformation between the two images with the new method, implemented as described in Sec. 4.1, and the registration result is illustrated in Fig. 4(c). We can see that an approximately correct transformation is recovered.

## 6 Conclusions

We have proposed a novel method for affine registration of images and point patterns. The method is direct so that it does not require any separate feature extraction step and correspondence search. In addition, the method is non-iterative and efficient since the computational complexity is only linearly proportional to the number of pixels in the images or to the number of points in the patterns.

The particular way of utilizing the entire intensity information also implies that the method has some limitations. Ideally, both images should have similar content, and hence, images that have only partial overlap should be first segmented. However, the performed experiments showed that the method is relatively robust and gives an approximately correct transformation also when this assumption is slightly violated. Therefore, in the first place, the method might be used as an efficient way of computing an initial registration which is then refined by using other more complex registration methods, such as [2] or [3], for example.

## References

1. Zitová, B., Flusser, J.: Image registration methods: a survey. *Image Vis. Comput.* **21** (2003) 977–1000
2. Fitzgibbon, A.W.: Robust registration of 2D and 3D point sets. In: Proc. British Machine Vision Conference. (2001)
3. Viola, P., Wells, W.M.: Alignment by maximization of mutual information. *International Journal of Computer Vision* (1997) 137–154
4. Rangarajan, A., Chui, H., Duncan, J.S.: Rigid point feature registration using mutual information. *Medical Image Analysis* **3** (1999) 425–440
5. Yang, Z., Cohen, F.: Cross-weighted moments and affine invariants for image registration and matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **21** (1999) 804–814

6. Heikkilä, J.: Pattern matching with affine moment descriptors. *Pattern Recognition* **37** (2004) 1825–1834
7. Rahtu, E., Salo, M., Heikkilä, J.: A new efficient method for producing global affine invariants. Submitted (2005)
8. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **9** (1987) 698–700
9. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **13** (1991) 376–380

# Joint Spatial-Temporal Color Demosaicking

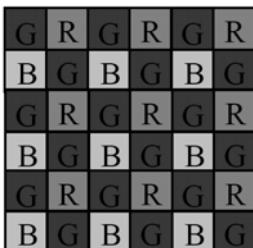
Xiaolin Wu\* and Lei Zhang

Department of Electrical and Computer Engineering,  
McMaster University,  
Hamilton, Ontario, Canada L8S 4K1  
`{xwu, johnray}@mail.ece.mcmaster.ca`

**Abstract.** Demosaicking of the color CCD data is a key to the image quality of digital still and video cameras. Limited by the Nyquist frequency of the color filter array (CFA), color artifacts often accompany high frequency contents in the reconstructed images. This paper presents a general approach of joint spatial-temporal color demosaicking that exploits all three forms of sample correlations: spatial, spectral, and temporal. By motion estimation and statistical data fusion between multiple estimates obtained from adjacent mosaic frames, the new approach can significantly outperform the existing spatial color demosaicking techniques both in objective measure and subjective visual quality.

## 1 Introduction

Digital photography has become a landmark of our information technology era. Unlike traditional film, the image sensors of most digital cameras capture a color image by sub-sampling color bands in a particular mosaic pattern, such as the Bayer color filter array (CFA) [3] shown in Fig. 1. At each pixel, only one of the three primary colors (red, green and blue) is sampled. The full color image is reconstructed by estimating the missing color samples based on spatial and spectral correlations. This process is called color demosaicking, which is critical to the quality of reconstructed color images.



**Fig. 1.** The Bayer pattern

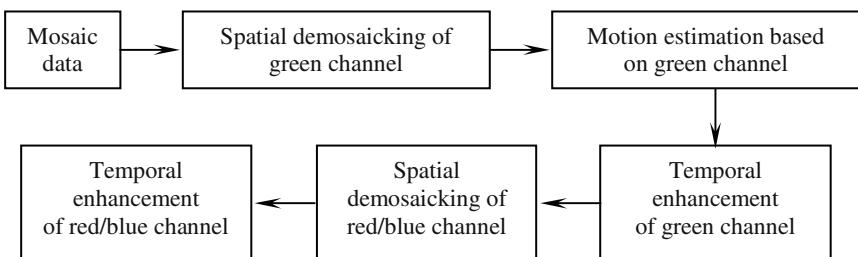
---

\* This research is supported in part by the Natural Sciences and Engineering Research Council of Canada through an industrial research chair in digital cinema.

In the past decades color demosaicking has been extensively studied, but mostly in the spatial domain for still digital cameras. The earlier spatial demosaicking methods are nearest neighbor replication, bilinear and bicubic interpolations [15]. They are easy to implement but highly susceptible to color artifacts such as blocking, blurring and zipper effect at edges. Later demosaicking methods exploited the spectral correlation between color channels. The smooth hue transition (SHT) methods [6, 1] assume images having slowly varying hue. But SHT methods tend to cause large interpolation errors in the red and blue channels when green values abruptly change. Since human visual systems are sensitive to the edge structures in an image, many adaptive demosaicking methods try to avoid interpolating across edges. In the well-known second order Laplacian filter proposed by Hamilton and Adams [9, 2], the second order color gradients are used as the correction terms to interpolate the color channels. In the gradient-based scheme of Chang *et al* [5], gradients in different directions are computed and a subset of them is selected by adaptive thresholding. The missing samples are estimated from the samples along the selected gradients. Recently, Zhang and Wu [28] proposed a linear minimum mean square-error estimation (LMMSE) based demosaicking method and achieved very good results. They reconstructed the primary difference signals (PDS) between the green channel and the red or blue channel instead of directly interpolating the missing color samples. In [17], Lukac and Plataniotis used a normalized color-ratio model in the color interpolation to suppress the color artifacts. They also proposed an edge-sensing method by using color correlation-correction based on a difference plane model [18]. Some color demosaicking techniques are iterative schemes. Kimmel's two-step iterative demosaicking process consists of a reconstruction step and an enhancement step [13]. Another iterative demosaicking scheme was proposed by Gunturk *et al* [8]. They reconstructed the color images by projecting the initial estimates onto so-called constraint sets. A wavelet-based iterative process was employed to update the high frequency details of color channels. More recently reported demosaicking methods of increased sophistication include the method of adaptive homogeneity by Hirakawa and Parks [10], the primary-consistent soft-decision method of Wu and Zhang [27], the principal vector method by Kakarala and Baharav [11], the bilinear interpolation of color difference by Pei and Tam [19]. The above methods can be classified into the class of spatial color demosaicking techniques.

Nevertheless, around the edges where sharp changes in both chrominance and luminance happen, the spatial demosaicking techniques, including those recently-developed sophisticated ones, are error prone due to a lack of correlation in both spectral and spatial domains. In order to overcome the limitation of spatial color demosaicking, additional knowledge and constraints of the original color signals are needed. For digital CCD video cameras, the temporal dimension of a sequence of color mosaic images often reveals more and new information on the color values that are not sampled by the CFA sensors. This potentially valuable information about the color composition of the scene would be unavailable in the spatial domain of individual mosaic frames. The correlation of adjacent frames can be exploited to aid the color demosaicking process if the camera and object motions can be estimated. We call this approach the temporal color demosaicking.

However, there seems to be a lack of research reported on temporal color demosaicking, despite its obvious potential. In this paper we present an effective temporal color demosaicking technique to enhance the color video quality. Without the loss of generality, we consider the Bayer CFA [3] that is widely used in digital color video cameras (see Fig. 1). The temporal demosaicking techniques to be developed by this paper can be readily generalized to other CFA patterns. In the Bayer pattern the sampling frequency of green channel is twice that of red or blue channel. This is because the sensitivity of human visual system peaks at the green wavelength and the green channel contributes the most to the luminance of an image [4]. For natural images, there exists high spectral correlation between the red/blue and green channels. Once the green channel is interpolated with the help of red/blue channel, it can then be used to guide the interpolation of red/blue channel. The main idea of the proposed temporal demosaicking scheme is to match the CFA green sample blocks in adjacent frames in such a way that missing color samples in one frame can be inferred from available color samples of matched adjacent frames. Since the green channel has higher spatial resolution than the red/blue channel, it is naturally employed in the motion estimation process of temporal demosaicking.



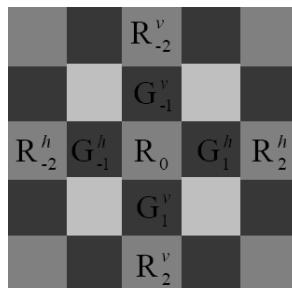
**Fig. 2.** Flow chart of the proposed temporal demosaicking scheme

Fig. 2 is a schematic description of the proposed spatial-temporal demosaicking framework. First, the green channels of all frames are demosaicked individually by intra-frame demosaicking. The motion estimation between adjacent frames for temporal color demosaicking is based on the reconstructed green channel sequence in order to feed the motion analysis with sufficient information. With the estimated motion vectors, adjacent frames are registered spatially. The reference green samples in adjacent frames are then fused with the intra-frame estimates of the missing green samples of the current frame to improve the quality of the previously estimated green channel. The resulting improved green channel will serve as an anchor to reconstruct the red and blue channels by interpolating the missing red and blue samples using both the intra-frame and inter-frame information. The red and blue channels are first recovered spatially with the help of temporally demosaicked green channel, and then guided by the motion vectors, the reference red/blue samples are fused with the spatially estimated samples.

This paper is structured as follows. Section II introduces a new gradient-based spatial demosaicking method for the green channel by optimally weighting the horizontal and vertical interpolation results. These resulting spatially demosaicked green frames are used to compute the relative motions of several adjacent frames in subpixel precision, which is the subject of Section III. After the frame registration, in section IV, the reference frames are fused optimally with the current frame to obtain more robust estimates of the missing color samples. Section V presents the experimental results and Section VI concludes.

## 2 Spatial Demosaicking of Green Channel

Most spatial demosaicking methods exploit the correlation between red, blue and green channels [2, 5, 8-11, 13, 16-20, 26-28]. Since human visual systems are sensitive to the edge structures in an image, it is important not to interpolate across edges. At each pixel the gradient is estimated, and the color interpolation is carried out directionally based on the estimated gradient. Directional filtering is the most popular approach to spatial demosaicking. A well-known directional interpolation scheme is the second order Laplacian correction proposed by Hamilton and Adams [9]. They used the second order gradients of blue and red samples and the first order gradient of green samples to interpolate the green channel. The red and blue samples are interpolated similarly with the correction of the second order gradients of the green samples. In this section, we propose a new spatial demosaicking method. The goal is to provide a good base for the next step of temporal demosaicking at a reasonable computational cost.



**Fig. 3.** A row and a column of mosaic data that intersect at a red sampling position

For ease of presentation and without loss of generality, we examine the case depicted by Fig. 3: a column and a row of alternating green and red samples intersect at a red sampling position where the missing green value needs to be estimated. The symmetric case of estimating the missing green values at the blue sampling positions of the Bayer pattern can be handled in the same way. Denote the red sample at the center of the window by  $R_0$ . Its interlaced red and green neighbors in horizontal

direction are labeled as  $R_i^h$ ,  $i \in \{-2, 2\}$ , and  $G_i^h$ ,  $i \in \{-1, 1\}$  respectively; similarly, the red and green neighbors of  $R_0$  in vertical direction are  $R_j^v$ ,  $j \in \{-2, 2\}$ , and  $G_j^v$ ,  $j \in \{-1, 1\}$ .

Most intra-frame demosaicking methods are based on an assumption that the difference between the green channel and the red/blue channel is a low-pass signal. Let  $\Delta_0 = G_0 - R_0$  be the unknown difference between green and red channels at the sample position of  $R_0$ . The idea is to obtain an estimate of  $\Delta_0$ , denoted by  $\hat{\Delta}_0$ , and then recover the missing green sample by

$$G_0 \approx R_0 + \hat{\Delta}_0 \quad (2-1)$$

The reason for estimating the color difference signal  $\Delta = G - R$  rather than the green signal  $G$  directly is that  $\Delta$  is much smoother than  $G$ . Referring to Fig. 3, the horizontal and vertical differences between the green and red channels at  $R_0$  can be estimated as

$$\Delta_0^h = \frac{1}{2}(G_{-1}^h + G_1^h) - \frac{1}{4}(2 \cdot R_0 + R_{-2}^h + R_2^h) \quad (2-2)$$

$$\Delta_0^v = \frac{1}{2}(G_{-1}^v + G_1^v) - \frac{1}{4}(2 \cdot R_0 + R_{-2}^v + R_2^v) \quad (2-3)$$

One can select between the two estimates  $\Delta_0^h$  and  $\Delta_0^v$ , depending on the gradient. But the binary decision may lose the information in the discarded estimate. Instead, we fuse the two estimates to obtain a more robust estimate of  $\Delta_0$ :

$$\hat{\Delta}_0 = w_h \Delta_0^h + w_v \Delta_0^v \quad (2-4)$$

where  $w_h + w_v = 1$ . Consider  $\Delta_0^h$  and  $\Delta_0^v$  as two independent measurements of the true color difference signal  $\Delta_0$ :

$$\Delta_0^h = \Delta_0 + v_0^h \quad \text{and} \quad \Delta_0^v = \Delta_0 + v_0^v \quad (2-5)$$

where  $v_0^h$  and  $v_0^v$  are the estimation errors of  $\Delta_0^h$  and  $\Delta_0^v$ . Denote by  $m_h$  and  $m_v$  the means of  $v_0^h$  and  $v_0^v$  and by  $c$  the correlation coefficient between  $v_0^h$  and  $v_0^v$ . We empirically observed that  $v_0^h$  and  $v_0^v$  are zero mean and nearly uncorrelated. These properties allow us to derive the optimal weights in the minimum mean square error sense:

$$w_h = \frac{\sigma_v^2}{\sigma_h^2 + \sigma_v^2}, \quad w_v = \frac{\sigma_h^2}{\sigma_h^2 + \sigma_v^2} \quad (2-6)$$

where  $\sigma_h^2 = \text{Var}(v_0^h)$  and  $\sigma_v^2 = \text{Var}(v_0^v)$ .

There are two main influence factors on the estimation errors of  $\Delta_0^h$  and  $\Delta_0^v$ . The first one is the amplitude of  $\Delta_0$ . Most natural scenes consist of predominantly pastoral (unsaturated) colors such that the color difference signal  $\Delta = G - R$  (or  $\Delta = G - B$ ) is not only smooth but also small in amplitude. The large amplitude of  $\Delta_0$  is typically associated with the discontinuity of the color difference signal at the position of  $R_0$ , increasing the risk of large estimation errors. In other words the amplitudes of  $\Delta_0^h$  and/or  $\Delta_0^v$  are proportional to the measurement noises  $v_0^h$  and  $v_0^v$ . The second factor affecting  $v_0^h$  and  $v_0^v$  is evidently the presence of high frequency component of the luminance signal. To account for the second factor we measure the gradient of the red channel at  $R_0$  by

$$d_0^h = R_0 - \frac{R_{-2}^h + R_2^h}{2} \text{ and } d_0^v = R_0 - \frac{R_{-2}^v + R_2^v}{2} \quad (2-7)$$

Let  $\Lambda_h = |\Delta_0^h| + |d_0^h|$  and  $\Lambda_v = |\Delta_0^v| + |d_0^v|$ . Experimental data show that  $\sigma_h$  ( $\sigma_v$ ) is approximately proportional to  $\Lambda_h$  ( $\Lambda_v$ ), i.e.,  $\sigma_h \approx \lambda_h \cdot \Lambda_h$ ,  $\sigma_v \approx \lambda_v \cdot \Lambda_v$  for some constant  $\lambda$ . Therefore,

$$w_h = \frac{\Lambda_v^2}{\Lambda_h^2 + \Lambda_v^2} \text{ and } w_v = \frac{\Lambda_h^2}{\Lambda_h^2 + \Lambda_v^2} \quad (2-8)$$

Obviously, if  $\Delta_0^h$  and  $d_0^h$  have large magnitude, then  $w_h$  is small, reducing the influence of  $\Delta_0^h$  on  $\hat{\Delta}_0$ ; vice versa. By fusing the two directional estimates  $\Delta_0^h$  and  $\Delta_0^v$  with optimal weights  $w_h$  and  $w_v$ , the final estimate  $\hat{\Delta}_0$  is more robust in reconstructing the missing green sample as  $G_0 = R_0 + \hat{\Delta}_0$ .

### 3 Motion Estimation and Re-Sampling

After spatially demosaicking individual green frames, we can improve the resulting green channel by exploiting the temporal correlation of the video signal. The green samples missed by the CFA subsampling process in one frame may be captured in neighboring frames. To use this information we register the frames by determining the relative motions between the current frame and the reference frames. Accurate motion estimation of the video frames is pivotal to temporal color demosaicking.

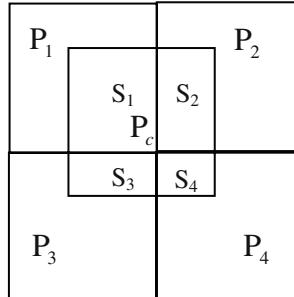
In the Bayer CFA the green channel has twice as many samples as the red and blue channels. Furthermore, the green signal is a good approximation of the luminance signal. For these reasons we estimate the motions in the green channel. This is also why spatial demosaicking of green channel is performed prior to temporal demosaicking. Any of the existing motion estimation techniques can be used to estimate the motion vector in the green channel [7, 14, 21-24]. A more accurate motion estimation method may lead to a better temporal demosaicking result, but of

course at a higher computational cost. It should be stressed, however, that the temporal enhancement technique to be developed in the next section is independent of the motion estimation method. For a good balance between estimation accuracy and low complexity, we choose the block-based motion estimation technique, which is widely used in MPEG 2/4 and other video coding standards [14]. Specifically, we adopt the cross correlation based method proposed in [29] to compute the motion vector in subpixel precision.

As the convention of this paper, we denote the original green samples by  $G$  and the interpolated green samples through the intra-frame demosaicking by  $\hat{G}$ . Let  $\mathbf{M}$  be a block of pixels in the current frame and  $\mathbf{M}_{i,j}$  a matched block in a reference frame with displacement  $(i, j)$ , where  $i$  and  $j$  are integers. Denote by  $(\tau_x, \tau_y)$  the real valued motion vector of  $\mathbf{M}$  from the current frame to the reference frame.

Since  $(\tau_x, \tau_y)$  is real valued in subpixel precision, the corresponding reference block matched to  $\mathbf{M}$ , denoted by  $\mathbf{M}_c$ , should be re-sampled from the reference frame. In the literature the value of a pixel is commonly modeled as the integral of the light over a unit square. Let  $P_c$  be a pixel in  $\mathbf{M}_c$  and suppose the pixel square of  $P_c$  overlaps with those of  $P_1, P_2, P_3$  and  $P_4$ , which are the pixels in the reference frame, as shown in Fig. 4.  $P_c$  is to be reproduced from  $P_1, P_2, P_3$  and  $P_4$ . Denote the areas of the overlaps as  $S_1, S_2, S_3$  and  $S_4$ , which can be computed from the fractional part of the real valued coordinate  $(\tau_x, \tau_y)$ . Then the value of pixel  $P_c$  can be calculated as

$$\text{the sum of the intensities over } S_1, S_2, S_3 \text{ and } S_4: P_c = \sum_{i=1}^4 S_i \cdot P_i .$$



**Fig. 4.** Re-sampling of the reference sample

Due to the structure of the sampling grid of the green channel, two of the four squares  $P_1, P_2, P_3$  and  $P_4$  are the original green samples  $G$  and the other two are the interpolated green samples  $\hat{G}$ . To factor in higher confidence on  $G$  than on  $\hat{G}$ , we put different confidence factors  $c_i$  on  $P_1, P_2, P_3$  and  $P_4$  when computing  $P_c$ :

$$\mathbf{P}_c = \sum_{i=1}^4 c_i \cdot \mathbf{S}_i \cdot \mathbf{P}_i \quad (3-1)$$

where weight  $c_i = 1.2$  if  $\mathbf{P}_i$  is an original green sample and  $c_i = 0.8$  if  $\mathbf{P}_i$  is an interpolated green sample. The sum of weights should be  $\sum_{i=1}^4 c_i = 4$ .

## 4 Temporal Demosaicking

Spatial demosaicking can fail if the discontinuity exists simultaneously in luminance and chrominance. In this case the artifacts cannot be removed by assuming high correlation between the color channels as most spatial demosaicking algorithms do. In contrast, temporal correlation of a mosaic color video signal provides badly needed information to resolve such difficult cases for color demosaicking.

### A. Temporal Update of Green Channel

With the motion estimation and re-sampling algorithms described in Section III, we can get a reference block of the current block  $\mathbf{M}_0$  in each reference frame. Suppose that  $K$  reference frames are used, and denote by  $\{\mathbf{M}_i\}_{i=1,2,\dots,K}$  the re-sampled reference blocks. The spatially demosaicked sample in  $\mathbf{M}_0$  is to be fused with the matched samples in  $\mathbf{M}_i$ . For expression convenience, we denote the spatially interpolated green sample in  $\mathbf{M}_0$  by  $\hat{\mathbf{G}}_0$ , the unknown true green sample corresponding to  $\hat{\mathbf{G}}_0$  by  $\mathbf{G}$ , and the associated reference samples in  $\mathbf{M}_i$  by  $\hat{\mathbf{G}}_i$ . Naturally, we can write  $\hat{\mathbf{G}}_0$  and  $\hat{\mathbf{G}}_i$  as the measurements of true sample  $\mathbf{G}$

$$\hat{\mathbf{G}}_i = \mathbf{G} + \mathbf{e}_i, \quad i = 0, 1, \dots, K \quad (4-1)$$

where  $\mathbf{e}_i$  are the interpolation errors of  $\hat{\mathbf{G}}_i$  in the spatial demosaicking process, and they are nearly uncorrelated with  $\mathbf{G}$ . Since  $\hat{\mathbf{G}}_i$  are independent observations of  $\mathbf{G}$  in different frames, we assume that errors  $\mathbf{e}_i$ ,  $i = 0, 1, \dots, K$ , are mutually nearly uncorrelated. Indeed, if  $\mathbf{e}_i$ 's are significantly correlated, then the observations  $\hat{\mathbf{G}}_i$  are very similar (e.g., if there is no acquisition noise and no motion,  $\hat{\mathbf{G}}_i$  will be identical to each other). In this case the reference frames offer very little new information and temporal demosaicking cannot improve spatial demosaicking anyways.

In order to fuse all the measurements  $\hat{\mathbf{G}}_i$  into a more robust estimate of  $\mathbf{G}$ , we consider the weighted estimate

$$\bar{\mathbf{G}} = \sum_{i=0}^K w_i \hat{\mathbf{G}}_i \quad (4-2)$$

where weights  $\sum_{i=0}^K w_i = 1$ . The criterion of determining  $w_i$  is to minimize the MSE of  $\bar{G}$ , i.e.,  $\{w_i\} = \arg \min_{\sum w_i=1} E[(\bar{G} - G)^2]$ , where  $E$  is the expectation operator.

The weights  $w_i$  may be determined off-line using an appropriate training set. The weights optimized for the training set can then be used in (4-2) to obtain the fused estimate  $\bar{G}$ . However, if the training dataset is not available, or/and if the best color demosaicking performance is desired, on-line adaptive estimation can be made as described below. Let

$$\Omega = E[(\bar{G} - G)^2] = E\left[\left(\sum_{i=1}^K w_i e_i + (1 - \sum_{i=1}^K w_i)e_0\right)^2\right] \quad (4-3)$$

Denote by  $\sigma_i^2$  the variance of error  $e_i$ . Differentiating  $\Omega$  with respect to  $e_i$ ,  $i = 1, 2, \dots, K$ , and setting the partial derivatives to zero. With  $E[e_i e_j]|_{i \neq j} \approx 0$  we have

$$\frac{\partial \Omega}{\partial w_i} = w_i \sigma_i^2 - \left(1 - \sum_{j=1}^K w_j\right) \sigma_0^2 = 0$$

from which we obtain the optimal weight vector  $\mathbf{w} = \text{col}\{w_1, w_2, \dots, w_K\}$  for the estimates in the  $K$  reference frames

$$\mathbf{w} = \mathbf{S}^{-1} \mathbf{1} \quad (4-4)$$

where  $\mathbf{1}$  is a column vector whose elements are all ones and the  $K \times K$  matrix  $\mathbf{S}$  is

$$\mathbf{S} = \begin{bmatrix} 1 + \sigma_1^2 / \sigma_0^2 & 1 & \cdots & 1 \\ 1 & 1 + \sigma_2^2 / \sigma_0^2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 + \sigma_K^2 / \sigma_0^2 \end{bmatrix} \quad (4-5)$$

Solving (4-4) for  $w_i$ , the fused estimate of  $G$  is then computed by (4-2).

## B. Estimation of the Error Variances

To implement the above algorithm, the error variances  $\sigma_i^2$  need to be estimated. From  $\hat{G}_i = G + e_i$  and  $E[e_i e_j]|_{i \neq j} \approx 0$ , we have

$$d_{i,j} = E[(\hat{G}_i - \hat{G}_j)^2] = \sigma_i^2 + \sigma_j^2, \quad i, j = 0, 1, \dots, K \text{ and } i \neq j \quad (4-6)$$

The values of  $d_{i,j}$  can be estimated adaptively from blocks  $\mathbf{M}_i$  and  $\mathbf{M}_j$ ,  $i \neq j$ :

$$d_{i,j} = \frac{1}{L} \sum_{\hat{G}_i \in \mathbf{M}_i, \hat{G}_j \in \mathbf{M}_j} (\hat{G}_i - \hat{G}_j)^2 \quad (4-7)$$

where  $L$  is the total number of missing green samples in blocks  $\mathbf{M}_i$ .

If  $\sigma_0^2$ , i.e., the variance of  $e_0$  in the current block  $\mathbf{M}_0$ , is a known prior, then the values of  $\sigma_i^2$  for other  $i$ 's can be calculated by (4-6) and (4-7). Otherwise, all the values of  $\sigma_i^2$  can be estimated as follows. Let

$$\boldsymbol{\sigma} = \text{col}\{\sigma_0^2, \dots, \sigma_K^2\} \quad (4-8)$$

be a  $K+1$ -dimensional vector that encompass all the  $\sigma_i^2$ , and let

$$\mathbf{d} = \text{col}\{d_{0,1}, \dots, d_{0,K}, d_{1,2}, \dots, d_{1,K}, \dots, d_{K-1,K}\} \quad (4-9)$$

be a  $K(K+1)/2$ -dimensional vector that encompass all the  $d_{i,j}$ , then there exists a  $K(K+1)/2 \times (K+1)$  matrix  $\mathbf{H}$  such that

$$\mathbf{d} = \mathbf{H}\boldsymbol{\sigma} \quad (4-10)$$

Denote  $\mathbf{h}_{i,j}$  as the row in  $\mathbf{H}$  such that  $d_{i,j} = \mathbf{h}_{i,j}\boldsymbol{\sigma}$ . Clearly only the  $i^{\text{th}}$  and  $j^{\text{th}}$  elements in  $\mathbf{h}_{i,j}$  are 1 and all other elements are zeros. We estimate  $\boldsymbol{\sigma}$  by the least square estimation technique:

$$\boldsymbol{\sigma} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{d} \quad (4-11)$$

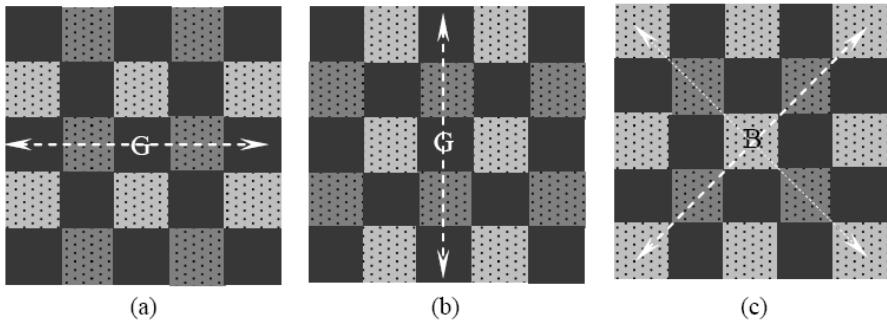
### C. Joint Spatial-Temporal Interpolation of Red/Blue Channels

After the green estimates are improved by the temporal demosaicking process described in Sections IV-A and B, they can in turn guide the demosaicking of the red and blue channels. Similarly to the demosaicking of the green channel, the missing red and blue samples are recovered in two steps. First we spatially interpolate them with the help of the temporally demosaicked green channel, and then temporally improve the interpolation results aided by motion vectors.

The spatial demosaicking of red/blue channel can be accomplished by any of the existing spatial methods. In this paper, we adopt the directional filtering strategy similar to Hamilton and Adams' method [9]. Since the interpolation of blue channel is symmetrical to that of red channel, we only describe the process of interpolating the red channel.

Referring to Fig. 5, there are three cases depending on the positions of missing red samples. Figs. 5 (a) and (b) show the two cases of the missing red samples at the original green pixel positions. Fig. 5 (c) shows the case of a missing red sample at the original blue pixel position. We stress the fact that the missing green samples at the red/blue positions have already been estimated. In the case of Fig. 5 (a), we can estimate the green-red difference signal by using the true red samples and temporally estimated green samples in horizontal direction, and in the case of Fig. 5 (b), we can estimate the green-red difference signal in vertical direction similarly. In the case of Fig. 5 (c), we can estimate two green-red difference values in 45 degree and 135 degree directions. These two values are fused to one result as what we did in

Section II. The missing red sample is estimated by subtracting the estimated green-red difference  $\hat{\Delta}$  from the original green value,  $\hat{R} = G - \hat{\Delta}$ , in cases (a) and (b), or from the estimated green value,  $\hat{R} = \hat{G} - \hat{\Delta}$ , in case (c). Since the above spatial demosaicking process exploits the spectral correlation between red/blue and green, and it operates on temporally demosaicked green channel, the spatial interpolation of red and blue channels indirectly benefits from the temporal redundancy in adjacent frames.



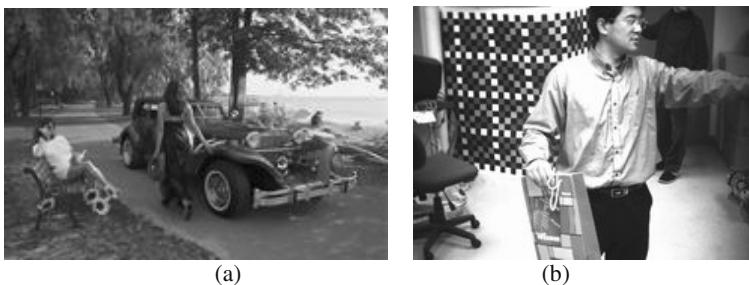
**Fig. 5.** The dark squares represent original green samples, and mid-gray squares original red samples and light-gray squares original blue samples. The dots on red and blue samples mean that the green samples have been estimated temporally. (a) Interpolation of a missing red sample at a green pixel whose horizontal neighbors are red pixels; (b) Interpolation of a missing red sample at a green pixel whose vertical neighbors are red pixels; (c) Interpolation of a missing red sample at a blue pixel

Similarly to the green channel, the spatially interpolated red and blue channels can be further improved via motion estimation and data fusion. However, the motion vectors are still computed in the green channel, because the motion estimation accuracy in the green channel is much higher than in the red and blue channels. Indeed, the sample frequency of red/blue channel is only half of that of the green channel, and the 2D sampling grid of red/blue channel employs inefficient square lattice as opposed to the diamond lattice for the green channel.

The temporal enhancement process of red/blue channel is similar to that of green channel. The main difference is in the confidence factor determination in the re-sampling step. Take the red channel for example, after the motion vector  $(\tau_x, \tau_y)$  between a current block  $\mathbf{M}$  and a reference block  $\mathbf{M}_c$  is computed, a pixel  $P_c$  in  $\mathbf{M}_c$  needs to be re-sampled from the four neighboring pixels  $P_1, P_2, P_3$  and  $P_4$  in the reference frame. In the sampling grid of the red channel, only one of the four pixels is an original red sample  $R$  and the other three are interpolated ones,  $\hat{R}$ . The confidence factors in the re-sampling process  $P_c = \sum_{i=1}^4 c_i \cdot S_i \cdot P_i$  are  $c_i = 1.6$  if  $P_i$  is an original red sample and  $c_i = 0.8$  if  $P_i$  is an interpolated red sample.

## 5 Experimental Results

The proposed joint spatial-temporal color demosaicking algorithm was implemented and tested on simulated color mosaic and real mosaic video data. The first test video sequence was originally captured on film and then digitized by high-resolution scanner. All the three color channels were known and we simulated the mosaic data by subsampling the red, green, blue channels according to the Bayer pattern. The second video sequence was captured directly by a digital video camera. In temporal demosaicking of a current frame, we used two immediately proceeding frames and two immediately succeeding frames as reference frames. Figs. 6 (a) and (b) show the scenes of the two video clips.

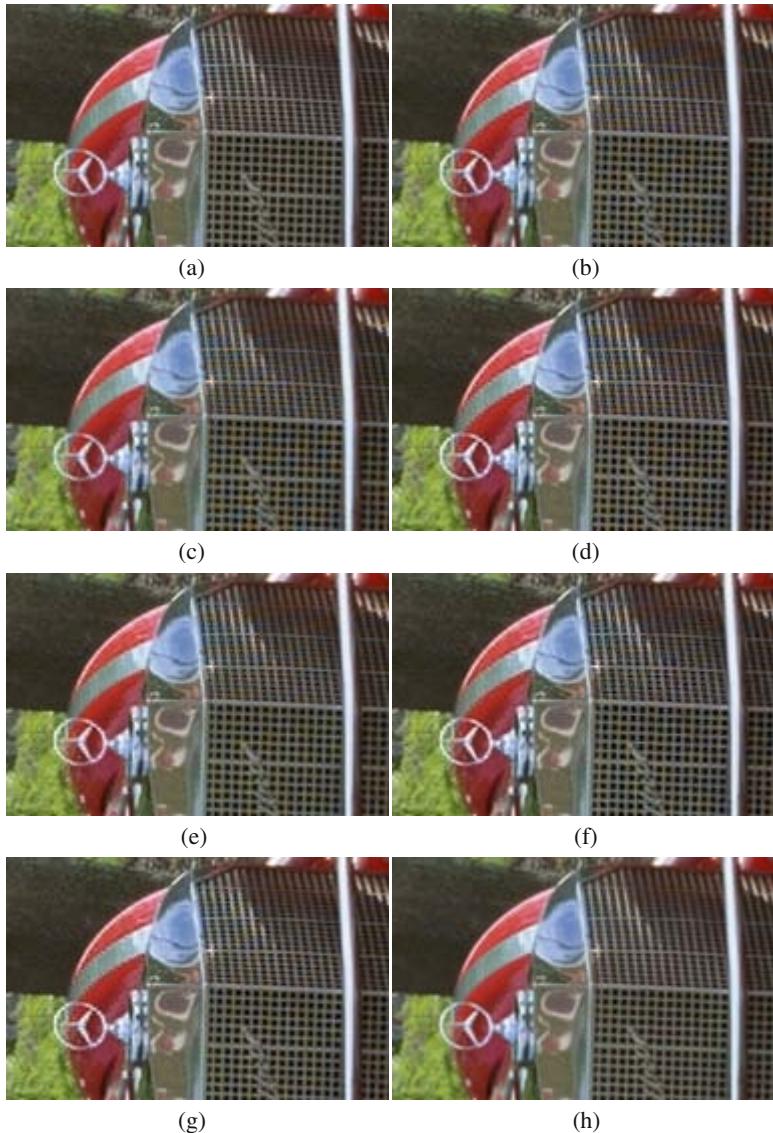


**Fig. 6.** (a) The scene in the first test clip. (b) The scene in the second test clip

Six recently developed spatial demosaicking algorithms were used in our comparison study: the method of second order Laplacian filtering by Hamilton and Adams [9], the gradients-based method by Chang *et al.* [5], the principal vector method by Kakarala and Baharav [11], the normalized color-ratio modeling by Lukac and Plataniotis [17], the method of adaptive homogeneity by Hirakawa and Parks [10], and the directional filtering and fusion method by Zhang and Wu [28].

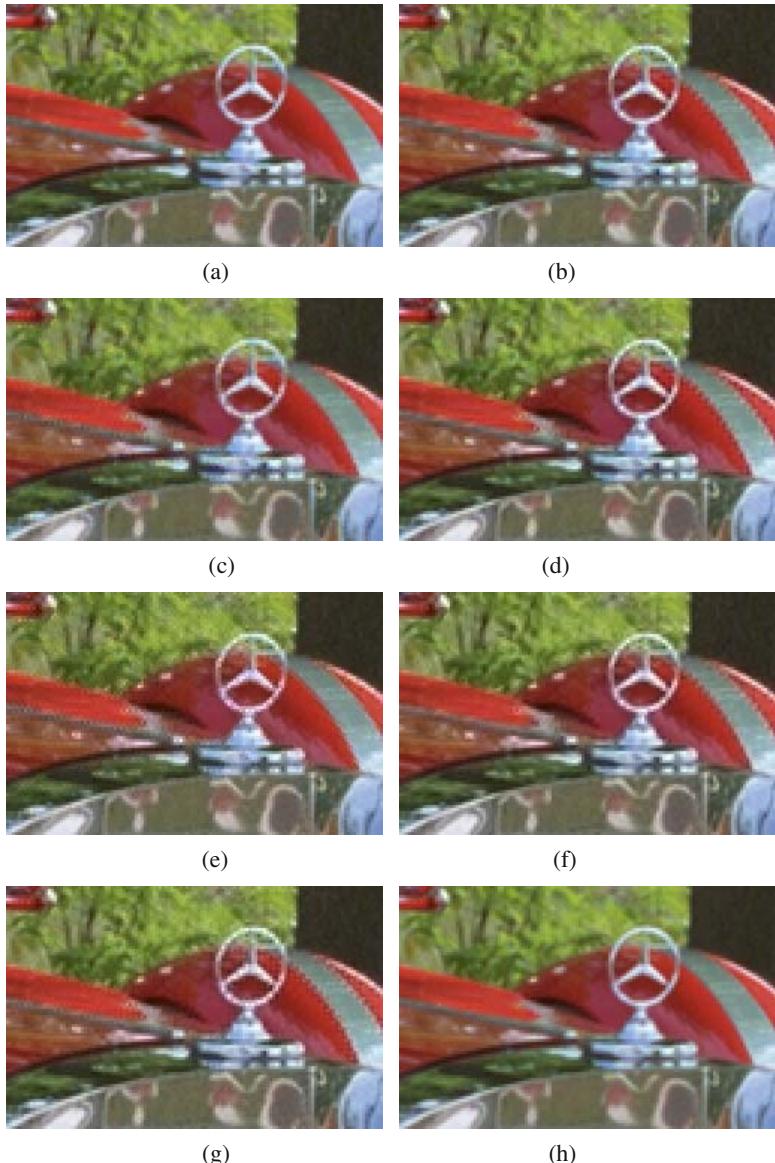
The scene in the first movie clip is a car in a park. The car is still but the camera is rotating around it. The video is captured at a rate of 24-frames/second. The spatial resolution of each frame is  $1948 \times 1280$  and the bit depth is 8 bits per color channel. In this clip, most of the smooth background objects such as the road, the lake, and trees can be reconstructed free of visible artifacts by spatial demosaicking techniques. However, on the car, where some sharp edges accompany abrupt color changes, the spatial demosaicking cannot faithfully recover the missing color components. Fig. 7 (a) shows a  $190 \times 120$  portion of the original frame in question.

Figs. 7 (b) ~ (h) show the demosaicked images by the methods in [9], [5], [11], [17], [10], [28] and the proposed method. Figs. 8 (a) ~ (h) present the close-ups of the demosaicked images by these methods. There are highly visible color artifacts in Figs. 7 (b) ~ (g), particularly on the grill of the car, where the true color signal frequency exceeds the sampling frequency of the Bayer CFA. The recent algorithm in [28] (see Fig. 7 (g)) has fewer color artifacts on the grill than other intra-frame demosaicking methods. But it still generates zipper effects along the boundary of the red and silver



**Fig. 7.** (a) Original image; demosaicked images by the methods in (b) [9]; (c) [5]; (d) [11]; (e) [17]; (f) [10]; (g) [28]; and (h) the proposed temporal scheme

colors and on the emblem of the car (see Fig. 8 (g)), as other spatial demosaicking methods (see Fig. 8 (b) ~ (f)). The color edges on which all spatial demosaicking algorithms fail have discontinuities in both luminance and chrominance. This invalidates the assumption underlying many of these methods that the color difference signal (chrominance) is smooth. For sharp edges of highly saturated colors where



**Fig. 8.** Zoom-in images of the demosaicked results. (a) Original image; demosaicked images by the methods in (b) [9]; (c) [5]; (d) [11]; (e) [17]; (f) [10]; (g) [28]; and (h) the proposed spatiotemporal method

spectral correlation is weak, spatial demosaicking does not have sufficient information to reconstruct the color signal. This is where the temporal correlation can come to the rescue.

**Table 1.** The PSNR results of the 7 demosaicking methods for the first video sequence

Methods		Method in [9]	Method in [5]	Method in [11]	Method in [17]	Method in [10]	Method in [28]	Proposed	
PSNR(dB)		R	30.58	30.85	29.45	29.75	28.14	29.59	<b>33.86</b>
	G	32.33	32.61	32.92	32.87	32.68	35.32	<b>35.54</b>	
	B	26.03	26.25	28.54	28.70	28.66	30.24	<b>30.43</b>	

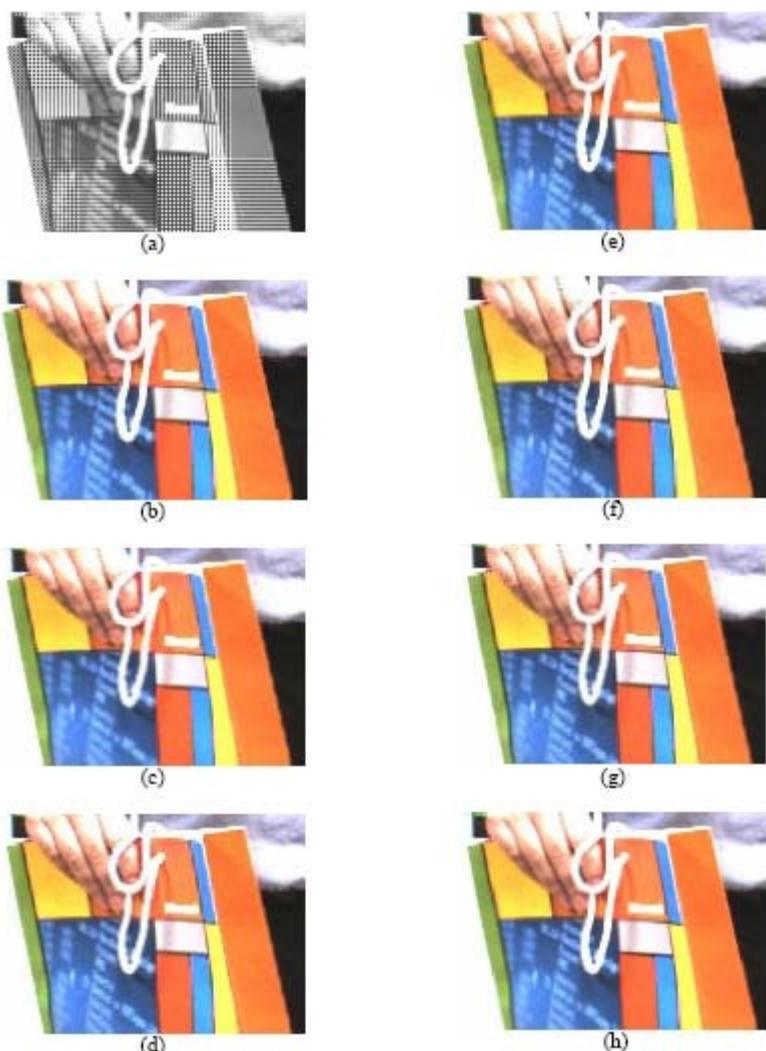
**Fig. 9.** (a) Original mosaic image; demosaicked images by the methods in (b) [9]; (c) [5]; (d) [11]; (e) [17]; (f) [10]; (g) [28]; and (h) the proposed spatiotemporal method

Fig. 7 (h) and Fig. 8 (h) are the demosaicked images by the proposed spatiotemporal demosaicking method. The proposed method has clear advantages over all others in terms of visual quality. Most of the color artifacts are eliminated. It nicely reconstructs many sharp edge structures that are missing or distorted by spatial demosaicking. The PSNR results of the three color channels by these demosaicking methods are listed in Table 1. The proposed method achieves significantly higher PSNR than others as well.

In the second clip, the camera is still but the man is moving with a colorful bag in hand. It was captured directly by a digital video camera at a rate of 25-frames/second. The spatial resolution is 640×480. Fig. 9 (a) shows a 120×150 portion of the mosaic image, where sharp edges of saturated colors exist. Figs. 9 (b) ~ (g) are the results by the spatial demosaicking methods. We can see many artifacts associated with sharp edges. Fig. 9 (h) is the result by the proposed temporal demosaicking method. The proposed method has the best visual quality among the competing methods.

Finally, we want to bring the reader's attention to the significant PSNR increases in the reconstructed red and blue channels by the proposed demosaicking method. This means that besides reducing color artifacts the proposed method also reproduces the color tones more precisely than other methods. Such significant improvements in reproduction precision of the red and blue channels should not come as a surprise, considering that the Bayer CFA has much lower sampling frequency and inferior sampling grid pattern for the red and blue channels. The design bias of the Bayer CFA against the red and blue channels in favor of the green channel makes the faithful reproduction of the red and blue signals more difficult if color demosaicking is carried out in the spatial domain only. This problem can be greatly alleviated by temporal demosaicking.

## 6 Conclusion

We proposed a general spatiotemporal color demosaicking approach that utilizes spatial, spectral, and temporal correlations to recover the missing color samples of raw mosaic data. The green channel is first reconstructed and it acts as an anchor to help recovering the red and blue channels. In reconstructing each one of the three channels, we first interpolate it using a spatial demosaicking method and then temporally enhance it with the help of adjacent frames. The experimental results showed that the proposed approach outperforms the existing spatial color demosaicking techniques by a significant margin, and can remove much of the color artifacts of the latter.

Temporal color demosaicking requires a fairly large buffer to hold multiple reference frames, and involves quite extensive computations compared with the intra-frame demosaicking. We can reduce the complexity by invoking temporal demosaicking judiciously to regions of high frequency contents, where infra-frame demosaicking is unreliable. In smooth regions of an image, which typically constitute the major portion of a frame, the sampling frequency of color mosaic is high enough to allow correct color demosaicking solely in spatial domain.

## Acknowledgement

We thank IMAX Corporation, Mississauga, Canada and Microsoft Research Asia, Beijing, China for providing us the test sequences. We are also indebted to Dr. Lukac, Dr. Kakarala and Dr. Hirakawa for sharing with us their demosaicing programs.

## References

- [1] J. E. Adams, "Intersections between color plane interpolation and other image processing functions in electronic photography," *Proceedings of SPIE*, vol. 2416, pp. 144-151, 1995.
- [2] J. E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," *Proceedings of SPIE*, vol. 3028, pp. 117-125, 1997.
- [3] B. E. Bayer and Eastman Kodak Company, "Color Imaging Array," US patent 3 971 065, 1975.
- [4] R. Bedford and G. Wyszecki, "Wavelength discrimination for point sources," *Journal of the Optical Society of America*, vol. 48, pp. 129-ff, 1958.
- [5] E. Chang, S. Cheung and D. Y. Pan, "Color filter array recovery using a threshold-based variable number of gradients," *Proceedings of SPIE*, vol. 3650, pp. 36-43, 1999.
- [6] D. R. Cok and Eastman Kodak Company, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," US patent 4 642 678, 1987.
- [7] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," *Proc. of IEEE*, vol. 83, pp. 858-876, June 1995.
- [8] B. K. Gunturk, Y. Altunbasak and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Processing*, vol. 11, pp. 997-1013, 2002.
- [9] J. F. Hamilton Jr. and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U. S. Patent, 5 629 734, 1997.
- [10] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm", *IEEE Trans. on Image Processing*, vol. 14, pp. 360-369, Mar. 2005.
- [11] R. Kakarala and Z. Baharav, "Adaptive demosaicing with the principal vector method," *IEEE Trans. Consumer Electronics*, vol. 48, pp. 932-937, Nov. 2002.
- [12] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Pearson Education; 1st edition, Mar. 1993.
- [13] R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," *IEEE Trans. Image Processing*, vol. 8, pp. 1221-1228, 1999.
- [14] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, Boston, 1999.
- [15] P. Longère, Xuemei Zhang, P. B. Delahunt and Davaid H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proc. of IEEE*, vol. 90, pp. 123-132, 2002.
- [16] Wenmiao Lu and Yap-peng Tan, "Color filter array demosaicing: new method and performance measures," *IEEE Trans. Image Processing*, vol. 12, pp. 1194-1210, Oct. 2003.
- [17] R. Lukac and K.N. Plataniotis, "Normalized color-ratio modelling for CFA interpolation," *IEEE Trans. Consumer Electronics*, vol. 50, pp.737- 745, May 2004.

- [18] R. Lukac, K.N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicing approach," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 256-261, February 2004.
- [19] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 503-513, June 2003.
- [20] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging*, vol. 12, No. 4, pp. 633-642, 2003.
- [21] R. R. Schultz, Li Meng and R. L. Stevenson, "Subpixel motion estimation for super-resolution image sequence enhancement," *Journal of Visual Communication and Image Representation*, vol. 9, pp. 38-50, March 1998.
- [22] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Processing*, vol. 5, pp. 996-1011, June 1996.
- [23] C. Stiller and J. Konrad, "Estimating motion in image sequence," *IEEE Signal Processing Magazine*, No. 7, pp. 70-91, July, 1999.
- [24] B. C. Tom and A. K. Katsaggelos, "Resolution enhancement of monochrome and color video using motion compensation," *IEEE Trans. Image Processing*, vol. 10, pp. 278-287, Feb. 2001.
- [25] H. J. Trussell and R. E. Hartwing, "Mathematics for demosaicking," *IEEE Trans. Image Processing*, vol. 11, pp. 485-492, 2002.
- [26] X. Wu, W. K. Choi and Paul Bao, "Color restoration from digital camera data by pattern matching," *Proceedings of SPIE*, vol. 3018, pp. 12-17, 1997.
- [27] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras", *IEEE Trans. Image Processing*, vol. 13, pp. 1263-1274, Sept. 2004.
- [28] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," to appear in *IEEE Trans. Image Processing*.
- [29] L. Zhang and X. Wu, "On cross correlation function based discrete time delay estimation," *ICASSP 2005*, Philadelphia, PA, USA.

# Shape Based Identification of Proteins in Volume Images

Ida-Maria Sintorn and Gunilla Borgefors

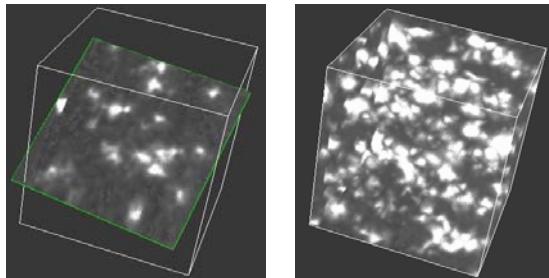
Centre for Image Analysis,  
Swedish University of Agricultural Sciences,  
Lägerhyddsvägen 3, SE-75237 Uppsala, Sweden  
[{ida, gunilla}@cb.uu.se](mailto:{ida, gunilla}@cb.uu.se)

**Abstract.** A template based matching method, adopted to the application of identifying individual proteins of a certain kind in volume images, is presented. Grey-level and gradient magnitude information is combined in the watershed algorithm to extract stable borders. These are used in a subsequent hierarchical matching algorithm. The matching algorithm uses a distance transform to search for local best fits between the edges of a template and edges in the underlying image. It is embedded in a resolution pyramid to decrease the risk of getting stuck in false local minima. This method makes it possible to find proteins attached to other proteins, or proteins appearing as split into parts in the images. It also decreases the amount of human interaction needed for identifying individual proteins of the searched kind. The method is demonstrated on a set of three volume images of the antibody IgG in solution.

## 1 Introduction

The study of the structure of proteins or protein complexes is often the key to understanding how flexible a protein is and how it can interact with, or bind to, other proteins or substances, see, for example [1, 2]. This, in turn, is important for understanding complicated biological systems, e.g., viral infections or the effects of a potential drug candidate. However, imaging of a protein or a protein complex is a difficult task. Atomic resolution (ngstrm scale) can be achieved through X-ray crystallography or nuclear magnetic resonance (NMR). Both have the drawbacks of being very time consuming and restricted to certain types of proteins. Using electron microscopy, almost atomic level is possible to achieve by making a 3D reconstruction based on averaging thousands of proteins of the same kind.

Here, images produced by an individual particle imaging method called Sidec<sup>TM</sup> Electron Tomography (SET), are studied. SET allows imaging of proteins in solution or in tissue at the *nm* scale. This is enough to reveal the main structural features of many proteins, how flexible they are, and how they interact with other proteins [3, 4]. SET is also very fast in comparison with the other methods.



**Fig. 1.** A cross section (left), and a maximum intensity projection (right), of a SET volume image of a solution containing IgG antibody molecules. An IgG molecule is seen as the triangle-shaped object consisting of three parts, almost in the middle of the cross section

Briefly described, the SET images are generated as follows. 2D projections of the flash frozen sample, in our case a solution, are collected at several tilt-angles in the transmission electron microscope. These projections are used to reconstruct the 3D sample by filtered back projection and a refinement method denoted Constrained Maximum Entropy Tomography, COMET [5]. A cross section of a SET volume image of a protein solution containing antibody molecules is shown in Figure 1 (left), and a maximum intensity projection in the direction of the cutting plane (right).

Each SET volume often contain hundreds of objects of which only one or a few are of interest. The proteins have very small volumes, each consisting of only a few thousand voxels, and the shape information is, hence, limited. The proteins of interest are so far found by visual inspection of objects having approximately the correct size after thresholding the intensity. The visual inspection is very time-consuming and the reconstructions also suffer from noise and varying background.

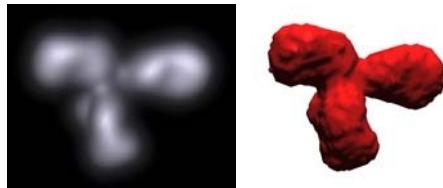
In this paper a segmentation method is presented which reduces the need for human interaction. The method also solves, partly or completely, the problems of varying background, objects touching other objects, and objects appearing as split into parts in the images. These problems can not be solved efficiently by grey-level thresholding in combination with size discrimination. The objects of interest will have different sizes depending on the background, and the size limit therefore needs to be generous not to discard true objects. This in turn increases the amount of objects to be visually judged. Objects of interest touching other objects will have a too large volume to pass the size discrimination step and they will, hence, easily be left unidentified. Objects that have been split into more than one connected component by the threshold will each have a too small volume to pass the size discrimination step and will also easily be left unidentified. The method presented here combines intensity, edge, and shape information using the watershed algorithm [6, 7] and the hierarchical chamfer matching algorithm [8] in a method which handles the problems of varying background, touching or split objects, and need for large amount of human interaction.

## 2 Method

The core of the segmentation method presented here is the chamfer matching algorithm. It was originally presented in [9], and embedded in a resolution pyramid in [8], therefore the name .. . . . . chamfer matching (HCMA). For all template matching algorithms a template is needed, and the image to be searched should represent the same type of information as the template. In the chamfer matching algorithm the template is a binary representation of the shape through, e.g., the contours of a typical object of interest and the images to be searched should therefore contain contour information. The chamfer matching method then finds good fits between the template contour and contours in the image by searching for local minima of a matching function. The matching function is an approximation of the  $L^2$  norm between the template and the contour image. A resolution pyramid with  $l$  levels of the contour image, where the resolution is lowered at each higher level of the pyramid, is calculated. Chamfer matching is then performed at the top (level  $l$ ) of the pyramid, that is in the image with lowest resolution. Good match positions are transferred to the next, lower, level of the pyramid where they serve as starting positions for another round of matching. This continues until matching has been performed in the original image, i.e., the base of the pyramid. When no a priori information is available about the possible local transformations of objects in the image, this multi-resolution approach is necessary for template matching to be a feasible segmentation method.

To extract the contours in the image a seeded watershed (WS) algorithm run on gradient magnitude information, similar to the first steps of the cell nuclei segmentation method in [10], is used. Gradient magnitude information reveals local intensity changes in an image. At places where the local contrast is high, e.g., on the border between a bright object and dark background, the gradient magnitude will be high, while at places with low local contrast, e.g., in the background or in the interior of objects, the gradient magnitude will be low. WS segmentation [6] is a region growing method in which regions grow from all local minima in an image based on grey-level information until they meet other regions. The algorithm is easily understood by interpreting the intensities in an image as a topological map. If a hole is drilled in each local hollow before the map is submerged in water, water will start to fill the map from the hollows. Where two different water basins are about to meet, a dam, or watershed is built to prevent water originating from different hollows to blend. When the entire map has been submerged in water, the resulting regionalization, or the separating watersheds, constitute the segmentation. The algorithm can be applied to different types of information and it has been used in a variety of situations, see e.g., [7, 11, 12], for overviews. Instead of growing regions from every local minimum seeds can be placed in the image as markers of regions of interest, see e.g., [7, 12]. The WS algorithm is then run as described above with the exception that watersheds are only built between regions that each contain a seed.

In the rest of this section we will explain how a template is constructed (or found), how the binary borders are extracted by using a seeded WS algorithm,



**Fig. 2.** A volume rendering of the density reconstruction of an antibody, PDB identification 1igt (left), and a surface rendering of the thresholded volume (right)

and finally how the HCMA algorithm is used to identify objects corresponding to a certain shape in a volume image.

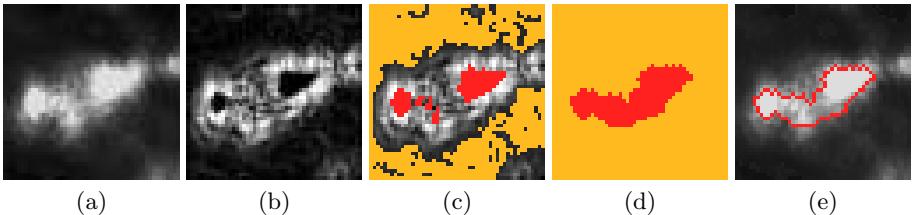
## 2.1 Template Construction

Two types of templates can be discussed for the application in consideration here. Either a typical protein molecule of interest can be found by visual inspection in a SET volume and extracted, or, better, if the atomic structure of the protein is solved and entered into the protein data bank [13], this information can be used to create a template. The latter approach was used in the examples presented here. A volume image where grey-levels depict density in a similar manner as in a SET volume can be constructed by substituting the atomic positions by a gauss kernel weighted by the size of the atom [14], see Figure 2 for such a volume image. Note that the shape of a protein in solution is neither fixed nor always the same, as parts of a protein are more or less flexible. The shape of the template and objects of interest in the image will, hence, not be exactly the same. Once an image of a protein has been constructed, the binary edges to be used as a template need to be extracted. This is done by grey-level thresholding and keeping all voxels of the template that has a face neighbor in the background. The template voxels are described as points in a coordinate system with the origin in the center of the template. This way of storing the template enables simple scaling and rotation of the template in the matching procedure by multiplication with a constant and rotation matrices, respectively.

## 2.2 Binary Contour Extraction

As mentioned in the Introduction, the background varies and a uniform grey-level threshold does not give a satisfactory regionalisation. Instead, a seeded WS algorithm applied to gradient magnitude information is used to achieve a regionalisation stable for smooth background changes. Borders to be used in the subsequent matching are extracted as voxels in the object region having a face neighbor in the background. The different steps to extract the borders are illustrated on part of a 2D slice from a SET volume in Figure 3.

Seeds corresponding to object regions and the background region should be placed in the image. The SET volumes contain bright objects on a dark background. Object seeds should, hence, be placed in bright regions of the image,



**Fig. 3.** The steps to extract the borders illustrated on small part of a 2D slice from one of the SET volumes. The original intensity image (a), the gradient magnitude image (b), and the background and object seeds (c). The watershed regionalization(d), and the extracted borders of the object region overlaid on the intensity image

and the background seed should be placed in the dark region of the image. Object seeds are found by thresholding the intensity image at a high threshold,  $TO$ , known to segment regions definitely inside objects. All voxels above this threshold are given a label stating that they belong to the object region. The background intensity varies to much to use an intensity threshold for background seeding. However, these variations have low frequency and therefore the gradient magnitude image, calculated as described below, is used to find a good background seed. In the gradient magnitude image, all values below a low threshold,  $TB$ , corresponding to flat regions in the original intensity image, are given a label stating that they are possibly part of the background seed. Of these, only the largest connected component is kept as the single background seed. Since most of the SET volumes are background the background seed will cover most of the volume, see Figure 3 (c).

Gradient magnitude information is a measure of local contrast in an image. A gradient magnitude image can be calculated by filtering the image with one, or a set of, local contrast detecting filters, see e.g. [15] for a number of commonly used filters. A set of six 3D Sobel filters are used here, each detecting edges in a different direction. The filter responses in each position of the image are combined to produce the gradient magnitude of that position. In Figure 3 (b), a gradient magnitude image calculated using a set of 2D Sobel filters are shown.

The WS algorithm can be efficiently implemented using sorted pixels lists [16]. In the method described in [16], thick watersheds often result, as image elements located at equal distance from two regions become part of the watershed. In the implementation used here, the water flows around these elements and as a final step these elements are assigned to the region to which most of its neighbors belong. In our implementation, only regions, and no separating watersheds, are the result, as each element of the watersheds is assigned to one of the regions it separates. In a seeded WS algorithm regions start to grow from every local minimum in the image, but as soon as a region containing a seed meets an unseeded region, the unseeded region is conquered by the seeded region. After the WS segmentation, we will, hence, have two regions, object and background, covering the whole image, see Figure 3 (d).

### 2.3 Hierarchical Chamfer Matching

The volumes are rather big and the proteins can have any position and rotation. The HCMA generally needs a starting point close to the optimal position. Very many starting positions are therefore needed in order to guarantee finding the best matches. The matching is performed in a resolution pyramid to speed up the matching and to avoid getting caught in false local minima. The matching is started at low resolution, and positions with reasonably good match scores are kept to the next level. This means that at the best resolution level, that is the original contour image, only rather good positions need to be explored.

From the original contour image, a resolution pyramid is constructed as follows. Each  $2 \times 2 \times 2$  block of voxel at one resolution level is represented by a single voxel at the next higher level. This means that the size of the image is reduced by  $7/8$  at each level. The lowest value in the  $2 \times 2 \times 2$  block is transferred to the corresponding position at the higher level. In the case where the starting image is a binary edge image, this corresponds to an OR-pyramid, since if any of the 8 lower level voxels is a zero, the corresponding voxel at the level in consideration becomes a zero.

A weighted distance transform (DT) is calculated from the contours at all levels of the pyramid. In the resulting distance images all voxels are thus given a value corresponding to the distance to the closest contour voxel. Local distance weights are propagated over the image in two passes [17]. This propagation of local distances is known as chamfering, and weighted DTs are therefore sometimes called chamfer DTs, hence also the name of the matching procedure. The weights, or local distances, used are 3, 4, 5 for the face, edge, and vertex neighbors respectively. These weights are low and produces a DT that is rotation independent enough for matching purposes [18]. During the first pass the image is processed from left to right, top to bottom and front to back. During the second pass the image is processed from right to left, bottom to top and back to front. The voxel under consideration is given the minimum value of itself and the values of its already visited neighbors, each increased by their respective local distance.

The matching is started at the top (level  $l$ ) of the distance transformed pyramid. The template is transformed to the corresponding resolution, placed over the distance image and rotated according to the starting positions, found as described below. The similarity is calculated as the sum of the square of all distance values hit by the template. This means that the lower the sum is, the better is the match. Using the sum of each value hit by the template squared, instead of just the sum, is preferable as it introduces fewer false minima [19]. A steepest descent and a line search method is used to find the closest local optimum for each starting position [20]. Iteratively, a search step is taken in the steepest descent direction if it produces a smaller similarity measure. If not, the step length is halved until it does, or until the step length is so small that the local minimum can be considered reached. For each step, the steepest descent direction is calculated from all seven parameters (three translational, three rotational, and scale). The gradient for each parameter is approximated by centered

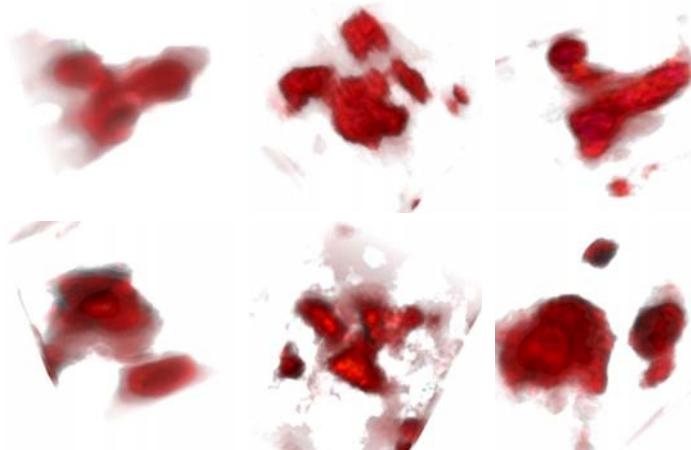
difference, where a translational step is the side length of a voxel, a rotational step is the angle which makes a point on the edge of the template move one voxel side in length, and a scale step is a 2% change in scale.

From one level to the next, only the scores better than a threshold,  $T_1$ , depending on the size of the template and the resolution level, are kept. Once the matching is done at all levels, the positions that are left are presented according to their similarity measures in ascending order, that is, the position with the lowest score is presented first. This means that visual inspection is needed only for judging positions where the image strongly resembles the template. When a few false objects have been presented, the visual inspection can be stopped as the remaining positions are unlikely to correspond to true objects of interest. Since there are many objects in the volumes but rather few true molecules of interest we believe this final decision should not be replaced by a fixed threshold, but should be performed through visual inspection by human experts.

Starting positions, each consisting of a center location and a rotation for the template, could be distributed evenly throughout the highest level ( $l$ ) of the resolution pyramid. This would result in a very large amount of starting positions. In addition, many of these starting positions would be located in the background, hence, requiring more search steps to reach a local minimum. The local minima reached from locations in the background are also more likely to correspond to false hits, as the contours contributing to the local minima can consist of contours from two or more different objects. To reduce the number of starting positions and at the same time increase the probability for a starting position to lead to a true hit, starting locations should only be chosen in the interior of the object region resulting from the WS segmentation. This is achieved by calculating a resolution pyramid of the object region from the WS segmentation, with as many levels as in the contour image pyramid. It is not beneficial to chose starting locations too far into object regions, since many search steps then will be needed to reach a good local minimum. A DT was calculated from the background into the object region, and starting locations were only chosen at distances corresponding to the distance between the center and the closest contour point of the template at that resolution. Once a starting location has been chosen, no more starting locations are picked closer to the location than half the size of the template. For each of these starting locations, rotations were generated with an angular spacing of  $\pi$  in each direction of rotation, leading to a total number of 4 starting positions for each location. Scaling of the template was not considered for generating starting positions, as the approximate scale is known. However, scale is allowed to change during matching.

### 3 Results

The method was tested on three SET volume images of solutions containing antibody IgG, each with one visually identified IgG molecule. The volumes were of size  $150 \times 150 \times 150$  voxels, with a voxel size of  $5.24 \times 5.24 \times 5.24$ , and grey-values in the interval  $[0, 255]$ . A template with a resolution of  $2nm$  and a voxel



**Fig. 4.** Volume renderings of the sub-volumes corresponding to the the best (top row) and second best (bottom row) for each of the three volumes. The visually judged true IgG proteins are at the best position for the first and third volume and at the second best position in the second volume

size of  $5.24 \times 5.24 \times 5.24$  was constructed from PDB (PDB identification 1igt). The constructed image was thresholded at grey-level 50 to extract the template borders. The template borders consisted of 1225 voxels, all fitting inside a box of size  $22 \times 30 \times 27$  voxels. In Figure 2, a volume rendering of the constructed intensity image and a surface rendering of the thresholded volume are shown. The HCMA was run using  $l=3$ , that is, a resolution pyramid with three levels. The threshold  $TO$ , used to identify object seeds in the original intensity image, was 230, and the threshold  $TB$ , used to identify the background seed in the gradient magnitude image, was 25. These thresholds can be easily chosen by studying a small part of the volume. They are not critical, as a range of values will serve the purpose. The rejection threshold  $T1$  was (the number of elements in the template  $\times 10$ ) / (hierarchical level), which only rejected very bad positions.

Volume renderings of sub-volumes of the original image at the two best positions in each image are shown in Figure 4. The three visually identified IgG molecules were in the first and third images found at the positions that best matched the template, and in the second image it was found at the second best position. Note that the structure corresponding to the best position in the second volume shows strong resemblance to an IgG protein and it might very well

a true IgG protein not found by the segmentation algorithm formerly used. The structures at the second best positions in the first and third volume, do not correspond to true IgG proteins. In these two cases the borders from different objects have contributed to their match with the template. Some such cases can, most likely, be identified and removed by studying the mean intensity inside the template placed at the position. We hope to test the method on more volumes in the near future, when more volumes will become available.

## 4 Discussion

As shown in the result section, the proposed method manages to find the objects of interest despite the problems of varying background and objects appearing as connected or split, and despite that the shape of the proteins in solution differ considerably from the crystal structure used as the template. The amount of visual inspection needed to find specific proteins can, hence, be reduced. In situations when the structure of a searched protein is not deposited in PDB, a protein to serve as a template must be found by visual inspection of the reconstructed volumes. Already identified proteins of interest can also be used to re-search volumes and possibly find proteins of interest whose shape differ too much from the crystal shape to be found in a first match procedure.

This paper shows that the proposed method is a promising tool in the analysis of SET images. The strength of the approach is that several types of information and a robust matching measure are combined to produce a stable identification method. In the object seeding process, information that true objects of interest internally have very high intensities is incorporated. Another possible object seeding approach would be to mark local maxima higher than a certain height as object seeds, as in [10]. This would remove the choice of the object threshold, but it would instead lead to many more object regions originating from local maxima that not corresponding to true proteins, as their internal intensity is to low. The problem with varying background is handled by finding the background seed in the gradient magnitude image and using a seeded WS algorithm to extract binary borders. Finally, shape information is used to search for positions in the image where the extracted borders resemble the shape of the template.

The method, or parts of it, can likely be improved by using other, more sophisticated, edge detecting algorithms and optimization methods. However, that the method works despite the simple edge detection and optimization algorithms, demonstrates its robustness. To be able to fine-tune the algorithms and parameters included in the method, its performance on more SET volumes of proteins of several different types need to be investigated.

## References

1. Norel, R., Petrey, D., Wolfson, H.J., Nussinov, R.: Examination of shape complementarity in docking of unbound proteins. *Prot. Struct. Func. Gen.* 36 (1999) 307–317
2. Liang, J., Edelsbrunner, H., Woodward, C.: Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Prot. Sci.* 7 (1998) 1884–1897
3. Banyay, M., Gilstring, F., Hauzenberger, E., Öfverstedt, L.G., Eriksson, A.B., Krupp, J.J., Larsson, O.: Three-dimensional imaging of *in situ* specimens with low-dose electron tomography to analyze protein conformation. *Assay Drug Dev. Technol.* 2 (2004) 561–567
4. Sandin, S., Öfverstedt, L.G., Wikström, A.C., Wrangle, O., Skoglund, U.: Structure and flexibility of individual immunoglobulin G molecules in solution. *Structure* 12 (2004) 409–415

5. Skoglund, U., Öfverstedt, L.G., Burnett, R., Bricogne, G.: Maximum-entropy three-dimensional reconstruction with deconvolution of the contrast transfer function: A test application with adenovirus. *J. Struct. Biol.* 117 (1996) 173–188
6. Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection. In: International Workshop on Image Processing: Real-time and Motion Detection/ Estimation. (1979) 2.1–2.12
7. Meyer, F., Beucher, S.: Morphological segmentation. *J. Vis. Comm. Im. Repr.* 1 (1990) 21–46
8. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pat. Anal. Mach. Intell.* 10 (1988) 849–865
9. Barrow, H.G., Tenenbaum, J.M., Bolles, R., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In: Proc. 5th Int. Joint Conf. Artif. Intell., Cambridge, Massachusetts (1977) 659–663
10. Wählby, C., Sintorn, I.M., Erlandsson, F., Borgefors, G., Bengtsson, E.: Combining intensity, edge, and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *J. Micr.* 215 (2004) 67–76
11. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Trans. Im. Proc.* 2 (1993) 176–201
12. Beucher, S.: The watershed transformation applied to image segmentation. *Scanning Microscopy* 6 (1992) 299–314
13. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalova, I., Bourne, P.: The protein data bank. *Nucl. Ac. Res.* 28 (2000) 235–242
14. Pittet, J.J., Henn, C., Engel, A., Heymann, J.B.: Visualizing 3D data obtained from microscopy on the internet. *J. Struct. Biol.* 125 (1999) 123–132
15. Sonka, M., Hlavac, V., Boyle, R.: 4. In: *Image Processing, Analysis, and Machine Vision*. 2nd edn. Brooks/Cole Publishing Company (1999) 77–88
16. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pat. Anal. Mach. Intell.* 13 (1991) 583–597
17. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. *J. Ass. Comp. Mach.* 13 (1966) 471–494
18. Borgefors, G.: On digital distance transforms in three dimensions. *Comp. Vis. Im. Underst.* 64 (1996) 368–376
19. Borgefors, G.: An improved version of the chamfer matching algorithm. In: 7th Int. Conf. on Pat. Rec., Montreal, Canada (1984) 1175–1177
20. Nash, S.G., Sofer, A.: 10.5, 11.1. In: *Linear and Nonlinear Programming*. McGraw-Hill (1996)

# Thickness Estimation of Discrete Tree-Like Tubular Objects: Application to Vessel Quantification

D. Chillet<sup>1,2,3</sup>, N. Passat<sup>1,2</sup>, M.-A. Jacob-Da Col<sup>1</sup>, and J. Baruthio<sup>2</sup>

<sup>1</sup> LSIIT, UMR 7005 CNRS-ULP, Parc d'Innovation,  
bd Sébastien Brant BP 10413, 67412 Illkirch cedex, France  
[{passat, dacol}@lsiit.u-strasbg.fr](mailto:{passat, dacol}@lsiit.u-strasbg.fr)

<sup>2</sup> IPB, UMR 7004 CNRS-ULP, Faculté de Médecine,  
4, Rue Kirschleger, 67085 Strasbourg cedex, France  
[baruthio@ipb.u-strasbg.fr](mailto:baruthio@ipb.u-strasbg.fr)

<sup>3</sup> École Supérieure Chimie Physique Électronique de Lyon,  
Domaine Scientifique de la Doua, 43, bd du 11 novembre 1918,  
BP 2077, 69616 Villeurbanne cedex, France  
[dini.chillet@cpe.fr](mailto:dini.chillet@cpe.fr)

**Abstract.** Thickness estimation of discrete objects is often a critical step for shape analysis and quantification in medical applications. In this paper, we propose an approach to estimate the thickness (diameter or cross-section area) of discrete tree-like tubular objects in 3D binary images. The estimation is performed by an iterative process involving skeletonization, skeleton simplification, discrete cross-section plane evaluation and finally area estimation. The method is essentially based on discrete geometry concepts (skeleton, discrete planes, and discrete area). It has been validated on phantoms in order to determine its robustness in case of thickness variations along the studied object. The method has also been applied for vessel quantification and computer-aided diagnosis of vascular pathologies in angiographic data, providing promising results.

**Keywords:** thickness estimation, discrete tree-like tubular objects, vascular imaging, vessel quantification, computer-aided diagnosis.

## 1 Introduction

Thickness estimation (diameter or cross-section area) of discrete objects is often an important step for shape analysis and quantification in medical applications. Since vascular diseases are one of the most important public health problems, quantification of discrete tubular structures can be of precious use in computer-aided diagnosis. Indeed, it can help detecting aneurysms and stenoses which are characterized by vessel thickness abnormal variations.

Several methods have been proposed in order to compute tubular object thickness for vessel quantification. Some of them are designed to directly process

grey-level images [6, 7] while others are devoted to segmented binary data [3, 8, 12]. The method presented in this paper is devoted to thickness estimation of discrete tree-like tubular objects contained in binary images.

It is composed of the following steps. First, a skeleton is extracted from the initial object. This skeleton is then simplified by a maximum-path pruning process and smoothed with Bézier curves in order to obtain the object medial axes. Tangent lines are then estimated along these medial axes in order to determine the object discrete normal planes. Thickness is finally deduced from the area of these cross-section planes.

The method, devoted to vascular structures (but applicable for any tree-like discrete object) has been validated on tubular phantoms in order to determine the relative estimation errors in case of thickness variations along the object. Finally, real angiographic data have been processed to evaluate the ability of the method to be used for quantification and computer-aided diagnosis.

This paper is organized as follows. In Section 2, previous approaches for thickness estimation of discrete tubular objects from binary images are described. The proposed algorithm is then fully described in Section 3. In Section 4, experimental results obtained from phantoms and real data are provided. Discussion and future projects are finally presented in Section 5.

## 2 Related Work

The main researches concerning 3D tubular object thickness estimation have focused on vascular image analysis, in order to quantify vessel attributes. Two different approaches have been proposed. The first one consists in directly working on 3D angiographic images and to extract thickness information from high-intensity grey-level structures [6, 7]. It can then take advantage of knowledge concerning the acquisition process. Nevertheless, it can not be used for other application fields. The second approach deals with binary images and can then be used for any segmented grey-level images but also for any application involving virtual or real binary data. The principal methods belonging to this category use the following steps: skeleton extraction, skeleton pruning to obtain the medial axis, and final estimation of a thickness parameter (area or diameter).

In [3], Chen et al. obtain the skeleton by using an iterative thinning algorithm. The skeleton is pruned in order to delete cycles and irrelevant branches. However, they do not focus on quantifying the vessels, only explaining that the diameter can be estimated from the cross-section area.

In a method proposed by Sorantin et al. in [12], the skeleton is obtained with the same thinning algorithm. During the pruning step, the shortest path between two points chosen by the user is determined. Then, the tangent lines along the skeleton are smoothed using the Casteljau's algorithm. The cross-section area is then computed along the object.

An original approach is proposed by Nyström et al. in [8]. The segmented vessels are first reduced to a reversible surface skeleton. A curve skeleton is then extracted from the surface skeleton. Finally, the minimal radius at each point of

the curve skeleton is estimated using the distance map associated to the initial binary image.

Despite their originality, it has to be noticed that these strategies present several drawbacks. In [3], the method provides a whole skeleton of the tubular tree, but thickness estimation is not explicitly provided. In [12], the proposed approach only processes a single vessel among the whole vascular network. Such a method can then be used for analysing a precise vessel but it requires user interaction for choosing extremity points. Finally, the method proposed in [8], being based on distance maps can only determine the minimal diameter of a non circular cross-section, only allowing to detect vessel narrowings (stenoses) but no vessel thickenings (aneurysms) in medical applications.

In the following, we describe an algorithm allowing to fix these problems. The method enables to compute a whole tree-like structure in order to obtain an accurate thickness estimation along all the branches without any user interaction. This method, essentially based on discrete geometry concepts, is described in the following sections.

### 3 Algorithm

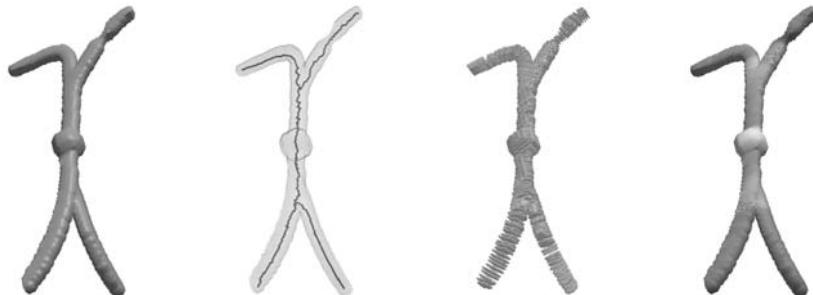
#### 3.1 Input and Output

Any 3D binary image  $I$  of dimensions  $dim_x$ ,  $dim_y$ ,  $dim_z$ , processed by the method can be considered as a subset of the 3D discrete space:  $I = [0, dim_x - 1] \times [0, dim_y - 1] \times [0, dim_z - 1] \subset \mathbb{Z}^3$ . It has to be noticed that the considered images are assumed to be isotropic (since real data acquisition processes generally provide 3D images with non cubic voxels, such images have to be preprocessed). However, no hypothesis has to be done on the voxel size. In the following, we will always consider the (26,6)-adjacency (26-adjacency for the foreground and 6-adjacency for the background).

The method takes as input binary images (here obtained with region-growing methods described in [9]). As output, it provides two kinds of data. The first one is a 3D image identical to the input image but presenting for each voxel a color depending on the object thickness at the current position. The second one is a graph indicating the thickness along the object. These two data then provide both qualitative and quantitative results. The different steps of the proposed method are illustrated on a simple shape example in Fig. 1.

#### 3.2 Skeleton

Skeletonization is a process modifying a given object into a thinner object, centered in the initial one. The skeleton is composed of much less points but is assumed to preserve the initial object shape and topology. The proposed method uses an iterative thinning algorithm to perform the skeletonization. At each iteration, only simple voxels (voxels whose deletion does not modify the object topology) are considered and deleted. This first step then provides a skeleton representing the initial object.



**Fig. 1.** Successive steps of the method applied on a virtual object. From left to right: initial object; skeleton computation; cross section planes computed by intersecting the object and the digital naive normal planes; thickness image obtained from area estimation (the grey-levels depend on the estimated area of the cross-section planes)

Since skeletonization is a noise-sensitive process, it might happen that the obtained skeleton contains irrelevant branches and non unit-width parts. The algorithm used to remove such branches and useless points is inspired from the method used in [3]. This method is based on an iterative longest path detection algorithm which requires a seed point. In [3], it is proposed to choose the point of highest intensity in the initial image. Nevertheless, since the processed data are binary images, we propose to choose a voxel belonging to the object and located on its bounding box, avoiding an interactive choice. Then, at each iteration, the longest path among all non visited points is searched, using Dijkstra's algorithm, and added to the pruned skeleton. The process is iterated until the longest path becomes shorter than a predetermined threshold.

### 3.3 Medial Axes

Once the skeleton is pruned, the tangent lines have to be computed for each point. The skeleton generally presents an irregular shape. Then, a direct estimation of the tangent lines from it can lead to incorrect results. In order to fix this problem each branch of the skeleton is smoothed by estimating a Bézier curve from it. The control points used for each branch are chosen by sampling the skeleton. The use of Bézier curves enables to significantly smooth the skeleton, finally providing medial axes being correctly oriented and centered in the object. It is then possible to efficiently determine the tangent line in each position of these axes by only considering a set of successive points.

### 3.4 Normal Planes

At this stage of the method, the tangent lines are determined for each point of the medial axes. The tangent at a point defines the normal vector  $\mathbf{n} = (n_x, n_y, n_z)$  of the cross-section plane at the current position. We propose to use the digital naive plane definition introduced by Reveillès in [11] to construct the cross-

section plane for each point. Using digital naive planes presents several advantages. Indeed, fast and efficient algorithms can be used to compute the number of voxels composing them. Moreover, they can be projected on orthogonal planes without loss of information. The cross-section planes are then obtained by intersecting the object and the computed naive planes.

### 3.5 Cross-Section Area

Every cross-section plane can be directly projected onto the three principal planes  $O_{yz}$ ,  $O_{xz}$ , and  $O_{xy}$ . The corresponding areas  $A_x$ ,  $A_y$ , and  $A_z$  are then estimated using a method proposed in [5]. For each projection, the area is estimated as a weighted sum of the pixels (projections of voxels), the weights depending on the pixel neighborhood configurations. Finally, the area of the cross-section plane  $A$  is computed as a combination of the three projected areas  $A_x$ ,  $A_y$ ,  $A_z$ , and the coordinates of the normal vector of the cross section plane  $\mathbf{n} = (n_x, n_y, n_z)$ :

$$A = A_x \cdot n_x + A_y \cdot n_y + A_z \cdot n_z,$$

using a formula proposed by Coeurjolly in [4]. For cross-section planes assumed to present a circular or nearly circular shape, the diameter can be directly obtained from the area value.

## 4 Experiments and Results

The proposed method has been implemented on the Medimax<sup>1</sup> software platform and use the ImLib3D<sup>2</sup> open source C++ library. The computer used to run the method was composed of a 3 GHz Pentium IV processor with 2 GB of memory. The algorithm then requires from 10 seconds for simple objects (Fig. 1) to 6 minutes for complex real objects (left picture of Fig. 2).

### 4.1 Error Estimation on Phantoms

The phantoms used for thickness error estimation are cylinders of length varying from 6 to 8 cm, simulating stenoses (narrowings) and aneurysms (thickenings) or having homogeneous diameters. They were created with silicon elastomer according to the method described by Stevanov et al. in [13]. The cylinder images have been acquired on a 1 Tesla scanner. The estimated thicknesses have then been compared to the thicknesses directly measured on the phantoms.

The results of error estimation are illustrated in Table 1. Since the first three cylinders (tubes 1, 2, and 3) do not present diameter variations, the average estimated diameter is directly compared with the real diameter. The highest relative error is 5.6% which can be considered as a satisfactory result. The other cylinders (tubes 4, 5, and 6) present thickness variations modeling aneurysms (tube 6) and stenoses (tubes 4 and 5). Both smallest and greatest diameters

---

<sup>1</sup> Available at <http://www-ipb.u-strasbg.fr>.

<sup>2</sup> Available at <http://imlib3d.sourceforge.net>.



**Fig. 2.** Vessel quantification of angiographic data. From left to right: cerebral veins and arteries, arch of aorta, iliac bifurcation

**Table 1.** Phantom thickness estimation errors: real and estimated diameters of phantoms

	estimated diameter (voxels)	real diameter (mm)	error (%)
tube 1	34.12	30.63	5.6
tube 2	10.64	9.56	2.5
tube 3	5.49	4.93	1.4
tube 4	6.13/ 9.81	5.51/ 8.81	5.70/ 9.80 0.6/10.1
tube 5	16.54/23.00	14.85/20.65	15.00/ 20.50 1.0/ 0.7
tube 6	16.37/22.84	14.65/20.51	14.50/ 20.20 1.3/ 1.5

**Table 2.** Phantom thickness estimation errors: real and estimated degrees of severity of simulated stenoses and aneurysms

	degree of severity		error
	estimated	real	(%)
tube 4	0.37	0.42	11.9
tube 5	0.28	0.27	3.7
tube 6	0.28	0.28	0.0

are compared in Table 1, while the real and estimated degrees of severity (an important criterion for diagnosis) are detailed in Table 2. This degree is defined by  $1 - \frac{\min}{\max}$  where  $\min$  (resp.  $\max$ ) is the smallest (resp. the largest) diameter. One can observe that most results are quite accurate (errors are generally lower

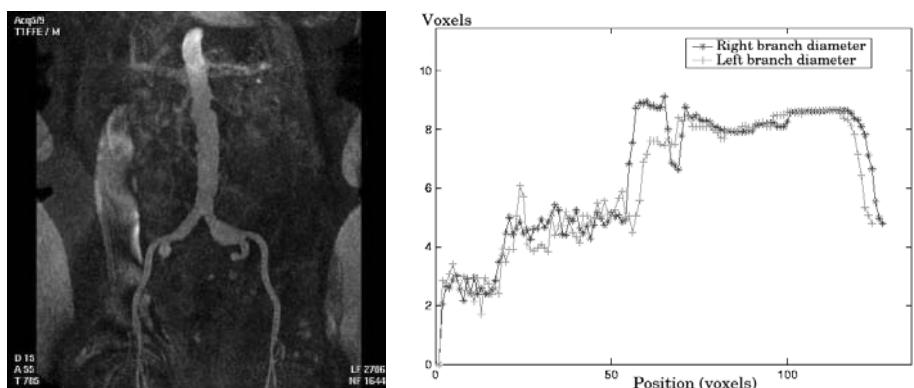
than 2%). Only one result is uncorrect, for tube 4 (error of approximately 10%). However, this error has been caused by a skeletonization error at the extremity of the cylinder which can be explained by the low length/diameter ratio of the phantom. In real data, the tubular structures being much longer, this kind of skeletonization errors do not occur.

## 4.2 Vessel Quantification and Computer-Aided Diagnosis

The method has been tested on real angiographic data (cerebral and carotid MRA, thoracic and abdominal CT scan) for vessel quantification or computer-aided diagnosis of vascular pathologies (aneurysms and stenoses). Examples of vessel quantification are illustrated for several vascular structures in Fig. 2.

This figure emphasizes the visualization improvement provided by thickness information, which makes the 3D data more easily readable than binary data. It has to be noticed that information on vessel size and orientation (provided by the estimated medial axes) is also of precious use for cerebral vascular atlases generation methods as the one proposed in [10]. However, quantification can also be used for computer-aided diagnosis, as described in the following example.

In the left picture of Fig. 3, a maximum intensity projection (MIP) of an iliac bifurcation angiography can be observed. For this data, the method has provided an image where the color (here the grey-level) depends on the thickness (right picture of Fig. 2), but also a graph representing the thickness (right part of Fig. 3). On the 3D colored image, it is possible to visually detect a dark part on the right branch, surrounded by two clear parts: this structure is the sought aneurysm. On the graph, the curves representing the thickness evolution along the two branches are quite similar apart a peak: this difference corresponds with



**Fig. 3.** Iliac bifurcation thickness estimation. Left: MIP showing an aneurysm on the right branch (see right picture of Figure 2 for the corresponding 3D visualization). Right: the right and left branches diameter evolution according to the position

the localisation of the aneurysm. The graph enables to determine not only its length but also its severity degree.

## 5 Discussion and Further Works

This paper has proposed a fully automatic method for thickness estimation of discrete tree-like tubular structures in binary images. This method, devoted to 3D vessel analysis and quantification, can however be used for any kind of objects presenting similar properties. The algorithm, essentially based on discrete geometry concepts presents several advantages by comparison to previous approaches: automation, ability to process a whole tree-like object and accurate estimation of cross-section areas. This accuracy has been validated on phantoms, emphasizing the method robustness even in case of thickness variations along the object. Tests have also been carried out on real angiographic images. They tend to prove that the method can be useful for vessel quantification and computer-aided diagnosis of vascular pathologies such as aneurysms and stenoses. Further work could now consist in modifying and improving the skeleton smoothing step, currently using Bézier curves, in order to obtain a method entirely based on discrete structures.

## Acknowledgement

The authors thank the EPML IRMC<sup>3</sup> (Équipe Projet Multi-Laboratoires Imagerie et Robotique Médicale et Chirurgicale, EPML #9 CNRS-STIC) for its financial support.

## References

1. G. Borgefors, I. Nyström, G. Sanniti di Baja: Computing skeletons in three dimensions. *Pattern Recognition* **32** (1999) 1225–1236
2. S.Y.J. Chen, J.D. Carroll, J.C. Messenger: Quantitative analysis of reconstructed 3D coronary arterial tree and intracoronary devices. *IEEE Transactions on Medical Imaging* **21** (2002) 724–740
3. Z. Chen, S. Mollo: Automatic 3D vascular tree construction in CT angiography. *Computerized Medical Imaging and Graphics* **27** (2003) 469–479
4. D. Coeurjolly: Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces. PhD Thesis, Université Lumière, Lyon 2 (2002). <http://liris.cnrs.fr/david.coeurjolly/These/these.pdf>
5. O. Duda, P.E. Hart: *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc. (1973)
6. A.F. Frangi, W.J. Niessen, P.J. Nederkoorn, J. Bakker, W.P.T.M. Mali, M.A. Viergever. Quantitative analysis of vascular morphology from 3D MR angiograms: in vitro and in vivo results. *Magnetic Resonance in Medicine* **45** (2001) 311–322

---

<sup>3</sup> <http://irmc.u-strasbg.fr>

7. R.M. Hoogeveen, C.J.G. Bakker, W. Mali, M.A. Viergever: Vessel diameter measurement in TOF and PC angiography: a need for standardization. In Annual Meeting of the International Society for Magnetic Resonance in Medicine (1997) 1847
8. I. Nyström, O. Smedby: New presentation method for magnetic resonance angiography images based on skeletonization. In Proc. SPIE Medical Imaging: Image Display and Visualization 2000 **3976** (2000) 515–522
9. N. Passat, C. Ronse, J. Baruthio, J.-P. Armspach, C. Maillet, C. Jahn: Atlas-based method for segmentation of cerebral vascular trees from phase-contrast magnetic resonance angiography. In Proc. SPIE Medical Imaging: Image Processing 2004 **5370** (2004) 420–431
10. N. Passat, C. Ronse, J. Baruthio, J.-P. Armspach, C. Maillet: Cerebral vascular atlas generation for anatomical knowledge modeling and segmentation purpose. To appear in IEEE Computer Vision and Pattern Recognition (2005)
11. J.P. Reveillès: Combinatorial pieces in digital lines and planes. In Proc. Vision Geometry **IV** (1995) 23–34
12. E. Sorantin, C. Halmai, B. Erdohelyi, K. Palagy, L.G. Nyul, K. Olle, B. Geiger, F. Lindbichler, G. Friedrich, K. Kiesler: Spiral-CT-based assessment of tracheal stenoses using 3D skeletonization. IEEE Transactions on Medical Imaging **21** (2002) 263–273
13. M. Stevanov, J. Baruthio, B. Eclancher: Fabrication of elastomer arterial models with specified compliance. Journal of Applied Physiology **88** (2000) 1291–1294

# Segmentation of Multimodal MRI of Hippocampus Using 3D Grey-Level Morphology Combined with Artificial Neural Networks

Roger Hult<sup>1</sup> and Ingrid Agartz<sup>2</sup>

<sup>1</sup> Centre for Image Analysis, Uppsala University,  
Lägerhyddsvägen 3, SE-752 37 Uppsala, Sweden  
[rogerh@cb.uu.se](mailto:rogerh@cb.uu.se)

<sup>2</sup> Dept. Clinical Neuroscience, Human Brain Informatics,  
Karolinska Institutet, SE-171 75, Stockholm, Sweden

**Abstract.** This paper presents an algorithm for improving the segmentation from a semi-automatic artificial neural network (ANN) hippocampus segmentation of co-registered T1-weighted and T2-weighted MRI data, in which the semi-automatic part is the selection of a bounding-box. Due to the morphological complexity of the hippocampus and the difficulty of separating from adjacent structures, reproducible segmentation using MR imaging is complicated.

The grey-level thresholding uses a histogram-based method to find robust thresholds. The T1-weighted data is grey-level eroded and dilated to reduce leaking from hippocampal tissue to the surrounding tissues and selecting possible foreground tissue. The method is a 3D approach, it uses  $3 \times 3 \times 3$  structure element for the grey-level morphology operations and the algorithms are applied in the three directions, sagittal, axial, and coronal, and the results are then combined together.

## 1 Introduction

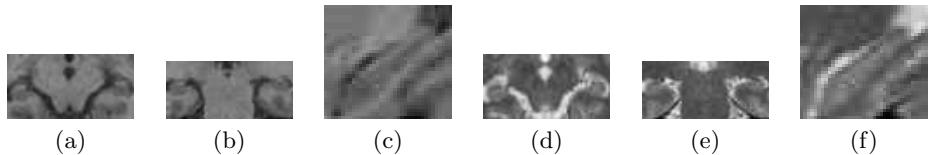
The hippocampus is a structure that is located in the medial temporal lobe of the human brain and is considered to be a part of the limbic system: it is divided into three parts: head, body, tail. The volume of the hippocampus is important in the studies in schizophrenia research. See Fig. 1 for a 3D view of the shape of the two hippocampi and a segmented sagittal slice.

There are several other techniques, which segment the hippocampus described in the literature. There are methods which depend on active shape models (ASM) or point distribution models (PDM) [1], the theory behind those kind of models is described in [2], [3]. The shape of the region that is segmented is changed iteratively until a fit is found. A good startposition is important for this segmentation to work. Regional fluid registration has also been used for segmentation of the hippocampus [4]. There are examples of matching the hippocampus to a brain atlas [5], [6].

Three MR modalities are used for the segmentation of the hippocampus, T1-weighted, T2-weighted, and a continuously classified stereo image. See Fig. 2 for a



**Fig. 1.** Segmented hippocampi viewed as contours. a) Contours in 3D. b) Contours in the axial plane



**Fig. 2.** Slices of the different modalities in the bounding box.: a) A T1-weighted axial slice. b) A T1-weighted coronal slice. c) A T1-weighted sagittal slice. d) A T2-weighted axial slice. e) A T2-weighted coronal slice. f) A T2-weighted sagittal slice

view of slices in the two different modalities used in the segmentation algorithm in this paper. Only what is within in the bounding box is showed. The classified image is used by the ANN method in BRAINS2, the two other modalities are used by grey-level morphology segmentation. Hippocampus is, for several reasons, a difficult region to delineate automatically. Since it is a bent structure, the cross-sections of the hippocampus change considerably in all three planes, and some parts of the hippocampus are very difficult to separate from surrounding brain structures on MR images. The hippocampus is easy to segment on the most lateral slice in the sagittal plane. On more medial parts of the hippocampus, the head and tail are seen as two independent structures. On proceeding further from the medial parts, the hippocampus becomes progressively harder to delineate.

Before the segmentation algorithm described herein can be applied, the hippocampus must be pre-segmented using an ANN approach in the program BRAINS2 [7] from the Mental Health Clinical Research Center, The University of Iowa College of Medicine and Hospitals and Clinics, Iowa City USA. This segmentation involves registration [8] of the T1-weighted and T2-weighted images and aligning them in the Talairach atlas space with the brain realigned in the AC-PC line. The AC-PC-plane is a line from one structure in the brain, the anterior commissure, to another part of the brain, the posterior commissure. The brain is then resized to fit an orthogonal frame based around the origin. The resizing is done using a 6-point linear transformation, and these methods therefore have a limited capability of correcting for any variations other than size. Anatomical localisation of an area of interest is then achieved by referring its

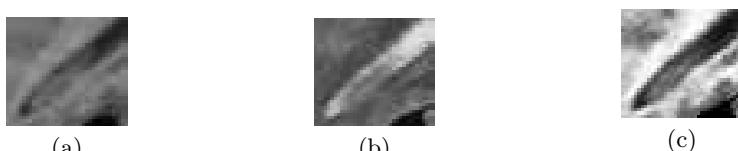
stereotactic co-ordinates in an atlas textbook, e.g. the Talairach and Tournoux atlas [9].

When the T1-weighted image has been set in the Talairach atlas space, and the T2-weighted images has been registered to the realigned T1-weighted image, the images are classified using a multimodal discriminant classifier. This classification recudes the variability in signal intensity across individual images sets [10]. After realignment the data is reduced to 8-bit data, which allows for only 256 grey-levels. After this classification there are specific intervals for the different kinds of tissue in the brain. These values are: cerebral spinal fluid (CSF) 10–70, grey matter (GM) 71–190, and white matter (WM) 191–250. Each voxel in the stereo image is assigned an intensity value based on the weights assigned by the discriminant function, reflecting the relative combinations of CSF, GM, and WM in a given voxel, producing a continuous classified image. There is also a discrete classified image: mostly white matter voxels are assigned the intensity 250, mostly grey matter voxels are assigned the intensity 130, and CSF voxels are assigned an intensity of 10.

For the segmentation of the hippocampus artificial neural nets [11], [12] (ANN) are being used. The ANN uses a fully connected feed-forward neural net with three layers. The output node was used as a "fuzzy value" to indicate if a voxel was in the ROI or not. The search space for the ANN in the normalised space was determined by creating a mask representing all manual traces in the training set. The input vector for the ANN consists of the intensity from the current voxel and also from neighbouring voxels. A bounding box is used to limit the search and contains both the left and right hippocampus. If the box is much larger than the hippocampus the ANN will fail in the segmentation process, which may result in an extra region close to the hippocampus. When the bounding box has been selected, the neural net generates a mask.

See Fig. 3 for sagittal slices of T1-weighted and T2-weighted data and the resulting slice from the classified volume used by the ANN method to generate a mask of the hippocampus. The classified volume is used by the ANN method to generate a mask of the hippocampus. The mask generated is smaller than the hippocampus, sometimes only 60% of the volume found when compared to a manual segmentation.

The ANN was trained from hippocampi manually traced in the sagittal plane. There are separate windows for coronal, axial, sagittal, and 3D ROI views. For a detailed explanation on how this should be done, see elsewhere [13]. An experienced tracer can delineate the hippocampus in about 60-90 min. The combined



**Fig. 3.** Sagittal slices of the different modalities in the bounding box.: a) A T1-weighted slice. b) A T2-weighted slice. c) A classified stereo slice

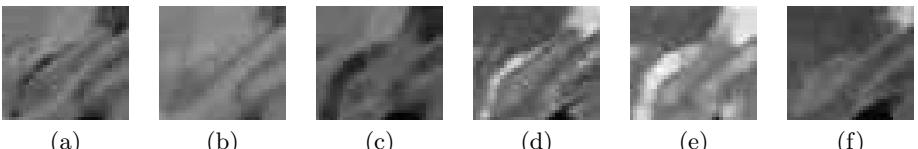
ANN and grey-level morphology approach only takes a few minutes to run on an ordinary workstation. The manual correction needed is then reduced to small corrections on a few slices only.

## 2 Methods

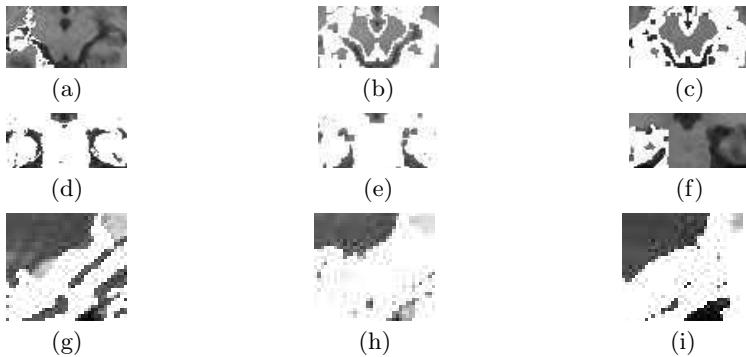
### 2.1 Overview

The method is based on pre-segmented data and grey-level morphology combined with anatomical knowledge and binary morphology. In addition to the original MRI-data, T1-weighted and T2-weighted, called **OrgImageT1** and **OrgImageT2**, four volumes are calculated using grey-level morphology [14], [15], [16]. The original MRI-data is grey-level-eroded using a  $3 \times 3 \times 3$  structure element; the new volumes are called **T1MinImage** and **T2MinImage**. The original MRI-data is also grey-level-dilated using a  $3 \times 3 \times 3$  structure element; those volumes are called **T1MaxImage** and **T2MaxImage**. See Fig. 4 for examples how these volumes may look. From the **T1MinImage** the foreground is thresholded using kernel density estimates (continuous histogram (KDE)) [17], [18], [19]. From this continuous histogram the second derivate is calculated and the lowest greatest maxima is selected. Initially the lateral outermost and innermost parts of the hippocampus in the sagittal plane are found. Further on, more of the head and tail of the hippocampus are found. Then the whole hippocampus is segmented in sagittal, coronal, and axial directions using information from the T1-weighted and the T2-weighted volume combined with grey-level eroded and dilated versions of the two modalities. The grey-levels are reduced to five grey-levels using a KDE approach for each slice of the images. Then the grey-level reduced versions of two modalities are thresholded between the highest and lowest grey-level found in the pre-segmented input mask. The result from the thresholding is then ANDed together, and also combined with the possible foreground.

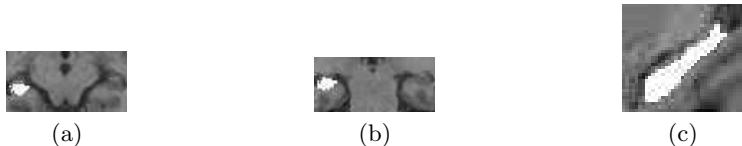
A sequence of binary morphology operations is then performed combined with labelling and flagging using the pre-segmented volume. See Fig. 5 for an example for the result of the segmentation algorithm. The results of the segmentation are then ANDed together. The result is then labelled and flagged using the pre-segmented volume in all three directions. See Fig. 6.



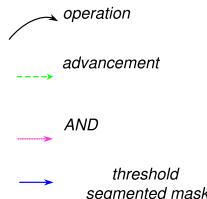
**Fig. 4.** Sagittal slices of the different modalities and in the bounding box.: a) A T1-weighted slice. b) A T1-weighted grey-level dilated slice. c) A T1-weighted grey-level eroded slice. d) A T2-weighted slice. e) A T2-weighted grey-level dilated slice. f) A T2-weighted grey-level eroded slice



**Fig. 5.** Result of slices of the different modalities in the bounding box: a) A segmented axial slice. b) A segmented grey-level dilated axial slice. c) A grey-level eroded axial slice. d) A segmented coronal slice. e) A segmented grey-level dilated coronal slice. f) A grey-level eroded coronal slice. g) A segmented sagittal slice. h) A segmented grey-level dilated sagittal slice. i) A grey-level eroded sagittal slice



**Fig. 6.** Result of segmentation with ANN and grey-level morphology, generated by ANDing the result from Fig. 5a–i): a) A axial slice. b) A coronal slice. c) A sagital slice



**Fig. 7.** Description of arrows used in the flow-charts

## 2.2 Segmentation Algorithms

The important steps of the algorithm are described using flow-charts. All binary dilations and erosions use a  $3 \times 3$  structure element. All grey-level dilations and erosions use a  $3 \times 3 \times 3$  structure element. Unless otherwise stated, operations on consecutive lines are performed in sequence using previous result. See Fig. 7 for a description of the arrows used in the flow-charts.

### Foreground detection algorithm

The Foreground is found from **T1MinImage** using a KDE approach.

**input:**

A grey-level eroded T1-weighted volume, **T1MinImage**.

**output:**

A mask with possible brain matter, **Matter**.

using the histograms

generate KDE from **T1MinImage**

find 2 largest local max in 2nd derivative sort

select threshold above 1st max → **Matter**

end;

### Lateral part of the hippocampus in sagittal slices

The algorithm is applied from the last sagittal slice on the ANN segmented volume containing hippocampus and continues laterally until there is no more hippocampus. See Fig 8 and Fig. 9 for flow-charts of the algorithm.

**input:**

T1-weighted MR data of the brain, **T1OrgImage**.

T2-weighted MR data of the brain, **T2OrgImage**.

Mask with segmented hippocampus, **AnnSegmMask**.

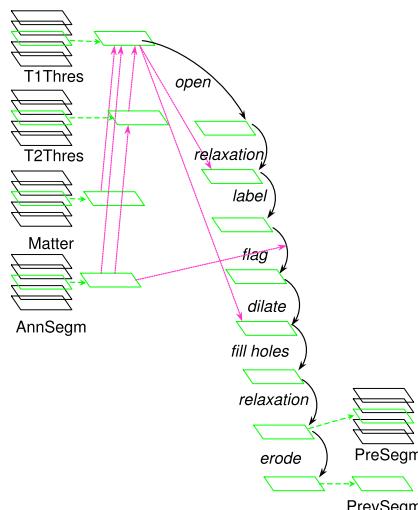
Mask with possible brain matter, **Matter**.

**output:**

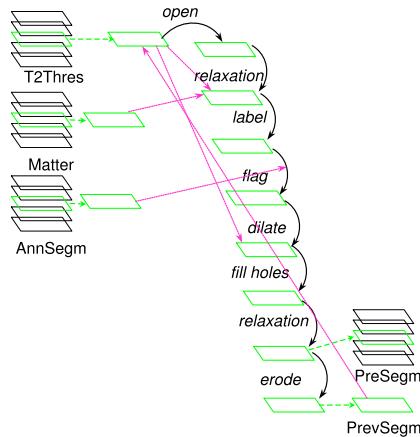
Mask with more lateral hippocampus, **PreSegmMask**.

### Hippocampus head and tail segmentation algorithm

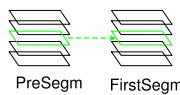
This algorithm is applied on the original data, grey-level eroded data, and grey-level dilated data; the results from those are then ANDed together. It starts at



**Fig. 8.** Lateral part of the hippocampus in sagittal slices, start slice



**Fig. 9.** Lateral part of the hippocampus in sagittal slices, traversing volume



**Fig. 10.** Hippocampus head and tail segmentation algorithm, start slice

the last slice containing hippocampal tissue, unless that is the first or last slice within the bounding box. It is also applied in coronal and axial directions in the head and tail, and the results from the two directions and two positions are ORed together. See Fig 10 and Fig. 11 for flow-charts of the algorithm.

#### input:

T1-weighted MR data of the brain, **T1Image**.

T2-weighted MR data of the brain, **T2Image**.

Mask with pre-segmented hippocampus, **PreSegmMask**.

A mask with possible brain matter, **Matter**.

#### output:

Mask with first segmented hippocampus, **FirstSegmMask**.

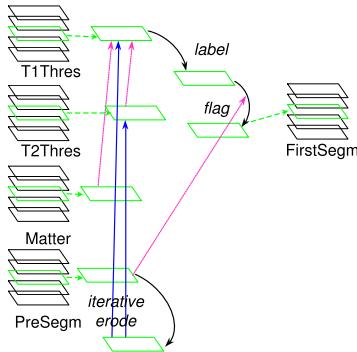
#### Hippocampus segmentation algorithm

This algorithm is applied on the original data, grey-level eroded data, and grey-level dilated data, and the results from those are ANDed together. It is also applied in sagittal, coronal, and axial directions, and the results from the three directions are ANDed together. See Fig 12 for a flow-chart of the algorithm.

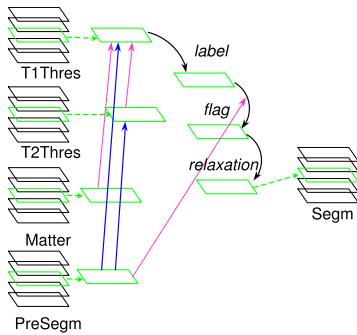
#### input:

T1-weighted MR data of the brain, **T1Image**.

T2-weighted MR data of the brain, **T2Image**.



**Fig. 11.** Hippocampus head and tail segmentation algorithm, traversing volume



**Fig. 12.** Hippocampus segmentation algorithm

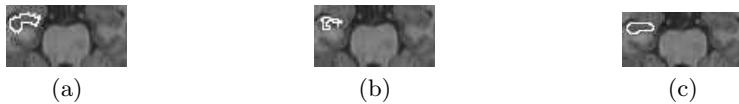
Mask with pre-segmented hippocampus, **FirstPreSegmMask**.  
A mask with possible brain matter, **Matter**.

**output:**

Mask with more segmented hippocampus, **SegmMask**.

### 3 Results

After testing the method on thirty data sets, the method seems promising: compared to the original ANN method, its segmentation of the hippocampus is much closer to the segmentation obtained by an experienced scientist. Most often the ANN segmentation only finds about 60% of the volume compared to the result from the human tracer, whereas the combined method finds at least 90% and up to 95% of the volume found by the human tracer. The new method does not remove the human intervention completely, but minimises the manual corrections to a few slices compared to almost half the slices.



**Fig. 13.** Result of segmentation on an axial slice: a) Hand-traced slice. b) Slices segmented with ANN. c) Slice segmented with ANN and grey-level morphology

Since the manual drawing is only done in sagittal slices there is a risk that the consecutive slices may drift. In Fig. 13 it can be seen that the manually traced hippocampus is more jagged than the combined method. The ANN method obviously misses large areas and the combined method can expand those areas to a better segmentation of the hippocampus.

To reduce manual drawing, a 3D-approach using grey-level morphology was implemented. In this paper, earlier research on grey-level based segmentation by the first author is used [20], [21], combining an imperfect segmentation tool using ANNs with an algorithm needing some information in each slice to be segmented, in order to create a new and better algorithm.

## 4 Future Work

The development of the hippocampus segmentation will continue in order to reduce human interaction even more. There will also be further development of segmentation algorithms for other structures, using grey-level morphology and including artificial neural nets (ANN) from BRAINS2.

## References

1. Kelemen, A., Szekely, G., Gerig, G.: Elastic Model-Based Segmentation of 3-D Neuroradiological Data Sets. *IEEE Transactions on Medical Imaging* **18** (1999) 828–839
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active Shape Models — Their Training and Application. *Computer Vision and Image Understanding* **61** (1995) 38–59
3. Cootes, T.F., Taylor, C.J.: Active Shape Models. Draft from a forthcoming report on ASM (1998)
4. Crum, W.R., Scähill, R.I., Fox, N.C.: Automated Hippocampal Segmentation by Regional Fluid Registration of Serial MRI: Validation and Application in Alzheimer's Disease. *NeuroImage* **13** (2001) 847–855
5. Haller, J.W., Bannerjee, A., Christensen, G.E., Gado, M., Joshi, S., Miller, M., Sheline, M.I., Vannier, M.W., Csernansky, J.G.: Three-Dimensional Hippocampal MR Morphometry with High-Dimensional Transformation of a Neuroanatomic Atlas. *Radiology* **51** (1997) 993–999
6. Fischl, B., Salat, D.H., Busa, E., Albert, M., Haselgrave, M.D.C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., Montillo, A., Makris, N., Rosen, B., Dale, A.M.: Whole Brain Segmentation: Automated Labeling of Neuroanatomical Structures in the Human Brain. *Neurotechnique* **33** (2002) 341–355

7. Magnotta, V.A., Harris, G., Andreasen, N.C., O'Leary, D.S., Yuh, W.T., Heckel, D.: Structural MR Image Processing using the BRAINS2 Toolbox. *Computerized Medical Imaging, Graphics* **26** (2002) 251–264
8. Woods, R.P., Grafton, S.T., Watson, J.D., Sicotte, N.L., Mazziota, J.C.: Improved Methods for Image Registration. *Journal of Computer Assisted Tomography* **22** (1998) 153–165
9. Talairach, J., Tournoux, P.: Co-planar Stereotactic Atlas of the Human Brain: 3-dimensional Proportional System—an Approach to Cerebral Imaging. Thieme Verlag, Stuttgart/New York (1988)
10. Harris, G., Andreasen, N.C., Cizadlo, T., Bailey, J.M., Bockholt, H.J., Magnotta, V.A., Arndt, S.: Improving Tissue Classification in MRI: A Three-Dimensional Multispectral Discriminant Analysis Method with Automated Training Class Selection. *Journal of Computer Assisted Tomography* **23** (1999) 144–154
11. Magnotta, V.A., Heckel, D., Andreasen, N.C., Cizadlo, T., Corson, P.W., Ehrhardt, J.C., Yuh, W.T.: Measurement of Brain Structures with Artificial Neural Networks: Two- and Three-Dimensional Applications. *Radiology* **211** (1999) 781–790
12. Pierson, R., Corson, P.W., Sears, L.L., Alicata, D., Magnotta, V., O'Leary, D.S., Andreasen, N.C.: Structural MR Image Processing Using the BRAINS2 Toolbox. *NeuroImage* **17** (2002) 61–76
13. Pantel, J., O'Leary, D.S., Cretsinger, K., Bockholt, H.J., Keefe, H., Magnotta, V.A., Andreasen, N.C.: A New Method for the In Vivo Volumetric Measurement of the Human Hippocampus with High Neuroanatomical Accuracy. *Hippocampus* **10** (2000) 752–758
14. Serra, J.: *Image Analysis and Mathematical Morphology*. Volume 1. Academic Press (1982)
15. Serra, J., ed.: *Image Analysis and Mathematical Morphology Volume 2: Theoretical Advances*. Volume 2. Academic Press (1988)
16. Gonzalez, R.C., Woods, R.E.: 8. In: *Digital Image Processing*. 2 edn. Addison-Wesley Publishing Company, Inc. (2002) 550–560
17. Lindblad, J.: Histogram Thresholding using Kernel Density Estimates. In: *Proceedings of SSAB (Swedish Society for Automated Image Analysis) Symposium on Image Analysis*, Halmstad, Sweden. (2000) 41–44
18. Rosenblatt, M.: Remarks on Some Nonparametric Estimates of a Density Function. *Annals of Mathematical Statistics* (1956) 642–669
19. Parzen, E.: On Estimation of Probability and Mode. *Annals of Mathematical Statistics* **33** (1962) 1065–1076
20. Hult, R., Bengtsson, E., Thurfjell, L.: Segmentation of the Brain in MRI Using Grey Level Morphology and Propagation of Information. In: *Proceedings of 11<sup>th</sup> Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland. Volume I. (1999) 367–373
21. Hult, R.: Grey-level Morphology Based Segmentation of MRI of the Human Cortex. In: *Proceedings of ICIAP01 11th International Conference on Image Analysis and Processing*, Palermo, Italy. (2001) 578–583

# Combined Segmentation and Tracking of Neural Stem-Cells

K. Althoff, J. Degerman, and T. Gustavsson

Department of Signals and Systems,  
Chalmers University of Technology,  
S-412 96 Gothenburg, Sweden

{althoff, degerman, gustavsson}@s2.chalmers.se  
<http://www.s2.chalmers.se/research/image/>

**Abstract.** In this paper we analyze neural stem/progenitor cells in an time-lapse image sequence. By using information about the previous positions of the cells, we are able to make a better selection of possible cells out of a collection of blob-like objects. As a blob detector we use Laplacian of Gaussian (LoG) filters at multiple scales, and the cell contours of the selected cells are segmented using dynamic programming. After the segmentation process the cells are tracked in the sequence using a combined nearest-neighbor and correlation matching technique. An evaluation of the system show that 95% of the cells were correctly segmented and tracked between consecutive frames.

## 1 Introduction

Combining the segmentation and tracking of objects in an image sequence has the potential of producing faster and more reliable results than keeping the segmentation and tracking separate. This is especially true when dealing with non-deformable objects or when the number of objects is constant throughout the sequence. However, even when tracking cells, that are usually deformable and also have the ability to split or disappear out of view, an interleaved segmentation and tracking procedure could be useful. A popular method that combines the segmentation and tracking is active contours (snakes), previously used for cell segmentation/tracking in e.g. [1],[2]. A significant drawback with the traditional snake is its inability to deal with splitting cells. In these situations, segmentation and tracking using level set methods are better [3]. However, in cases where previously separated cells move so close together that the boundary between them becomes blurred, a level-set method will merge the two cells into one while the traditional snake would keep the two contours separated. A method which allows for cells to split while keeping clustered cells separate is topology snakes [1],[4]. Another limitation of active contours is that it requires a high sampling rate, so that the cells do not move long distances or exchange positions between two consecutive frames.

Another technique combining segmentation and tracking is demonstrated in [5], where the directional movement of cells induced by a direct current (gal-

vanotaxis) is studied. Standard thresholding combined with clustering results in a coarse segmentation, which is refined further using the result of an association/tracking algorithm. Tracking is done using a modified version of the Kalman filter. This method works well because of the known directional movement of the cells. However, assumptions about a cell's state equations are potentially risky in cases where little is known about the laws governing the cell motion, and when the purpose of the analysis is to gain more information about cell kinematics.

In this work we track neural stem/progenitor cells *in vitro* in order to better understand the mechanisms of the differentiation of the neural stem/progenitor cells. To detect the cells, we use multi-scale Laplacian of Gaussian filters to find blob-like structures of different size in the image. Then we use information about the cells' previous positions to decide which detected blobs correspond to real cells. Using the positions of the selected blobs, the contours of the cells are segmented using dynamic programming. To track the cells we use a combination of a nearest-neighbor and a correlation matching technique.

## 2 Material

Nine image sequences of the neural stem/progenitor cells were used to evaluate the automatic cell tracking system. The images had been acquired every 10 minutes for 48 hours using a time-lapse brightfield microscopy system described elsewhere [6]. Each sequence contained 288 images of size 634x504 pixels. After the last image was acquired, the cells were stained with antibodies making it possible to differentiate between glial progenitor cells, type-1 and type-2 astrocytes when viewed in a fluorescence microscope.

## 3 Cell Segmentation

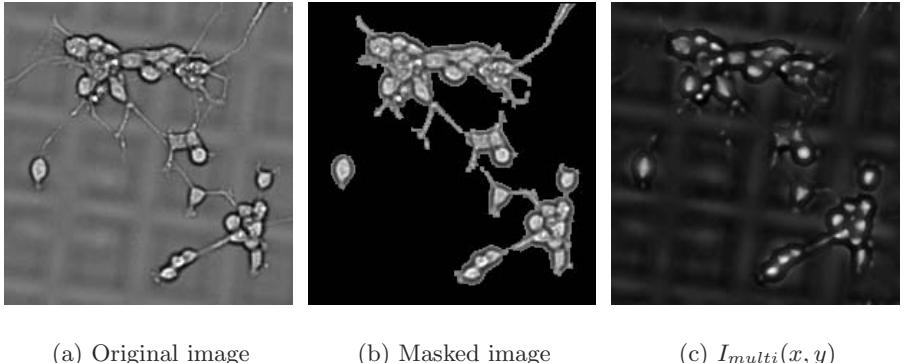
The automatic segmentation and tracking algorithm is an iterative process that for every image in the sequence can be described as follows:

1. Separate background from cell regions.
2. Find centroids of all blob-like objects in the image.
3. Select the centroids of the blobs that are most likely to be cells.
4. Segment the cells using dynamic programming.
5. Assign the segmented cells to the established tracks using the auction algorithm.

Steps 1-4 are described in this section and the last step is described in the next section. The segmentation and tracking is performed backwards, starting at frame 288, since the information about what kinds of cells the stem/progenitor cells have become is given subsequent to acquiring the last frame. In the initial frame (288) the segmentation is done using step 1-4, although step 2 is performed differently, see Sect. 3.3.

### 3.1 Background Removal

The first step in the segmentation process is to separate the cells from the background. This is efficiently done using the image variance, calculated according to Wu et. al. [7] with the window size set to  $3 \times 3$  pixels. The variance image is then thresholded using Kittler's and Illingworth's method [8]. To create the final mask,  $I_{mask}$ , the morphological operation "close" is applied to the binary image and then holes smaller than 100 pixels are filled and regions smaller than 55 pixels are removed. Figure 1(a) shows an example of part of an image, and Fig. 1(b) the corresponding masked image.



**Fig. 1.** (a) An example of part of an image ( $201 \times 201$  pixels). (b) The masked original image. (c)  $I_{multi}(x, y)$  calculated according to (3)

### 3.2 Blob Detection

To detect blob-like objects in an image, we convolve the image with a Laplacian of Gaussian (LoG) filter:

$$\hat{I}_k(x, y) = I(x, y) * L(x, y; \sigma_k) \quad (1)$$

where  $L(x, y; \sigma_k)$  is the LoG filter for 2D images and  $x$  and  $y$  are the spatial coordinates and  $\sigma$  determines the width of the filter.

Since the blob-like objects in our images, the cells, are of different size, we need to apply multiple filters of different scale, i.e. different  $\sigma$ , to detect all cells. In this work, we use  $2 \leq \sigma \leq 8$  with a step size of 0.33. To avoid the problem that the response from the LoG filter decreases with increasing scale, we replace  $L(x, y; \sigma_k)$  in (1) with the normalized LoG filter [9],[10]:

$$\tilde{L}(x, y; \sigma) = \sigma^2 \cdot L(x, y; \sigma). \quad (2)$$

$\hat{I}_k(x, y)$  will have high intensities in pixels close to the center of blob-like objects of size proportional to  $\sigma_k$ . We then create a new image,  $I_{multi}(x, y)$ , by maximizing image intensity over scale:

$$I_{multi}(x, y) = \max\{\hat{I}_2(x, y), \hat{I}_{2.33}(x, y), \dots, \hat{I}_8(x, y)\}. \quad (3)$$

Figure 1(c) shows  $I_{multi}(x, y)$  for the image shown in Fig. 1(a). For more details on blob-detectors or multi-scale image analysis, see e.g. [9] and [10].

### 3.3 Choosing Cell Points

In accordance with [9] and [10], we consider the local maxima pixels in  $I_{multi}(x, y)$  to be the center points of the existing blobs. To get a more reasonable result we first calculate  $\tilde{I}_{multi}(x, y)$  by smoothing  $I_{multi}(x, y)$  with a Gaussian kernel (size 5x5,  $\sigma=3$ ). We also remove all local maxima belonging to the background using the mask calculated in Sect. 3.1. Fig. 2(a) shows  $\tilde{I}_{multi}(x, y)$  for the image shown in Fig. 1(a) with all the local maxima found. To use the information about the previous cell positions in the sequence, we assign every cell present in the previous image to one local maxima each and increase the intensity of  $\tilde{I}_{multi}(x, y)$  in those local maxima. The assignment problem is solved using Bertsekas auction algorithm [11] and the assignment weights are calculated according to:

$$a(c, n) = \begin{cases} 10 & \text{if } \delta(c, n) < 0.1 \\ \frac{1}{\delta(c, n)} & \text{if } 0.1 \leq \delta(c, n) \leq 20 \\ -10 & \text{if } \delta(c, n) > 20 \end{cases} \quad (4)$$

where  $c$  is the index of the cell in the previous image,  $n$  is the index of the local maximum and  $\delta(c, n)$  is the spatial distance between the local maxima  $n$  and the centroid of cell  $c$  in the previous image. All the details of this assignment problem is not shown here, a similar assignment problem is shown in more detail in Sect. 4.1. The intensity of  $\tilde{I}_{multi}(x, y)$  (with intensities ranging from 0 to 255) in the selected points are increased with  $\beta_n$ :

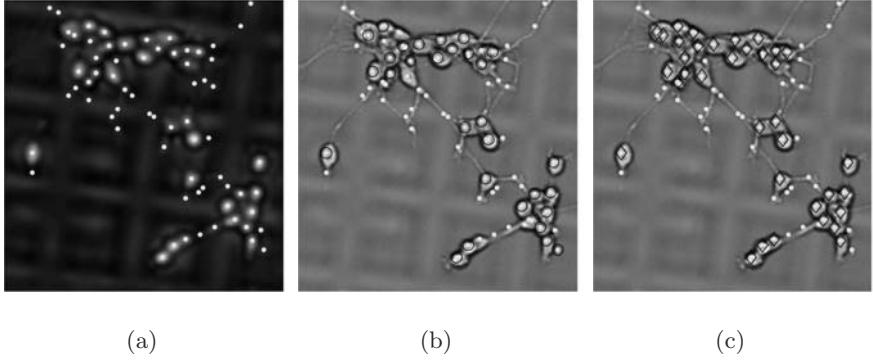
$$\beta_n = \begin{cases} \frac{1}{\delta(c_n, n)} \cdot 50 & \text{if } \delta(c_n, n) \geq 1 \\ 50 & \text{if } \delta(c_n, n) < 1 \end{cases} \quad (5)$$

where  $\delta(c_n, n)$  is the spatial distance between the local maximum  $n$  and the centroid of the cell in the previous image that was assigned to the local maximum  $n$ . Figure 2(b) shows the original image with all local maxima marked with dots and the local maxima where the intensity of  $\tilde{I}_{multi}(x, y)$  is increased are shown with circles. The intensities of the modified  $\tilde{I}_{multi}(x, y)$  in all the local maxima were then thresholded using Otsu's method [12], so that the most likely cell positions were selected, see Fig. 2(c).

In the initial frame we use all local maxima in the masked  $\tilde{I}_{multi}(x, y)$  as starting points for the dynamic programming. The result is then manually corrected where needed.

### 3.4 Cell Contour Segmentation

The selected points are sorted based on their intensity in  $\tilde{I}_{multi}(x, y)$ . Starting with the point with the highest intensity, the contour of the potential cell is



**Fig. 2.** (a) Smoothed  $I_{multi}(x, y)$  with all local maxima (maxima in background removed) marked with a point. (b) Original image with all local maxima marked with points and the points were the intensity of  $\tilde{I}_{multi}(x, y)$  is increased marked with circles. (c) Original image with all local maxima marked with points and the points chosen to be starting points for segmentation marked with diamonds

segmented using dynamic programming, see [13] for details. To be considered a valid cell, the segmented region has to be larger than 55 pixels and not extend more than 25% of its area into the region classified as the background. If the segmented region overlaps with a region previously segmented in that image, the overlap is removed from the last segmented region since it is assumed that the points previously used for segmentation (higher intensity in  $\tilde{I}_{multi}(x, y)$ ) give a more accurate segmentation result.

## 4 Cell Tracking

In this section the different parts of the tracking system are described. The tracking step consists of three parts: (A) solve the assignment problem, (B) examine unassigned tracks, (C) examine unassigned objects and update results.

### 4.1 The Assignment Problem

The asymmetric assignment problem, i.e. the problem of matching  $m$  objects with  $n$  tracks in an optimal way when  $n \neq m$ , can be formulated as:

$$\max \sum_{i=0}^n \sum_{j=0}^m a_{ij} \tau_{ij} \quad \forall (i, j) \in \Gamma \quad (6)$$

where  $a_{ij}$  is a assignment weight (i.e. the benefit of matching track  $i$  with object  $j$ ),  $\Gamma$  is the set of pairs  $(i, j)$  that can be matched and  $\tau_{ij}$  is defined as:

$$\tau_{ij} = \begin{cases} 1 & \text{if track } i \text{ is assigned to object } j \\ 0 & \text{otherwise} \end{cases}$$

Track  $i=0$  and object  $j=0$  are called the dummy track and the dummy object, respectively. The assignment is a one-to-one assignment, except for the dummy track and the dummy object which are allowed multiple assignments. Equation 6 was solved using the modified auction algorithm, see [11] for details. The assignment weights,  $a_{ij}$  used in (6), are calculated according to:

$$a_{ij} = \left( \frac{1}{\delta_{ij}} + \phi_{ij} + \psi_{ij} \right) \cdot w_i \quad (7)$$

where  $\delta_{ij}$ ,  $\phi_{ij}$  and  $\psi_{ij}$  are defined below, and  $w_i$  is the weight for track  $i$ , initially  $w_i = 1$ .

The distance measure,  $\delta_{ij}$ , is a function depending on the distance between object  $j$  and the last known object in track  $i$ :

$$\delta_{i,j} = C_1 \cdot \left( \sqrt{(x_j(t) - x_i(t+1))^2 + (y_j(t) - y_i(t+1))^2} + 0.1 \right) \quad (8)$$

where  $(x_j(t), y_j(t))$  is the centroid of object  $j$  at time  $t$ , and  $(x_i(t+1), y_i(t+1))$  is the centroid of the object belonging to track  $i$  in the previous image (at time  $t+1$  since the tracking is done backwards).  $C_1$  is a constant and the 0.1 term is added to avoid the denominator in (7) becoming zero.

The function  $\phi_{ij}$  is a measure of the correlation between object  $j$  and the last known object in track  $i$ . Assume that  $I_j$  is the smallest possible image containing object  $j$  in frame  $t$ , with the background set to zero. Then:

$$I_{corrj}(x, y) = \frac{C_2 \cdot \sum_k \sum_l I_j(s_x+k, s_y+l, t) \cdot I(x+k, y+l, t+1)}{\sqrt{\sum_k \sum_l I_j(s_x+k, s_y+l, t)^2} \cdot \sqrt{\sum_k \sum_l I(x+k, y+l, t+1)^2}}$$

for  $k = -s_x+1, \dots, s_x$  and  $l = -s_y+1, \dots, s_y$  (9)

where  $C_2$  is a constant and  $s_x$  is the number of rows in  $I_j$  divided by 2 and  $s_y$  is the number of columns in  $I_j$  divided by 2.  $\phi_{ij}$  can then be calculated:

$$\phi_{ij} = \max(I_{corrj}(x, y)) \quad \forall (x, y) \in \Omega \quad (10)$$

where  $\Omega$  is the set of pixels  $(x, y)$  belonging to the last known cell in track  $i$  in a previous frame.

The last function  $\psi_{ij}$  depends on the difference in area of object  $j$  and the last known object in track  $i$ :

$$\psi_{ij}^0 = C_3 \cdot \frac{A_i(t+1)}{|A_i(t+1) - A_j(t) + 1|} \quad (11)$$

where  $A_j(t)$  is the area of object  $j$  in frame  $t$  and  $A_i(t+1)$  is the area of track  $i$  in frame  $t+1$ .  $C_3$  is a constant. To avoid the risk of the denominator becoming zero, one is added. Also the contribution from  $\psi_{ij}^0$  is limited by:

$$\psi_{ij} = \begin{cases} \psi_{ij}^0 & \text{if } \psi_{ij}^0 \leq K \\ K & \text{if } \psi_{ij}^0 > K \end{cases} \quad (12)$$

where  $K$  is a constant.

Constants  $C_1$ ,  $C_2$ ,  $C_3$  and  $K$  should be chosen so that two, not so similar, cells in consecutive frames with centroids closer than about half a typical cell radius ( $r_{cell} \sim 7$  pixels) get a higher assignment weight than two cells further apart but very alike. In these investigations suitable values for  $C_1$ ,  $C_2$ ,  $C_3$  and  $K$  were found to be 0.03, 5, 1 and 3 respectively. These values give that the maximum value of  $\phi_{ij} + \psi_{ij} = C_2 + K = 8$ , which means that  $\delta_{ij}$  will give the largest contribution to the assignment weight for distances between two cells up to  $\sim 4$  pixels ( $\frac{1}{0.03 \cdot 8} - 0.1$ ).

## 4.2 Unassigned Tracks

There are several reasons as to why a track might not get assigned to a real object (i.e. are assigned to the dummy object):

1. Cells merge (or rather, one cell splits, since tracking is done backwards)
2. Cells disappear outside the image boundaries
3. Cells disappear into clusters
4. Errors in the segmentation

In the first three cases, the track should be terminated, while in the last case the segmentation should be corrected.

**Cells Merge.** Previous investigations have shown that most merged and merging cells have a characteristic appearance. Therefore there are three requirements for two cells,  $\alpha$  and  $\beta$ , in frame  $t+1$  to be considered to have merged into cell  $\gamma$  in frame  $t$ . First, the sum of the areas of cells  $\alpha$  and  $\beta$  should not differ more than 20% compared to the area of cell  $\gamma$ . Second, the mean intensity of the interior of cell  $\gamma$  has to be at least twice the mean intensity of the contour of cell  $\gamma$ . Last, cell  $\gamma$  must be fairly circular. Therefore, the ratio between the major and minor axis of an ellipse with the same normalized second central moments as cell  $\gamma$  has to be less than 1.5.

**Cells Disappear Outside the Image Boundaries.** If the center of the last known cell in track  $i$  is closer than 11 pixels to any of the image borders, it is considered very likely that the cell has moved out of the image in the current frame. However, the track is not terminated at once. Assignment weights are calculated according to (7) for the track in the next two frames, but the track weight,  $w_i$ , is decreased, making the track less attractive. If no cell is associated with the track for three frames, the track is terminated.

**Cells Disappears Into Clusters.** There is currently no way to resolve cells that are above or below the image plane. A vanished cell is considered to have disappeared into a cell cluster if the track is not near an image border, and if no segmentation error or merging of cells could be detected. Just as in the case with cells disappearing from within the image borders, the tracks are not immediately terminated, but remembered for another two frames and the track weight is decreased.

**Errors in the Segmentation.** In situations when two potentially merging cells fulfill the requirement made on the cell sizes, but fail on either one of the other two requirements (see Sect. 4.2) the segmentation might have failed to split two neighboring cells. Assume that cell  $\alpha_i(t+1)$  is the cell belonging to track  $i$  in the previous image,  $\gamma_k(t)$  is the potentially incorrectly segmented object belonging to track  $k$  in the current image, and  $\beta_k(t+1)$  is the cell assigned to track  $k$  in the previous image. The centroids of  $\alpha_i(t+1)$  and  $\beta_k(t+1)$  are then the starting points for the dynamic programming segmentation. The cells found through this segmentation are considered valid when they have a size  $> 55$  pixels, a mean intensity of the cell contour  $< 100$ , and a mean intensity of the interior of the cell  $> 130$ , otherwise the new segmentation result is rejected and the unassigned track is considered to belong to case 3, i.e. the cell is assumed to have joined a cluster.

Another potential error in the segmentation is that a cell is missed in the segmentation. However, based on the segmentation procedure, this means that the cell is either small ( $< 55$  pixels) or very unclear (not blob-like). Unfortunately the problem of detecting such cells is not yet solved, so in that case the cell would be assumed to have joined a cluster.

### 4.3 Unassigned Objects

Unassigned objects are divided into three groups:

1. Cells appearing from outside the image boundaries
2. Cells previously hidden in a cell cluster
3. False cells

If the center of an unassigned cell is closer than 11 pixels from any of the image borders, it is assumed to have appeared from outside the image and a new track is started. A new track is also started for objects assigned to group 2, i.e. objects appearing in the interior of an image, and fulfilling the criteria for valid cells (see Sect. 4.2). If an unassigned object does not belong to either group 1 or 2, it is likely that it is not a cell and therefore belongs to group 3. A new track is started but given a low track weight. However, if no cell can be associated with the track in the next frame, it is deleted from both frames.

## 5 Result

To evaluate the automatic segmentation and tracking system, nine image sequences were segmented and tracked throughout each sequence. The tracks for certain cells were then visually inspected and manually corrected where found necessary. The tracks that were manually corrected were chosen based on the result of the antibody staining, i.e. only classified cells were inspected. Since the tracks were not chosen based on ease-of-tracking or because cells remained within the field of view for the complete sequence they represent a fairly random selection of cells available in the final image of each time-lapse experiment. In total, 87 cell tracks were selected for manual correction and on average

they were present in 150 frames in their sequence before vanishing. The manually corrected sequences were compared with the corresponding automatically tracked sequences. In total only 18 out of the 87 (=20%) cell tracks were correctly segmented and tracked automatically through the whole image sequence. However, if we instead consider each cell-to-cell association in two consecutive frames as a separate event (the total number of cell-to-cell associations were 12944), it was found that 95% of the separate cell-to-cell associations were correct.

## 6 Discussion

By manual inspection of the images it is clear that many, although not all, of the ambiguities in a single image can be resolved by looking at the images before and/or after the current image. It therefore seems clear that information about the previous segmentation results should be used when segmenting the new image, but the difficult question is how much impact that information should have on the new segmentation result. Since the number of cells appearing or disappearing between two consecutive frames is significant, we can not let the influence of the previous segmentation result be too large because that would undoubtedly lead to a lot of errors. By letting the previous segmentation result aid in the decision of what blobs to be segmented we favor the blobs found close to where there were cells previously. This will keep cell tracks of slow moving cells intact even in images where the cell might be difficult to distinguish, while still allowing for blob-like cells appearing into the image to be detected. Although this procedure is working well, an improvement might be to use segmentation results from the next frame in the sequence as well, by going back and fourth in the sequence. This of course will be a quite time consuming procedure, since it requires each frame to be segmented at least twice.

The aim of this project is to use the data acquired from the sequence (cell migration, tendency to split, appearance etc) to gain a better understanding of the processes behind the differentiation of neural stem/progenitor cells into neurons, astrocytes or oligodendrocytes. Since information about the identity of the cells is only given in the last frame of the sequence it is of high importance that the cells tracks are not mixed up along the way in the sequence. Although our error rate is low regarding cell-to-cell associations, we saw that the number of tracks where the cell was correctly segmented and tracked through the whole sequence is too low to be able to use the results directly for e.g. migration modelling of the different cell types. A manual correction of the result is therefore necessary. However, manually adjusting the tracking or the segmentation in about 7-8 images on average per cell track (each cell track exist in 150 frames on average) seems reasonable and not too time-consuming. Therefore we conclude that our current system can be used in its present state for examining the very large amount of data needed for the neural stem/progenitor investigations.

## Acknowledgments

This project was partly funded by the Swedish Foundation for Strategic Research (SSF) under the VISIT program, No. A30-960626 and the Swedish Research Council (VR), No. 621-2001-3531.

## References

1. Zimmer, C., Labruyere, E., Meas-Yedid, V., Guillen, N., Olivo-Marin, J.C.: Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Transactions on Medical Imaging* **21** (2002) 1212–1221
2. Debeir, O., Camby, I., Kiss, R., Van Ham, P., Decaestecker, C.: A model-based approach for automated in vitro cell tracking and chemotaxis analyses. *Cytometry* **60A** (2004) 29–40
3. Mukherjee, D., Ray, N., Acton, S.: Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image Processing* **13** (2004) 562–572
4. Delingette, H., Montagnat, J.: Shape and topology constraints on parametric active contours. *Computer Vision and Image Understanding* **83** (2001) 140–171
5. Kirubarajan, T., Bar-Shalom, Y., Pattipati, K.: Multiassignment for tracking a large number of overlapping objects. *IEEE Transactions on Aerospace and Electronic Systems* **37** (2001) 2–21
6. Gustavsson, T., Althoff, K., Degerman, J., Olsson, T., Thoreson, A.C., Thorlin, T., Eriksson, P.: Time-lapse microscopy and image processing for stem cell research modeling cell migration. In: *Medical Imaging 2003: Image Processing*. Volume 5032. (2003) 1–15
7. Wu, K., Gauthier, D., Levine, M.: Live cell image segmentation. *IEEE Transactions on Biomedical Engineering* **42** (1995) 1–12
8. Kittler, J., Illingworth, J.: Minimun error thresholding. *Pattern Recognition* **19** (1986) 41–47
9. ter Haar Romeny, B.: *Front-End Vision & Multi-Scale Image Analysis*. Kluwer Academic Publishers (2003)
10. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attentions. *International Journal of Computer Vision* **11** (1993) 283–318
11. Bertsekas, D.: The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces* **20** (1990) 133–149
12. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-9** (1979) 62–66
13. Althoff, K., Degerman, J., Gustavsson, T.: Tracking neural stem cells in time-lapse microscopy image sequences. In: *Medical Imaging 2005: Image Processing*. (2005)

# Morphons: Paint on Priors and Elastic Canvas for Segmentation and Registration

Hans Knutsson and Mats Andersson

Medical Informatics, Dept. of Biomedical Engineering &  
Center for Medical Image Science and Visualization,

Linköping University, Linköping Sweden

{knutte,matsa}@imt.liu.se

[www.imt.liu.se/mi](http://www.imt.liu.se/mi)

**Abstract.** This paper presents a new robust approach for registration and segmentation. Segmentation as well as registration is attained by morphing of an  $N$ -dimensional model, the *Morphon*, onto the  $N$ -dimensional data. The approach is general and can, in fact, be said to encompass much of the deformable model ideas that have evolved over the years. However, in contrast to commonly used models, a distinguishing feature of the Morphon approach is that it allows an intuitive interface for specifying prior information, hence the expression *paint on priors*. In this way it is simple to design Morphons for specific situations.

The priors determine the behavior of the Morphon and can be seen as local data interpreters and response generators. There are three different kinds of priors: - *material parameter fields* (elasticity, viscosity, anisotropy etc.), - *context fields* (brightness, hue, scale, phase, anisotropy, certainty etc.) and - *global programs* (filter banks, estimation procedures, adaptive mechanisms etc.).

The morphing is performed using a dense displacement field. The field is updated iteratively until a stop criterion is met. Both the material parameter and context fields are addressed via the present displacement field. In each iteration the neighborhood operators are applied, using both data and the displaced parameter fields, and an incremental displacement field is computed.

An example of the performance is given using a 2D ultrasound heart image sequence where the purpose is to segment the heart wall. This is a difficult task even for trained specialists yet the Morphon generated segmentation is highly robust. Further it is demonstrated how the Morphon approach can be used to register the individual images. This is accomplished by first finding the displacement field that aligns the morphon model with the heart wall structure in each image separately and then using the displacement field differences to align the images.

## 1 Introduction

The history of image segmentation goes back to the very beginning of image processing and is still a topic of major concern. Naive thresholding based

approaches are basically abandoned and the need to incorporate strong prior information in non-trivial segmentation tasks is disputed by no one. A priori information in terms of allowable deformation of a prototype object has been frequently used in segmentation of 2D images and for tracking of deformable objects in video. Examples of such algorithms are deformable templates/prototypes [6, 7], trainable snakes [1], balloon models[2] and more recent approaches [10, 4, 3].

To fit a prototype to an image these methods generally use image related and geometric terms. The image features are typically based on edge detection for alignment of the object boundary. The geometric deformation of the prototype can e.g. be controlled by limiting the deformation of the object to a low order parametric motion model or introduce a triangular meshgrid and define the stiffness of the prototype using a FEM model.

Existing deformable models are however sensitive to the initial conditions of the computation and often get trapped in local minima. In such cases interactive measures are paramount to attain successful results. For most users interacting with the deformable model is complicated and non-intuitive. In addition there are typically no means for user interaction during the extraction process.

This paper presents a new method for segmentation which is both robust and has intuitive parameters settings.



**Fig. 1.** The Morphon acts like an elastic canvas with painted on smart local operators each determining where to go next

Figure 1 is meant to illustrate the basic mode of operation of the Morphon. The priors form a field with different individual priors at each location of an elastic canvas (of  $N$  dimensions). The priors determine how the neighborhood will be perceived and how this percept translates into an appropriate new location. Thus the canvas is morphed according to the outputs of the ‘smart operators’ that are ‘painted’ on the canvas.

The new method can also be used to obtain robust image registration benefitting directly from the features of the Morphon segmentation approach. In this case the Morphon model enables object specific regularization of the displacement fields, see section 4.

## 2 More on Morphons

It is helpful to group the priors into three groups:

### – Material

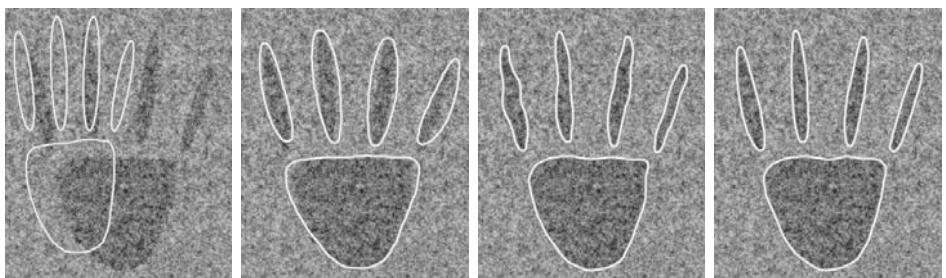
determining the local ‘material’ properties. The fact that the effect of these parameters are mediated through spatio-temporal filtering is a novel feature of the Morphon approach. The term ‘material’ need not be given a strict physical interpretation but gives intuitively useful associations. Typical properties that are locally determined are elasticity, viscosity, anisotropy and memory behavior.

### – Context

holding information that support interpretation of the image data. Typically one part of the context is of the same type as the data at hand, e.g. intensity. Other types of information that can be present in the context is, for example, local scale, orientation, phase, anisotropy and certainty.

### – Program

of what to do, how to use priors and data and what outputs to produce. Typical examples are filter banks, estimation algorithms and adaptive mechanisms.



**Fig. 2.** From left to right: Initial position of the ‘hand’ Morphon (white sketch) and image. Result using ‘stiff’ material. ‘Medium stiff’ material. Anisotropic material

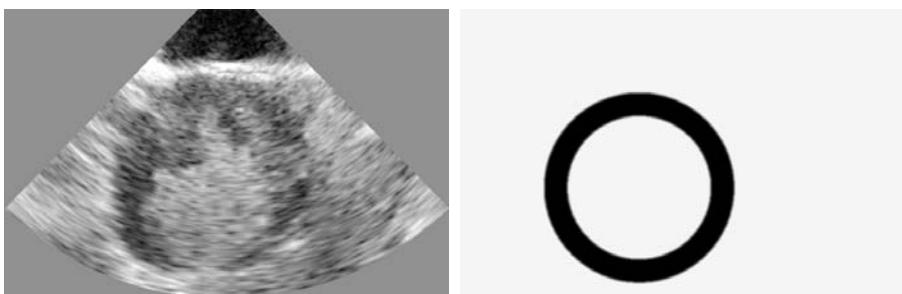
Figure 2 shows the initial situation and three different segmentations of a stylized ‘hand’ object. The main mode of freedom of the hand is that the fingers can be spread. The initial Morphon hand has the fingers much closer together than the hand in the image. All three segmentation results are in some sense successful in that all fingers are found. However, fig. 2 shows how the segmentation result can be significantly improved by using material parameters that reflect the properties of the object.

If a stiff material is used the Morphon fingers still can be spread apart but they will get thicker as the material will resist non-linear displacement attempts. If the stiffness is reduced the fingers are allowed to retain their original thickness

but the noise in the image now introduces wiggles (which is not a natural mode of variation of fingers). The solution is to give the material anisotropic stiffness, i.e. a material that is ‘soft’ in the finger spreading direction but resists variations in displacement along the fingers, see the lower right result in fig. 2. This illustrates the straightforward relation between the modes of variation of the object class to be segmented and the ‘painting’ of the priors.

### 3 Heart Wall Segmentation from Ultrasound Images

As an initial evaluation of the performance, the Morphon approach is used to segment out the heart wall in an ultrasound image of a heart. The ultrasound image is shown in the left part of fig. 3. The presence of contrast agent in the blood makes the heart wall appear darker than the surrounding tissue and the blood pool. The dark objects in the interior of the heart is tissue not belonging to the heart wall. The image in the right part of fig. 3 defines the context field for the initial state of the Morphon. The heart wall is a priori expected to be a closed object with an approximate size and wall thickness. Note that this context field provides itself to a very simple interface to a human operator.

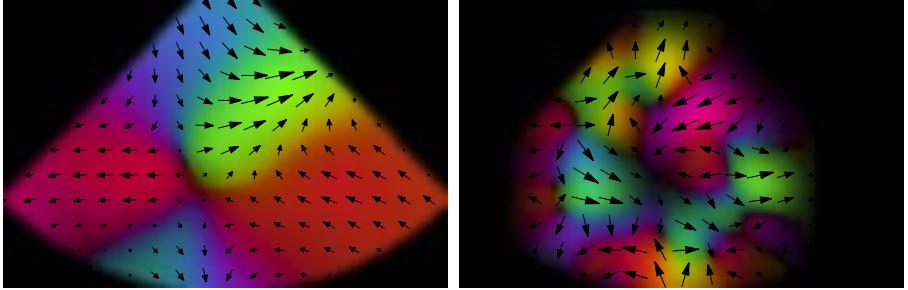


**Fig. 3.** Left: The original ultrasound image of a heart. Right: The Morphon context field

#### 3.1 Morphing the Morphon

The adaptation of the Morphon is an iterative process using a multitude of scales. A central part of the algorithm is to compute a dense displacement field that morphs the initial context fields of the Morphon onto the target, in this case the heart wall. The computation of the displacement field is not critical since the algorithm is iterative. There exists a large variety of displacement/motion estimation algorithms, pick your favorite. The authors prefer a phase-based quadrature filter approach which is briefly described in section 3.4.

A dense displacement field is accumulated from a number of iterations from coarse to fine scale. At each scale the present accumulated displacement field,  $d_a$ , is updated by an additional displacement field for the current scale,  $d_i$ .



**Fig. 4.** Left: Accumulated displacement field,  $\mathbf{d}_a$ . The mean value is subtracted. Right: Incremental displacement field for scale  $i$ ,  $\mathbf{d}_i$

Performing 2-5 iterations on each scale makes the precision of the incremental displacement field less critical. In fig. 4 the accumulated and incremental displacement fields are illustrated at an intermediary scale of the adaptation process. The method by which these fields and their associated certainties are combined and updated comprise a central part of the proposed algorithm.

### 3.2 Displacement Field Accumulation

For the moment we leave the details of the displacement and certainty computation and focus on the accumulation of the displacement fields with respect to the estimated certainties. For each iteration the following fields are computed.

Accumulated displacement:  $\mathbf{d}_a$ .      Accumulated certainty:  $c_a$ .

Incremental displacement:  $\mathbf{d}_i$ .      Incremental certainty:  $c_i$ .

Since the certainty field for iteration  $k$  comprise a total displacement corresponding to  $\mathbf{d}_a + \mathbf{d}_i$  it is natural to update the accumulated displacement field as:

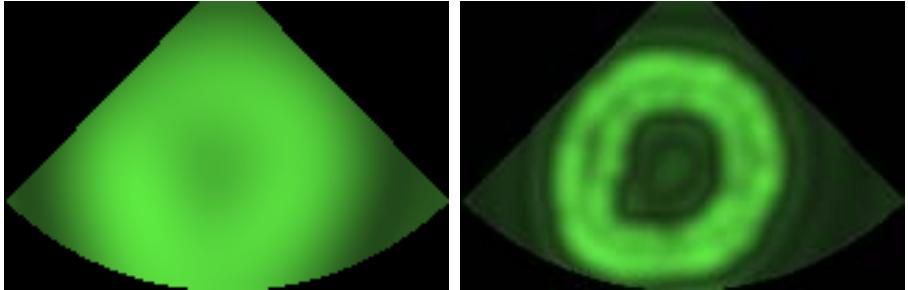
$$\mathbf{d}_a = \frac{c_a \mathbf{d}_a + c_i (\mathbf{d}_a + \mathbf{d}_i)}{c_a + c_i} \quad (1)$$

From eq. (1) it is clear that the impact of  $\mathbf{d}_i$  is very small unless the certainty for the current iteration,  $c_i$ , is significant compared to the accumulated certainty,  $c_a$ .

The adaptation of the accumulated certainty,  $c_a$ , is a bit more complicated as it would be desirable to have certainty estimates of the certainty estimates themselves. A straightforward method to solve this problem is simply to use  $c_a$  and  $c_i$  as their own certainties, i.e. update certainties as

$$c_a = \frac{c_a^2 + c_i^2}{c_a + c_i} \quad (2)$$

In practice it turns out that it is beneficial to increase the importance  $c_i$  for a fine resolution scale to maintain full adaptivity of the Morphon over all scales. Introducing the scale parameter,  $\rho$ , the accumulated certainty is computed as



**Fig. 5.** Left: certainty in an early stage of the morphing process. Right: certainty in the final state

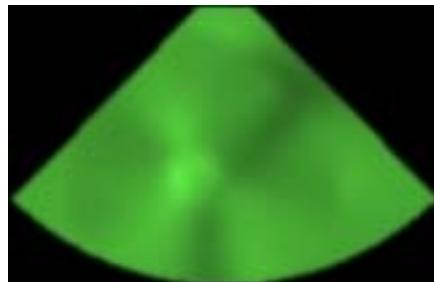
$$c_a = \frac{c_a^2 + (2^{-\rho} c_i)^2}{c_a + 2^{-\rho} c_i} \quad (3)$$

where  $\rho = 0$  at original resolution and  $\rho > 0$  at the coarser scales. In this example the scales are separated by half an octave comprising a total of 3 octaves in scale space. Figure 5 shows how the accumulated certainty evolves during the adaptation from coarse to fine scale.

### 3.3 Normalized Regularization of the Displacement Field

In the general case there exists a large class of displacement fields that result in practically identical Morphon context fields. In most cases we prefer the most simple displacement field. This provides a more robust algorithm. For this reason the incremental displacement field is subject to a regularization at each iteration. The regularization is performed by filtering using a Gaussian lowpass kernel,  $g$ , and normalized convolution with respect to  $c_i$ .

$$\mathbf{d}_i = \frac{(c_i \mathbf{d}_i) * g}{c_i * g} \quad (4)$$



**Fig. 6.** The determinant of the Jacobian of the accumulated motion field

The overall ‘stiffness’ of the morphon is defined by size of the Gaussian filter and the variance of  $g$  is decreased for finer scales. In this example the context field of the Morphon has isotropic stiffness.

To monitor the regularization of the displacement field the determinant of the Jacobian of the displacement field is computed describing the local area change of the Morphon context field. Negative values in the determinant field implies that the prototype is ‘folded over’. If this is the case a more powerful regularization is applied. Figure 6 shows the determinant of the Jacobian of the accumulated displacement field.

### 3.4 Phase Based Estimation of the Displacement Field

There exists a large variety of displacement/motion estimation algorithms that can be used for the computation of the displacement field. There is however one common issue that must be considered. Due to the aperture problem the displacement field can not be unambiguously estimated for all neighbourhoods. In these cases it is important to maintain as large degree of freedom as possible and not unnecessarily prevent displacements along the border of the Morphon.

In the present example the displacement estimate is, for each scale, based on conjugate products of quadrature filter [8] responses, i.e.  $\mathbf{Q} = \mathbf{q}_M \mathbf{q}_T^*$  where  $\mathbf{q}_M$  and  $\mathbf{q}_T$  are the responses for the present state of morphon context field and the target image respectively. These estimates are performed in four directions and the final displacement estimate,  $\mathbf{v}$ , is obtained as the solution of a least squares problem:

$$\min_{\mathbf{v}} \sum_{k=1}^4 \left[ c_k \hat{\mathbf{n}}_k^T \mathbf{T} (\hat{\mathbf{n}}_k d_k - \mathbf{v}) \right]^2$$

Where:

$$c_k = \sqrt{|\mathbf{Q}|} [1 + \cos(\arg(\mathbf{Q}))] \text{ Certainty in direction } k$$

$$\hat{\mathbf{n}}_k \quad \text{Direction of filter } k$$

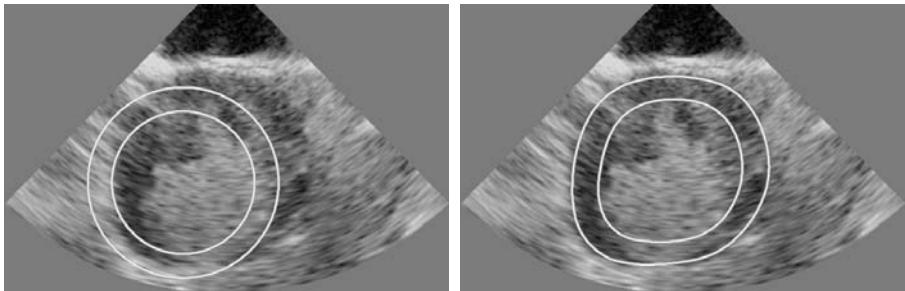
$$\mathbf{T} \quad \text{Local structure tensor, see [9, 5]}$$

$$d_k \propto \arg(\mathbf{Q}_k) \quad \text{Displacement estimate in direction } k.$$

$$\mathbf{v} = (v_x, v_y)^T \quad \text{Estimated displacement}$$

### 3.5 Segmentation Result

Figure 7 shows the initial and final position of the Morphon context field. The proposed performs extremely robust with respect to the noisy ultrasound image and the simple a priori information. The algorithm provides a simple and intuitive interface for an operator that is not specialized in image processing.



**Fig. 7.** Left: initial position of Morphon Right: final adaptation of prototype pattern

## 4 Morphon Based Ultrasound Sequence Registration

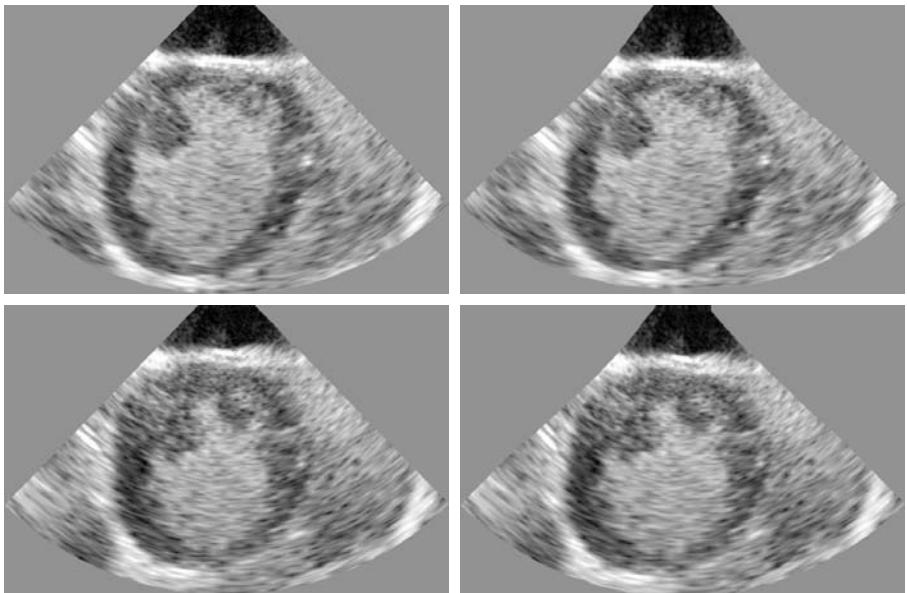
To demonstrate the use of Morphons for robust registration 25 frames of the ultrasound sequence in fig. 3 where registered relative to each other using Morphon based registration. The registration algorithm is based on the Morphon segmentation.

Each image in the sequence is segmented individually using the the Morphon segmentation described in the previous section and the resulting accumulated displacement fields,  $\mathbf{d}_a$ , are saved. The next step is to compute an average displacement field  $\bar{\mathbf{d}}_a$  over the entire sequence. By letting the average displacement field  $\bar{\mathbf{d}}_a$  act on the the original context field (right part in fig. 3) a new average Morphon for the sequence is computed, see fig. 8. The objective is to compute new displacement fields relative to this average Morphon to define the registration. The previous segmentation algorithm is consequently re-run using the new context field. However in this run the most coarse scales need not to be used and the regularization of the displacement fields are more conservative compared to the initial run. In a low noise environment of the image these restrictions have negligible effect on the adaptation while for high SNR's the spatial support from an additional dimension improves the robustness considerably.

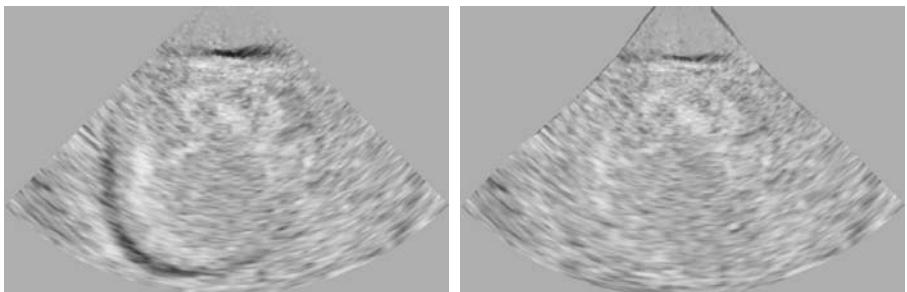
The registration between two images is finally obtained by the differences between their associated displacement fields. The left column of fig. 9 shows two



**Fig. 8.** Average Morphon context field used for registration



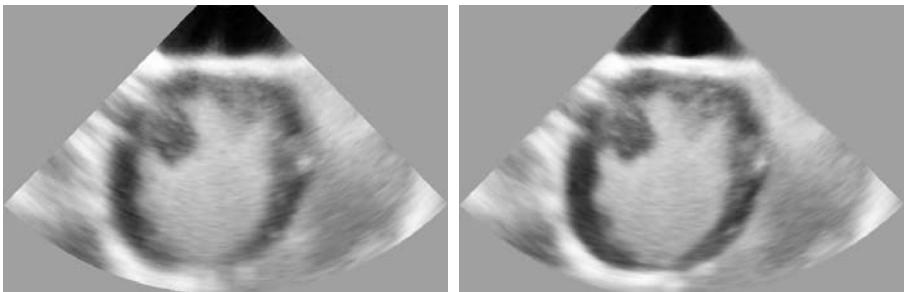
**Fig. 9.** Left column: Two consecutive images before registration. Right column: Same images after Morphon registration



**Fig. 10.** The difference of the two consecutive frames from fig. 9. Without registration (left) and using Morphon registration (right)

frames of the original sequence. The right column in the same figure show the result after the Morphon registration. Figure. 10 shows the difference between the images in fig. 9. The largest displacements occur at the lower left heart wall which is compensated by the registration algorithm.

Finally the sum of all 25 frames in the sequence are computed with and without registration. The result is displayed in fig. 11. The registered sequence provide a superior detail and contrast. Note that the heart wall can be clearly distinguished in the lower part of the image.



**Fig. 11.** Sum of 25 frames in the ultrasound sequence. Left: using no compensation. Right: Using Morphon registration to the average displacement field from the segmentation of the heart wall

## Acknowledgments

This work was funded by The Swedish Research Council (VR), the Swedish Agency for Innovation Systems (VINNOVA/NIMED) and Amersham Health. We are grateful to Lars Hoff, PhD and Birgitta Janerot Sjöberg, MD, PhD for providing the ultrasound data sequence and valuable clinical support.

## References

1. A. Baumberg and J. Hogg. Generating spatiotemporal models from examples. *Image and Vision comput.*, 14(8):525–532, 1996.
2. L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218, 1991.
3. D. Cremers. Bayesian approaches to motion-based image and video segmentation. 1st International Workshop on Complex Motion, DAGM, 2004.
4. D. Cremers and C. Schnörr. Motion competition: Variational integration of motion segmentation and shape regularization. German Conf. on Pattern Recognition (DAGM), 2002.
5. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
6. Anil K. Jain, Yu Zhong, and Sridhar Lakshmanan. Object matching using deformable templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(3):267–278, 1996.
7. G. Klein. *Deformable models for volume feature tracking*. PhD thesis, University of California, Berkeley, 1999.
8. H. Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.
9. H. Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH-ISY-I-1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.
10. Stan Sclaroff and John Isidoro. Active blobs: region-based, deformable appearance models. *Comput. Vis. Image Underst.*, 89(2/3):197–225, 2003.

# Efficient 1-Pass Prediction for Volume Compression

Nils Jensen<sup>1</sup>, Gabriele von Voigt<sup>2</sup>, Wolfgang Nejdl<sup>1</sup>, and Johannes Bernarding<sup>3</sup>

<sup>1</sup> Forschungszentrum L3S, Universität Hannover,  
Deutscher Pavillon, Expo Plaza 1,  
D-30539 Hannover, Germany  
[{jensen, nejdl}@l3s.de](mailto:{jensen, nejdl}@l3s.de)

<sup>2</sup> Regionales Rechenzentrum für Niedersachsen,  
Universität Hannover, Schloßwender Str. 5,  
D-30159 Hannover, Germany  
[vonvoigt@rrzn.uni-hannover.de](mailto:vonvoigt@rrzn.uni-hannover.de)

<sup>3</sup> Institut für Biometrie und Medizinische Informatik,  
Medizinische Fakultät der Otto-von-Guericke Universität Magdeburg,  
Leipziger Str. 44, D-39120 Magdeburg, Germany  
[johannes.bernarding@medizin.uni-magdeburg.de](mailto:johannes.bernarding@medizin.uni-magdeburg.de)

**Abstract.** The aim is to compress and decompress structured volume graphics in a lossless way. Lossless compression is necessary when the original scans must be preserved. Algorithms must deliver a fair compression ratio, have low run-time and space complexity, and work numerically robust. We have developed PR0 to meet the goals. PR0 traces runs of voxels in 3D and compensates for noise in the least significant bits by way of using differential pulse-code modulation (DPCM). PR0 reduces data to 46% of the original size at best, and 54% on average. A combination of PR0 and Worst-Zip (Zip with weakest compression enabled) gives reductions of 34% at best, and 45% on average. The combination takes the same or less time than Best-Zip, and gives 13%, respectively 5%, better results. To conduct the tests, we have written a non-optimised, sequential prototype of PR0, processed CT and MRI scans of different size and content, and measured speed and compression ratio.

## 1 Introduction

Volume compression means to reduce the mean number of bits to describe regularly positioned scalars (voxels) in 3D. Volume decompression reconstructs the data. Information that is meaningful to the user must be preserved.

The paper specifies novel algorithms, PR0, to compress and decompress volumes in a lossless way. The advantage of a lossless algorithm is it can be combined with lossless and lossy schemes to optimise performance, accuracy of data, or compression ratio. But to design an efficient solution that meets the goals is challenging. Compvox [3] uses seven passes, compared to one in PR0.

PR0 uses predictors [10] in raster-scan order on each scalar, slice by slice. It uses seven predictors to estimate values in a slice by way of referring to the

previously scanned values. At any time, only two slices must be kept in main memory. Dependent on which predictor triggers for a local data pattern (the first-order continuation along the 12-neighbourhood), PR0 appends the static Huffman code of the predictor and optionally writes the difference between the predicted and the correct value in a fixed number of bits. One more predictor can always match, because it does not refer to previously scanned data but uses a separately maintained dictionary of uncompressed scalar values. Decompression works in the same way. The challenge was to design predictors that would match often. A novel approach is to collect some predictors under one binary code and to define a mechanism that helps the decompressor to disambiguate between them. The paper specifies the mechanism.

Areas of application are medical and scientific data archiving, multimedia streaming, and 3D-texture compression on graphics cards.

Section 2 specifies related work in lossless volume compression. Section 3 specifies PR0. The Sections 4 and 5 give interpreted results from test runs. The concluding Section draws on future work.

## 2 Lossless Volume Compression

### 2.1 Rationale for the Development of Specialised Algorithms

Zip is the most widely used compression and decompression tool. Implementations of Zip follow the LZW algorithm [12][10]. LZW's advantages are availability, flexibility, speed, maturity, and efficacy. But for volume compression, LZW or any other general-purpose algorithm is sub-optimal because it must “rediscover” information that is intrinsic to the generic structure of volumes that are inspected by human beings. To use LZW to compress volumes uses more processor (CPU) cycles and memory than necessary.

Further, volumes are more and more frequently used, and to manage them in an economic way is important for the success of many areas of application. Examples are archives that host Terabytes of medical data on discs, Web-based components that transfer Gigabits of multimedia data over the Internet, and graphic cards that optimise memory use and bus traffic.

The paper discusses a promising approach toward real-time, lossless processing of volumes. We do not discuss lossy compression [8] to give the user control over the data quality degradation process, by way of selecting and composing data removal and information removal algorithms that complement each other. This means to separate quantisation from compression.

### 2.2 Specialised Lossless Algorithms

We survey prior work about volume compression before we specify our new approach in Section 3. Related applications have been discussed [6].

Fowler and Yagel [3] specify the Compvox algorithm that traces runs of voxels in 3D and uses DPCM to achieve compression ratios of 2:1 in the presence of noise in the input data. Compvox uses one predictor that scans orthogonal neighbours

of a voxel to estimate it and uses a combination of delta and Huffman encoding to store estimated values. The algorithm determines the weighting co-efficients to prepare the calculations in seven passes, each time over all input data.

Researchers have experimented with techniques that are known from 2D image processing [4]. They achieve compression ratios of 3:1 by means of combining background suppression and intra-slice prediction. The authors report weaker results from the application of, surprisingly, inter-slice prediction, multi-resolution pyramids, symmetry coding for which they match two halves of a volume to eliminate redundant data, and model-based coding for which they match a volume with a reference volume and encode only the difference between them. The compression ratio is only representative for volumes of MRI brain scans, the authors neither report results from compressing other volume types nor execution times.

A conceptually simple method for the lossless compression of volumes has been proposed [7]. The input data are decomposed in subbands and block-sorted. An arithmetic encoder post-processes data. The advantage of the approach is that the algorithm supports progressive decoding. Part of the method is equivalent to background suppression [4] because it classifies the input data in relevant and irrelevant regions. The reported reduction of 3:1 is verified for one dataset.

Steingötter and colleagues [11] compare how run-length-encoding (RLE), Huffman encoding, and JPEG, for which each is enhanced to find inter-slice correlation, compress a medical scan of the human thorax. They claim that compression ratios of 3:1 can be achieved but they do not report details.

Other systems use Wavelets, most predominant are 3D-SPIHT and its successor, AT-SPIHT [1]. SPIHT means to partition input data in a hierarchical tree to sort, compress, and decompress data in a lossless way. SPIHT stores the Wavelet-coefficients that are created during the computation in bit-plane order, orders them by magnitude, and extracts the least significant bits. For each category of importance to the 3D image, the data are written to the output stream. Enhancements to improve the efficiency of the algorithm are regularly reported, in this case AT-SPIHT. It unbalances the Wavelet-coefficient-tree to achieve better compression ratios than 3D-SPIHT. The reported reduction of 25% of the original size of a volume is the best result which has been reported. Another advantage is that SPIHT supports progressive decoding, as every Wavelet-based system. The disadvantages are that SPIHT uses more than one pass to remove data, and that the algorithm is complex, which restricts its implementation in hardware. Practically relevant is SPIHT is patented.

## 3 The PR0-Algorithm

### 3.1 Overview

The idea is to separate the topology from the topography of the volume and compress each by way of using a different algorithm. Further, the algorithm must work for any kind of volumes in an effective way where large connected regions of a volume have voxels of similar values. We can then define that the

connectivity is the topology, and that the topography is the represented set of scalar voxel values.

PR0 splits in one pass a volume in a dictionary of uncompressed values and a sequence of selected predictor codes. They interleave, if necessary, with DPCM bits to make predictor matching robust against noise in the least significant bits.

### 3.2 Design

We have selected the following 8 predictors in empirical test runs to compress and decompress a volume to support training (`bonsai`).  $q$  is a queue that contains uncompressed values,  $p$  is a predictor, and  $v$  is a voxel value at a 3D-position.

$$p_1(x, y, z) = (v(x - 1, y, z - 1) + v(x + 1, y, z - 1))/2 \quad (1)$$

$$p_2(x, y, z) = (v(x, y - 1, z - 1) + v(x, y + 1, z - 1))/2 \quad (2)$$

$$p_3(x, y, z) = (v(x - 1, y - 1, z - 1) + v(x + 1, y + 1, z - 1))/2 \quad (3)$$

$$p_4(x, y, z) = (v(x - 1, y + 1, z - 1) + v(x + 1, y - 1, z - 1))/2 \quad (4)$$

$$p_5(x, y, z) = (v(x - 1, y, z) + v(x, y - 1, z))/2 \quad (5)$$

$$p_6(x, y, z) = v(x, y, z - 1) \quad (6)$$

$$p_7(x, y, z) = (v(x - 1, y, z) + v(x, y - 1, z) + v(x - 1, y - 1, z))/3 \quad (7)$$

$$p_8(x, y, z) = \text{fifopop}(q) \quad (8)$$

The algorithm probes for each voxel all predictors in sequence if they reconstruct the original value properly. It encodes the first one that matches and optionally modifies it to store DPCM bits. The C-style pseudo code for the compressor is ( $t$  is a heuristic function which is explained below)

```
for all v in input data, s for output stream, q for queue
  for all i in p_i from predictors and t_i from triggers
    if | p_i(x, y, z) - v(x, y, z) | <= epsilon and t_i(x, y, z) == 1
      then fifopush(s, c) where c is the partial code of predictor id i
      if i == 8 then fifopush(q, v(x, y, z))
      delta = (p_i(x, y, z) - v(x, y, z))
      fifopush(s, tocode(delta))
```

The pseudo code for the decompressor is similar:

```
read queue q
for all c in input stream s, v for output data
  for all i's in the set C that is associated with the partial code c,
    p_i are predictors, t_i are triggers
    if i == 8 then v(x, y, z) = fifopop(q)
    else if t_i(x, y, z) == 1 then v(x, y, z) = p_i(x, y, z)
    if c contains modulation code delta then
      v(x, y, z) += fromcode(delta)
```

To use  $t$  reduces the number of bits to specify which predictor was selected, by way of using implicit information in the input data. We call the principle,

, which works in detail as follows: some predictors (in PR0, the first six) share the same predictor code and the decompressor must disambiguate between the sub-predictors if it encounters a code that refers to one of them. The compressor and decompressor use, in this case, the heuristic  $t$  to resolve the conflict by means of indicating which predictor would be the most suitable for the current data pattern. In PR0, we assign each of the first five predictors a trigger function to decide which of the sub-predictors matched ( $\epsilon$  is an arbitrary but fixed threshold that bounds the maximum error).

$$t_1(x, y, z) = (\epsilon \geq |v(x - 1, y, z - 1) - v(x + 1, y, z - 1)|) \quad (9)$$

$$t_2(x, y, z) = (\epsilon \geq |v(x, y - 1, z - 1) - v(x, y + 1, z - 1)|) \quad (10)$$

$$t_3(x, y, z) = (\epsilon \geq |v(x - 1, y - 1, z - 1) - v(x + 1, y + 1, z - 1)|) \quad (11)$$

$$t_4(x, y, z) = (\epsilon \geq |v(x - 1, y + 1, z - 1) - v(x + 1, y - 1, z - 1)|) \quad (12)$$

$$t_5(x, y, z) = (\epsilon \geq |v(x - 1, y, z) - v(x, y - 1, z)|) \quad (13)$$

The seventh and the eighth predictor are explicitly encoded in the compressed data and therefore, have no trigger functions associated with them.

The eighth predictor is always applicable because it stores a complete value in the queue  $q$  for decompression. The number of entries in the queue matches the number of times the default predictor was encoded.

### 3.3 Code Format

Because we use trigger functions, we approximately halve the number of bits to identify the predictors and add a 1-bit DPCM-code. This uses at most 3 bits per predicted value. The compressor attaches more DPCM-bits when it processes volumes that comprise more than 8 bits per voxel because then the noise spreads over more than one bit. We have achieved a good compression ratio with 6 bits to encode the complete modulation. Table 1 specifies the codes to identify selected predictors to estimate 8-bit data types (six predictors share a code).

**Table 1.** Huffman codes (with prefix-condition)

Predictor id $i$	Perfect match	Correction: -1	Correction: +1
1..6	01	001	000
7	110	101	100
8	111	—	—

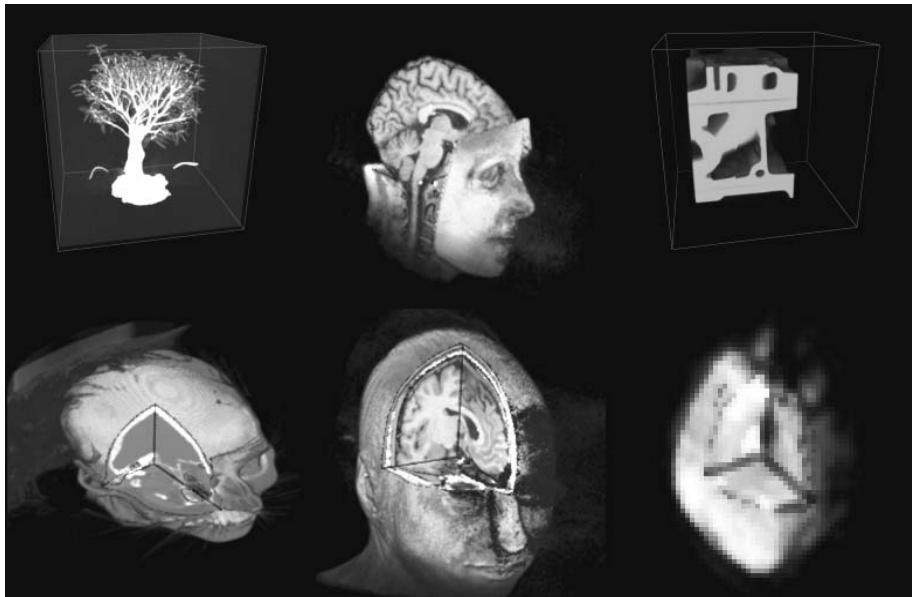
## 4 Evaluation

### 4.1 Setup

The first author implemented PR0 in C and compiled the parts PREDICT\_INCORE and PREDICT\_FAST. PREDICT\_SHORTDICT in combination with the documented

**Table 2.** Samples (see Fig 1 from left to right, top to bottom), size is given in voxels

Name	X-size	Y-size	Z-size	Bits per voxel	Number of volumes
bonsai	256	256	128	8	1
brain	256	256	109	16	1
engine	256	256	128	8	1
head	256	256	113	16	1
crt-2	256	256	192	16	1
funcbrain	64	64	32	16	900

**Fig. 1.** The volumes for the evaluation, visualized by OpenQVis and MRIcro

typedefs enabled support for 16-bit voxels. The executable was optimised for speed on a Pentium 4. The source code did not contain optimisations. The author used MS Windows XP and Microsoft C++ Version 13.10.3077. His workstation was a Dell Precision 340 with 37 GB harddisc, 512 MB main memory, and a Pentium 4 at a clock rate of 2.4 GHz.

The first author took samples from [2][9] and converted them to Little-Endian format and merged them to one file each. The fourth author provided a 24.5 MB CRT scan (crt-2) and a sequence of volumes (funcbrain) that describe in 225 MB functional measurements of brain activity over time. Table 2 specifies the datasets.

## 4.2 Performance

The first author compressed each file once in mode  $-1$  (Worst-Zip) and in mode  $-9$  (Best-Zip) of Info-Zip version 2.3 under Cygwin 1.3, on top of the XP installation. He used Cygwin for the evaluation of Zip's performance because it was already available in the distribution and usable with negligible overheads. The author used PR0 for each file and compressed the .pr0 files with Info-Zip once in mode  $-1$  and once in mode  $-9$  (the latter was omitted in the Tables because it is two to three times slower and saves just a few KB). He measured how long each action took for each dataset. The complete process was repeated twice, except for `funcbrain`, and the results averaged. To retrieve fine measurements, the author enabled `PREDICT_BENCHMARK` in the PR0 implementation to use `GetTickCount()` of the MS Windows API to calculate and print the average time for processing each slice of the crt-2 file. The fine measurements were done in a separate test run. We then compiled the results.

## 4.3 Results

PR0 reported less than 8 milliseconds to process  $256^2 \cdot 16$  bit. To compress crt-2 took 896 ms. Table 3 gives the compression in the percentage of the size of the original uncompressed file. Table 4 specifies execution time. (The execution times to process `funcbrain` were rounded to seconds.)

**Table 3.** Comparison by the size of the output data, given by the percentage of the size of the input data in bytes

Name	PR0	Worst-Zip	Best-Zip	Worst-Zip, PR0 pre-processes
bonsai	58	50	47	50
brain	58	57	55	53
engine	46	40	38	34
head	52	51	50	44
crt-2	56	49	48	45
average	54	49	48	45
funcbrain	59	49	46	46

**Table 4.** Comparison by the execution time in sec, for compression / decompression

Name	PR0	Worst-Zip	Best-Zip	Worst-Zip, PR0 pre-processes
bonsai	1.8 / 1.5	0.3 / 1.0	3.7 / 1.0	0.7 / 0.3
brain	1.7 / 2.0	2.3 / 1.0	5.0 / 1.0	1.7 / 1.3
engine	1.5 / 1.2	1.0 / 0.7	3.7 / 1.0	1.0 / < 0.1
head	1.7 / 2.0	2.3 / 0.7	4.0 / 1.0	1.3 / 1.0
crt-2	2.9 / 3.7	2.0 / 3.7	19.7 / 2.3	4.0 / 1.7
average	1.9 / 2.1	1.6 / 1.4	7.2 / 1.3	1.7 / 0.9
funcbrain	49.0 / 43.0	39.0 / 53.0	215.0 / 42.0	30.0 / 22.0

## 5 Discussion

### 5.1 Interpretation of Results

Tables 3 and 4 specify that PR0 performs well in comparison with the standard tool. PR0 gives nearly the same compression ratio as Zip. In both PR0 and Zip, the decompression time is proportional to the target file size, therefore the better the compression, the faster the decompression is. This is shown by Best-Zip which generates an excellent compression ratio in general, takes long to compress, and takes short time to decompress. Best-Zip's poor compression speed for the *crt-2* and *funcbrain* datasets is probably due to a non-optimal access scheme encountered in the dictionary management and the inability to recognise the 16-bit format of the voxel stream and the sequences of slices of static extent, which PR0 handles by definition in an optimal way.

To zip PR0's output reduces it further. The Tables 3 and 4 show it. We can produce the most remarkable reduction rate with the *engine* dataset, because it describes a simple structure. Generally, a combination of Worst-Zip and PR0 generates better compression rates than Best-Zip alone, with the exception of the *bonsai* volume (3% loss of compression) and the *funcbrain* sequence of volumes (no improvement of compression but drastic reduction of execution time). A better compression for larger volumes indicates that the latter is caused by the implementation of PR0 which does not compress volume boundaries. A modified implementation will avoid the problem (see Section 5.3).

Without closer inspection, one could assume that PR0 gives nice but not dramatically better results. A combination of PR0 and Worst-Zip could not replace Best-Zip because the sum of the execution times of PR0 and Worst-Zip is greater than those for Worst-Zip and not substantially better than Best-Zip for some cases. And for some applications, the loss of decompression performance could be a disadvantage. But none of these arguments is backed up by the measurements. The times to execute the algorithm, which are shown in the Tables, include measurements for the disc access routines, which dominate compression and decompression as one sees from comparing the table entry for PR0 and *ct-2* with the detailed measurement: compression takes 0.9 sec and read-write file access more than twice the time, that means read, (de-)compress, and write take equally long. Two disc accesses can be removed when combining PR0 and Zip by way of forwarding intermediary data streams in main memory. Hence, a combination of PR0 and Zip can minimise the file size ... execution time. Our data indicate that this effect will amplify in favour of PR0 when the size of the volumes increases. Each discussed improvement is small but accumulates to substantial savings when it is applied to support batch- or streaming-applications.

In summary, the evaluation of PR0 has indicated advantages of our algorithm:

1. Zip almost consistently compresses data better when they are pre-processed with PR0.
2. To combine Zip at compression mode  $-1$  (weakest) with PR0 takes substantially less time as Zip at compression mode  $-9$  (strongest). This is most evident for the examples *crt-2* and *funcbrain*.

The disadvantage is that the current implementation of PR0 is less efficient than Worst-Zip, which surprises us because PR0 should theoretically be faster due to the simpler management of the data dictionary. Hence, we believe an optimal implementation of PR0 would at least match Worst-Zip’s runtime behaviour.

## 5.2 Comparison with Prior Work

PR0 has a fair compression ratio for realistic data, is small (about 300 lines of code), uses simple predictors that are selected during one compression run, and is numerically robust because only few fundamental operations are necessary to compress and decompress data. Because the algorithms are lossless, they do not generate hidden errors by accident.

The algorithms can be parallelised because the referenced dataset is local and it exploits the multi-dimensional data coherence and constant constraints, for example the size of the input data slices.

Last, one achieves fine-grained balancing between cost/profit parameters by way of replacing or removing predictors.

Code compactness, greater potential for parallelisation schemes, and simplicity as well as customisability are factors that have not been addressed elsewhere to our knowledge. Instead, reports of similar approaches often focus on the data reduction as the single criterion. However, more factors are relevant [5][10].

In specific comparisons, Fowler and Yagel’s [3] implementation compresses approximately equally to a combination of PR0 and Zip and is three times slower than Zip. Different to [4] which achieve better average compression ratios, PR0 is useful for a larger class of volumes, not just medical datasets. The approach by [7] achieves a better reduction of data than PR0, but as the authors admit, has weak compression performance. Finally, AT-SPIHT [1] does not meet fixed real-time constraints but PR0 does.

## 5.3 Limitations

- (i) We did not specify the entropy of the volumes.
- (ii) PR0’s implementation does not compress boundaries. Predictors could read constants to compensate nil values.
- (iii) DPCM bits are uncompressed. They could be Huffman-encoded.

## 6 Conclusions

We have specified novel predictor-based compression and decompression algorithms. The design follows recommendations to balance between compression ratio and speed [10]. As demonstrated, the algorithms combine well with other codecs. The source code and data material are at [www.l3s.de/~jensen/lossless/](http://www.l3s.de/~jensen/lossless/).

Researchers should investigate the quality of the selected predictors in a theoretical analysis and improve them. Developers can combine PR0 and other codecs, evaluate the benefit of adaptively selecting predictors, and optimise the reference implementation. We speculate that an FPGA-implementation of PR0 will process animation sequences of volumes in real-time.

## Acknowledgments

We thank S. Olbrich for comments, and the DFG for funding the project EVITA.

## References

1. Cho, S., Kim, D., Pearlman, W.A.: Lossless Compression of Volumetric Medical Images with Improved Three-Dimensional SPIHT Algorithm. In: Honeyman-Buck, J.C. (ed.): *Journal of Digital Imaging*. Vol. 17. **1** (2004) 57–63
2. Engel, K.: Pre-integrated Volume Rendering. At: <http://wwwvis.informatik.uni-stuttgart.de/~engel/{Bonsai.zip, Engine.zip}>. (2001) Accessed: 28-Jan-05
3. Fowler, J.E., Yagel, R.: Lossless Compression of Volume Data. In: Kaufman, A., Krüger, W. (eds.): *Proceedings of the 1994 Symposium on Volume Visualization*. ACM Press, New York (1994) 43–50
4. Hargreaves, B., Johanson, B., Nayak, K.: Lossless Compression of 3D MRI Brain Images. At: <http://ise.stanford.edu/class/ee392c/demos/hargreaves-johanson-nayak/>. (1997) Accessed: 2-Dec-04
5. Herrero, I., Salmeron, J.L.: Using the DEA Methodology to Rank Software Technical Efficiency. In: Crawford, D. (ed.): *Comm. of the ACM*. Vol. 48. **1** (2005) 101–105
6. Jones, M.: Distance Field Compression. In: Skala, V. (ed.): *Journal of WSCG*. Vol. 1–3. (2004) 199–204
7. Klappenecker, A., May, F., Beth, Th.: Lossless Compression of 3D MRI and CT Data. In: Laine, A.F., Unser, M.A., Aldroubi, A.(eds.): *Proceedings of SPIE on Wavelet Applications in Signal and Imaging Processing VI*. (1998) 140–149
8. Krishnan, K., Marcellin, M., Bilgin, A., Nadar, M.: Compression / Decompression Strategies for Large Volume Medical Imagery. In: Ratib, O.M., Huang, H.K. (eds.): *Proceedings of SPIE Medical Imaging 2004: PACS and Imaging Informatics*. Vol. 5371. (2004) 152–159
9. Levoy, M.: The Stanford Volume Data Archive. At: <http://graphics.stanford.edu/-data/voldata/{CThead.tar.gz, MRbrain.tar.gz}>. (2004) Accessed: 28-Jan-05
10. Salomon, D.: *Data Compression – The Complete Reference*. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (2004)
11. Steingötter, A., Werner, C., Sachse, F., Dössel, O.: Kompression drei- und vierdimensionaler medizinischer Bilddaten. In: *Biomedizinische Technik*. Vol. 43. (1998) 478–479
12. Welch, T.A.: A Technique for High-Performance Data Compression. In: You, S.S. (ed.): *IEEE Computer*. Vol. 17. **6** (1984) 8–19

# Lossless Compression of Map Contours by Context Tree Modeling of Chain Codes

Alexander Akimov, Alexander Kolesnikov, and Pasi Fränti

Department of Computer Science,  
University of Joensuu,  
P.O. Box 111, 80110 Joensuu, Finland  
{akimov, koles, franti}@cs.joensuu.fi

**Abstract.** We consider lossless compression of digital contours in map images. The problem is attacked by the use of context-based statistical modeling and entropy coding of chain codes. We propose to generate an optimal context tree by first constructing a complete tree up to a predefined depth, and then create the optimal tree by pruning out nodes that do not provide improvement in compression. Experiments show that the proposed method gives lower bit rates than the existing methods for the set of test images.

## 1 Introduction

Digital maps are usually stored as vector graphics in a database for retrieving the data using spatial location as the search key. The visual outlook of maps representing the same region varies depending on the type of the map (topographic or road map), and on the desired scale (local or regional map). Vector representation is convenient for zooming as the maps can be displayed in any resolution defined by the user. The maps can be converted to raster images for data transmission, distribution via internet, or because of incompatibility of the vector representations of different systems.

In order to increase the efficiency of raster map compression, we consider the variant, when some object, instead of being rasterized, will be described by *chain codes* and compressed separately from rest of map data. This can lead to a more efficient representation of the map and, consequently, to improve of compression. Chain coding is a common approach for representing different rasterized shapes such as line-drawings, planar curves and contours. We consider thin digital curves of one pixel width, extracted from the vector data before rasterization.

The previous works consider different schemes of encoding and chain code representation [3], [10], [11]. For example, the method in [12] uses second order context model based on 8 directional chain codes. Further development of the context-based compression of chain codes was presented in [4]. The authors have improved the performance of the chain codes encoding by increasing the size of finite context models. The problem of encoding of chain codes by *prediction by partial matching* (PPM) algorithm [2] has been considered in [1].

In principle, context based compression can be improved by using a larger number of neighboring symbols in the context. But the increase of the context size leads to the

problem of context dilution, in which the statistics are distributed over too many contexts, and thus, affects the accuracy of the probability estimates.

*Context tree* provides a more flexible approach for modeling the contexts so that a larger number of neighbor pixels can be taken into account without the context dilution problem [13]. The context tree algorithm was originally introduced in [16], and analyzed in [10]. Practical solutions for the context tree based compression algorithms for grey-scale and bi-level images have been described at [19] and [13] respectively.

In this paper, we use the context tree approach for encoding the chain codes. We provide algorithm for optimal context tree construction. We compare the compression performance of the rasterized map contours when encoded by JBIG [9], and by the optimal context tree chain codes, representing the same contours. The results show that the proposed method provides 25% lower bit rate than JBIG, and is 40% faster because only the contour pixels need to be processed.

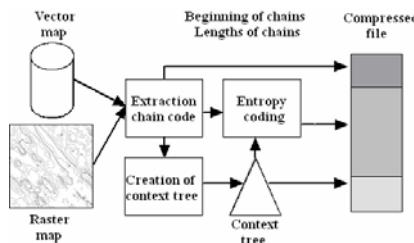
The overall scheme of the proposed compression method is as follows:

Step 1: Extract contours from the vector or raster map and convert them into chain codes. Store the start points and the lengths of the chains.

Step 2: Create and store the optimal context tree for the chain codes.

Step 3: By using of context tree modeling and any entropy coding, encode the chain codes.

This scheme is shown in Figure 1. For simplicity, we store the size and the beginning of each chain (BOC) as such without any further compression.



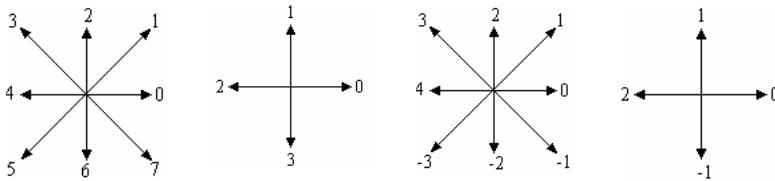
**Fig. 1.** Overall system diagram of the proposed method

## 2 Chain Code Representation

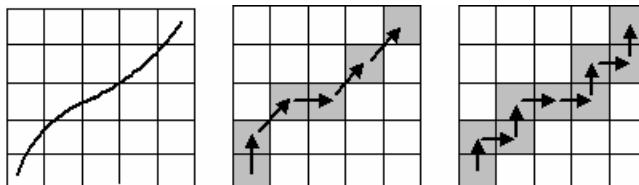
Freeman [5], [6] proposed chain coding of digital contours drawings and description. The chain codes represent the digital contour by a sequence of line segments of specified length and direction, see Figure 2. We consider both 8- and 4-directional chain coding schemes; see Figure 3.

The chain code representation is constructed as follows.

Step 1: Select a starting point of the contour. Represent this point by its absolute coordinates in the image.



**Fig. 2.** 8-connected and 4-connected chain codes and their differential chain codes



**Fig. 3.** An example of chain code construction: original curve (left), 8-directional (center) and 4-directional (right)

Step 2: Represent every consecutive point by a chain code showing the transition needed to go from the current point to the next point on the contour.

Step 3: Stop if the next point is the initial point, or the end of the contour. Store the lengths of the contours into the file.

An alternative way for chain code representation are differential chain codes [5]. Each differential chain code  $k_i$  is representing by the difference of the current chain code  $c_i$  and the preceding chain code  $c_{i-1}$ :  $k_i = c_i - c_{i-1}$ .

The chain codes of contours can be extracted from the map image in two ways. Firstly, if the map is obtained directly from the map database, we can extract contours directly from the vector data. On the other hand, if the map is provided as a color raster image, we can use *color separation* and following by *vectorization* (must be used to extract the contours).

### 3 Context Tree Modeling

#### 3.1 Finite Context Modeling

We compress the chain codes sequentially according their order in the input data. Consider the current symbol  $x_i$  and the string of the  $M$  previous symbols  $x_{i-1}, \dots, x_{i-M-1}$  denoted as  $x^{i-1}$ . In the *context based modeling* the probability of the next symbol  $x_i$  is conditioned on its *context*  $x^{i-1}$ . The probabilities of the symbols generated in a given context, are treated as independent [17]. Thus, a model becomes a collection of independent sources of random variables. By the assumption of independence, it is easy to assign probabilities to each new symbol generated at the current context. Let us denote the cardinality of the alphabet of the encoded data as  $N$ . If

$n_1(x^{i-1}), \dots, n_N(x^{i-1})$  are the counts of all symbols generated at the given context  $x^{i-1}$ , then the conditional probability of the event  $x_i = k$ ,  $k \in [1, \dots, N]$  is:

$$p(x_i = k | x^{i-1}) = \frac{n_k(x^{i-1})}{\sum_{j=1}^N n_j(x^{i-1})} \quad (1)$$

We consider the encoding of the given statistical model by entropy-based encoder. The probability for the entropy-based coder is estimated as:

$$p(x_i = k | x^{i-1}) = \frac{n_k(x^{i-1}) + \delta}{\sum_{j=1}^N n_j(x^{i-1}) + N \cdot \delta} \quad (2)$$

The parameter  $\delta$  here depends on different arithmetic coders, but it usually equals to  $1/N$  [8], [14].

### 3.2 Context Algorithm Revisited

Context tree is applied for the compression in the same manner as the fixed size context; only the context selection is different. It is made by traversing the context tree from the root to a terminal node, each time selecting the branch according to the corresponding previous symbol value. If the corresponding symbol points to a non existing branch, or the current node is a leaf, then we came to a terminal node, which points to the statistical model that is to be used.

The context tree can be constructed beforehand (*static approach*) or optimized directly for the encoded data (*semi-adaptive approach*). In the second case, the tree structure must be stored in the compressed file. The process of optimal tree construction consists of two main phases: initialization of the context tree, and pruning of the tree.

### 3.3 Construction of Initial Context Tree

To construct an initial context tree, we process the image to collect statistics for all potential contexts, leaves and internal nodes. Each node stores information of  $N$  counts for all symbols generated at the current context. The algorithm of the context tree construction is:

Step 1: Create a root of the tree.

Step2: For all  $i = 1$  to  $n$ , traverse the tree along the path defined by the past string  $x^{i-1}$ . If some indices of the symbols in  $x^{i-1}$  are less than one, then set these symbols to zero. If some node, visited according the correspondent symbol of the string  $x^{i-1}$ , does not have a consequent branch (for transition to the next symbol of  $x^{i-1}$ ), then create the necessary child node and process it. Each new node has  $N$  counts, which are initially set to zero. In all visited nodes, increase the count of  $x_i$  by 1.

This completes the construction of the context tree for all possible contexts. The time complexity of this algorithm is  $O(n)$ .

### 3.4 Construction of Optimal Context Tree

The initial context tree needs to be pruned by comparing the parent node and its children nodes for finding the optimal combination of siblings. Let us denote by  $c(T)$  the number of bits, required to store the tree structure in the compressed file. For different strategies of the tree construction it will be different:

$$c(T) = \begin{cases} 0, & \text{static approach} \\ K, & \text{semiadaptive approach, complete tree} \\ N \cdot K, & \text{semiadaptive approach, incomplete tree,} \end{cases} \quad (3)$$

where  $K$  is the cardinality of the tree  $T$ . We will denote the set of all terminal nodes as  $S(T)$ . Let us denote as  $n_i(s)$ ,  $s \in S(T)$ , the count of the symbol  $i$ , encoded by the statistical model, pointed by the terminal node  $s$ . By the cost of a terminal node  $s$  here we understand the following expression [7], [13]:

$$\tilde{c}(n_1(s), n_2(s), \dots, n_N(s)) = \begin{cases} 0, & \text{if } n_1(s) = n_2(s) = \dots = n_N(s) = 0 \\ -\log_2 \frac{\prod_{i=1}^N \prod_{j=0}^{n_i(s)-1} (j+\delta)}{\prod_{j=0}^{n_0(s)+n_1(s)+\dots+n_N(s)-1} (j+N \cdot \delta)}, & \text{otherwise.} \end{cases} \quad (4)$$

This definition corresponds algorithmically to the use of a one pass arithmetic coding without the update of the statistical model [8]. By the cost of the context tree  $T$ , we will denote the following expression:

$$L(T) = c(T) + \sum_{s \in S(T)} \tilde{c}(n_1(s), n_2(s), \dots, n_N(s)) \quad (5)$$

The problem of the tree pruning is to modify the structure of the full context tree so that the expression (5) will be minimized. For solving this problem, we use a bottom-up algorithm [15]. The main principle of this algorithm is that the optimal tree consists of optimal sub-trees.

For any node  $t$  from the tree  $T$ , let us denote the vector of counts as  $\tilde{n}(t) = (n_1(t), n_2(t), \dots, n_N(t))$ , the child nodes as  $t_i$ , and the node configuration vector as  $v = (v_1, \dots, v_N)$ ,  $v_i \in \{0,1\}$ . The vector  $v$  defines which of the node branches will remain: if  $v_i = 0$ , then the  $i$ th branch will be deleted from the node. Then the principle of sub optimality for any given sub tree  $\hat{T}$ , starting from the given node  $t$  can be represented as follows: the optimal cost  $L_{opt}(\hat{T})$  for any given sub tree  $\hat{T} \subseteq T$  can be expressed by the following recursive equation:

$$L_{opt}(\hat{T}) = \begin{cases} 0, & \text{if } \hat{T} \text{ is null} \\ \tilde{c}(\tilde{n}(t)) + \alpha, & \text{if } \hat{T} \text{ has no children} \\ \min_v \{L_v(\hat{T}, v)\} & \text{otherwise,} \end{cases} \quad (6)$$

where

$$L_v(\hat{T}, v) = \tilde{c} \left( \tilde{n}(t) - v \circ \left( \sum_i \tilde{n}(t_i) \right) \right) + \sum_i (v_i \cdot L_{opt}(\hat{T}_i)) + \alpha \quad (7)$$

The tree  $\hat{T}_i \subset \hat{T}$  is a sub tree of  $\hat{T}$ , starting from its child node  $t_i$  and the constant  $\alpha$  is the amount of bits required for describing a single node.

In general, the cost calculation of an optimal context tree  $T$  can be described as follows:

Step 1: If  $T$  has no child nodes, then return the accumulated code length of its root according to (4).

Step 2: For all sub trees  $T_i \subset T$ , starting from the child nodes of  $T$  root, calculate their optimal costs  $L_{opt}(T_i)$ .

Step 3: According to the found  $L_{opt}(T_i)$ , the vectors of counts  $\tilde{n}(t)$ , and  $\tilde{n}(t_1), \dots, \tilde{n}(t_N)$ , find the optimal vector  $\tilde{v} = \arg \min_v L_v(T, v)$ .

Step 4: Prune out the children sub trees according the vector  $\tilde{v}$ .

Step 5: Return the value  $L_v(T, \tilde{v})$ .

The algorithm recursively prunes out all unnecessary sub trees, and finally gets the optimal structure of the context tree, see Figure 3.

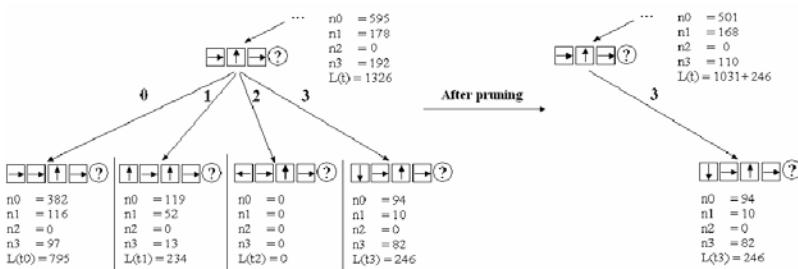


Fig. 3. An example of pruning of the context tree

## 4 Experiments

We provided two different series of experiments. The first one illustrates the efficiency of the optimal context tree encoding of the chain codes. The second one illustrates the ability of the independent chain encoding to increase the compression

performance of the map image compression in general. Images of Figure 4, which we use, are vector maps, rasterized with the resolution of  $5000 \times 5000$  pixels. The statistics for all images are shown in Table 1. The first three images are contours of geographical objects, and the last image is a collection of elevation lines. The absolute chain codes were transformed into differential chain codes before compression. As the entropy coder we used range-coder [14].

Tables 1 and 2 show the results for different depth of the context model in the case of 8-connected and 4-connected chain code representations. The numbers in Table 1 are the estimated bit rate according (6). The numbers in Table 2 are the real bit rate, resulted after the range coder. For comparing the compression efficiency, there are results of chain codes compression *PPMd* algorithm with the maximum context order 8 [17]. Higher context order in PPM leads us to the context dilution problem and, consequently, decreasing the compression performance. For the test images the efficient range of the context tree depth is from 4 to 10.



**Fig. 4.** The set of test images

**Table 1.** Test image properties and estimated bit rate (bits per symbol)

<b>4-connected chain codes</b>						
	Number of chain codes	Depth				
		4	6	8	10	12
Image #1	37888	0.782	0.721	0.711	0.706	0.706
Image #2	86216	1.031	0.999	0.994	0.992	0.992
Image #3	208320	1.488	1.482	1.481	1.481	1.481
Image #4	519316	0.673	0.595	0.568	0.553	0.545
Average	212935	0.994	0.949	0.939	0.933	0.931
<b>8-connected chain codes</b>						
	Number of chain codes	Depth				
		4	6	8	10	12
Image #1	27306	0.992	0.977	0.970	0.970	0.970
Image #2	64220	1.386	1.377	1.375	1.375	1.375
Image #3	160763	2.273	2.272	2.272	2.272	2.272
Image #4	356839	0.841	0.794	0.782	0.777	0.775
Average	152282	1.373	1.355	1.350	1.349	1.348

The second series of experiments were aimed to estimate the efficiency of chain codes compression in comparison to an efficient raster image compression algorithm, namely the JBIG. Table 3 summarizes compressed file sizes, when compressed by the optimal context tree algorithm (CTC 4 and CTC 8), and for corresponding raster

images compressed by JBIG. The raster map images are obtained by rasterization from 4-connected chain codes. The experiments show that the running time of the CTC algorithm is than that of the JBIG. This is because of much smaller amount of encoded information: JBIG encodes 2 500 000 pixels at each image, when CTC encodes only 500 000 chain codes.

Table 4 represents the structure of the compressed file: the percentage of all three types of the data in the file: beginning of the chains (BOC), structure of the context tree (CT), and the encoded chain codes (Chain Codes). The most used contexts in Image#4 for CTC 4 and for CTC 8 compression are shown in Tables 5 and 6 consequently. The most used contexts describe horizontal, vertical or diagonal straight lines. All the experiments were provided on computer P3 500MHz, 256 Mb RAM, Windows NT.

**Table 2.** Real bit rate (bits per symbol)

<b>4-connected chain codes</b>							
Depth	CTC 4						
	8	4	6	8	10	12	14
Image #1	0.725	0.801	0.746	0.744	0.742	0.742	0.742
Image #2	1.032	1.044	1.017	1.012	1.010	1.010	1.010
Image #3	1.561	1.500	1.494	1.493	1.493	1.493	1.493
Image #4	0.578	0.678	0.601	0.577	0.565	0.560	0.559
Average	0.974	1.006	0.965	0.957	0.953	0.951	0.951
<b>8-connected chain codes</b>							
Depth	CTC 8						
	8	4	6	8	10	12	14
Image #1	0.994	1.020	1.022	1.015	1.015	1.015	1.015
Image #2	1.447	1.404	1.397	1.393	1.393	1.393	1.393
Image #3	2.381	2.288	2.289	2.289	2.289	2.289	2.289
Image #4	0.794	0.850	0.808	0.804	0.803	0.803	0.803
Average	1.404	1.391	1.379	1.375	1.375	1.375	1.375

**Table 3.** Comparison of the CTC-encoded chains and JBIG encoded raster images

	Compressed file size (bytes)			Compression time (sec)		
	JBIG	CTC 4	CTC 8	JBIG	CTC 4	CTC 8
Image #1	5719	3518	3470	99	3	14
Image #2	16027	10887	11189	100	9	50
Image #3	41789	39661	46774	104	25	97
Image #4	71128	39300	38850	100	31	83
Average	33666	23342	25071	101	17	61

**Table 4.** The proportions of different parts in the compressed file

	BOC	CT	Chain codes
Image #1	0.2%	1.5%	98.3%
Image #2	0.1%	0.6%	99.3%
Image #3	1.8%	0.3%	97.9%
Image #4	7.9%	1.0%	91.1%

**Table 5.** Three most used context for Image #4 in CTC 4

Context	n0	n1	n2	n3	total
	10577	0	2	1943	12522
	4	1355	7924	0	9283
	5904	0	8	1910	7822

**Table 6.** Three most used context for Image #4 in CTC 8

Context	n0	n1	n2	n3	n4	n5	n6	n7	total
	2	0	1	0	3	560	6473	854	7893
	0	0	6	681	4883	477	8	0	6055
	45	2	2	0	0	0	4879	994	5922

## 5 Conclusions

We have proposed context tree algorithm for encoding chain codes of contours in map images. The proposed algorithm increased the compression performance over the PPM algorithm by 2-3%. The use of chain codes, instead of the compression of rasterized contours, improves the compression by 25%, on average. The results could be improved up to the theoretical limits by using a more suitable entropy encoder, instead of sub-optimal range coder.

## References

- [1] Bossen, F., Ebrahimi, T.: Region shape coding, *Technical Report M0318*, ISO/IEC JTC1/SC29/WG11, November 1995
- [2] Cleary, J., Witten, I.: Data compression using adaptive coding and partial string matching, *IEEE Trans. on Communications*, 32(4), April 1984, 396-402
- [3] Eden, M., Kocher, M.: On performance of a contour coding algorithm in the context of image Coding Part 1: Contour Segment Coding, *Signal Processing*, 1985, 8, 381-386
- [4] Estes, R., Algazi, R.: Efficient error free encoding of binary documents, In: *Proc. of IEEE Data Compression Conference*, March 1995, 122-131
- [5] Freeman, H.: Computer processing of line drawing images, *ACM Computing Surveys*, 6, March 1974, 57-59
- [6] Freeman, H.: Application of the generalized chain coding scheme to map data processing, In: *Proc. of IEEE Pattern Recognition and Image Processing*, May 1978, 220-226
- [7] Helfgott, H., Cohn, M.: Linear-time construction of optimal context trees, In: *Proc. of the IEEE Data Compression Conference*, April 1998, 369-377
- [8] Howard, P., Vitter, J.: Analyses of arithmetic coding for data compression, In: *Proc. of the IEEE Data Compression Conference*, 1991, 3-12
- [9]JBIG: Progressive bi-level image compression, *ISO/IEC International Standard 11544*, 1993
- [10] Kaneko, T., Okudara, M.: Encoding of arbitrary curves based on chain code representation, *IEEE Trans. on Communications*, July 1985, 33, 697-707
- [11] Liu, Y.K., Zalik, B.: An efficient chain code with Huffman coding, *Pattern Recognition*, 38(4), 2005, 553-557

- [12] Lu, C.C., Dunham, G.: Highly efficient coding schemes for contour lines based on chain code representations, *IEEE Trans. on Communications*, 39(10), October 1991, 1511-1514
- [13] Martins, B., Forchhammer, S.: Tree coding of bi-level images, *IEEE Trans. on Image Processing*, 7(4), April 1998, 517-528
- [14] Martin, G.: An algorithm for removing redundancy from a digitized message, Presented at: *Video and Data Recording Conference*, July 1979
- [15] Norhe, R.: Topics in descriptive complexity, *PhD Thesis*, University of Lingkoping, Sweden, 1994
- [16] Rissanen, J.: A universal data compression system, *IEEE Transactions on Information Theory*, 29(5), September 1983, 656-664
- [17] Shkarin, D.: PPM: one step to practicality, In: *Proc. of the IEEE Data Compression Conference*, April 2002, 202-211
- [18] Weinberger, M., Rissanen J.: A universal finite memory source, *IEEE Trans on Information Theory*, 41(3), May 1995, 643-652
- [19] Weinberger, M., Rissanen, J., Arps, R.: Application of universal context modeling to lossless compression of grey-scale images, *IEEE Transactions on Image Processing*, 5, April 1996, 575-586

# Optimal Estimation of Homogeneous Vectors

Matthias Mühlich<sup>1,2</sup> and Rudolf Mester<sup>1</sup>

<sup>1</sup> Computer Vision Group, Goethe University, 60054 Frankfurt, Germany

{muehlich, mester}@iap.uni-frankfurt.de

<sup>2</sup> Lehrstuhl für Bildverarbeitung, RWTH Aachen,  
Sommerfeldstr. 24, 52074 Aachen, Germany

**Abstract.** Estimation of *inhomogeneous* vectors is well-studied in estimation theory. For instance, given covariance matrices of input data allow to compute optimal estimates and characterize their certainty. But a similar statement does not hold for *homogeneous* vectors and unfortunately, the majority of estimation problems arising in computer vision refers to such homogeneous vectors...

The aim of this paper is twofold: First, we will describe several iterative estimation schemes for homogeneous estimation problems in a unified framework, thus presenting the missing link between those apparently different approaches. And secondly, we will present a novel approach called IETLS (for iterative equilibrated total least squares) which is insensitive to data preprocessing and shows better stability in presence of higher noise levels where other schemes often fail to converge.

## 1 Introduction: The Homogeneous Estimation Problem

Parameter estimation problems of the general form

$$\varphi(\bar{\mathbf{x}}_i, \mathbf{p}) = 0 \quad \forall i = 1, \dots, m \quad (1)$$

are ubiquitous in computer vision. Here  $\mathbf{p}$  stands for the parameter vector that has to be estimated and  $\bar{\mathbf{x}}_i \in \mathbb{R}^\ell$  denotes some true but unknown vectors (bar accent for true values; index  $i = 1, \dots, m$  for different measurements), of which only some error-prone versions

$$\mathbf{x}_i = \bar{\mathbf{x}}_i + \mathbf{e}_i \quad (2)$$

(additive noise model) are available. For instance,  $\mathbf{x}_i \in \mathbb{R}^4$  could be the stacked coordinates of corresponding points in a stereo image. When  $\bar{\mathbf{x}}_i$  is replaced by  $\mathbf{x}_i$  in (1), we only achieve approximate equality:  $\varphi(\mathbf{x}_i, \mathbf{p}) \approx 0$ . In statistical literature, this approach is known as errors-in-variables (EIV) model; equations (1) and (2) define  $\dots$  and  $\dots$ , respectively. The estimation problem can now be rephrased as obtaining statistically optimal estimates when some statistical information on the errors is given.

Projection from the 3D world to one or several 2D image(s) cancels out the scale factor—this is the underlying reason why the majority of estimation problems in computer vision refer to  $\dots$ , i.e. we can only estimate

the sought parameter vector  $\mathbf{p}$ . By normalizing homogeneous vectors to unit vectors, these estimation problems can be linked to distributions of vectors with undefined sign; distributions of vectors with undefined sign are also known as [1].

[1]. Homogeneous estimation therefore means that we are estimating on a unit hyperspheres and iterative schemes which are motivated from real space (e.g. variational approaches) do not consider the curved and ambiguous nature of our estimation space. For instance, classical point data covariance matrices are (at most) defined in a tangent hyperplane (here: additionally after declaring one hemisphere as ‘valid’). Obviously, this can only be meaningful if the axial distribution is highly concentrated around a preferred axis. For higher error levels, concepts from real space cannot be transferred easily; this is the root of all problems in homogeneous estimation...

All homogeneous data model constraints (1) can be represented as orthogonality constraints  $\mathbf{a}_i^T \mathbf{p} \approx 0$ . Stacking different measurements on top of each other then results in a matrix equation  $\mathbf{A}\mathbf{p} \approx \mathbf{0}$ . Such Estimation problems are known as (homogeneous) problems [2]. If we additionally allow ancillary constraints  $\psi_j(\mathbf{p}) = 0$  (e.g. a single constraint which enforces zero determinant for fundamental matrix estimation), the TLS problem formulation is the general mathematical model for homogeneous estimation. This paper will examine statistically optimal estimation for such models.

## 2 Error Models and Cost Functions

The additive TLS error model is defined by  $\mathbf{A} = \bar{\mathbf{A}} + \mathbf{D}$  with a true data matrix  $\bar{\mathbf{A}}$  and an error matrix  $\mathbf{D}$ , both of which being unknown. In this section, we will answer the question how to exploit information on the error model for a statistically optimized estimate of the sought parameter vector and present existing approaches in a unified framework.

### 2.1 TLS-Based Estimation Approaches

The first and second order statistical moments of the random matrix  $\mathbf{D}$  can be used to describe the error model. In the same way as mean vector and covariance matrix describe random, a mean matrix  $\mathbf{E}[\mathbf{D}]$  and a tensor rank 4 ( $(\mathcal{C}_{\mathbf{A}})_{ipjq} = \text{Cov}[(\mathbf{D})_{ip}, (\mathbf{D})_{jq}]$ ) characterize the statistical properties of random matrices like  $\mathbf{D}$ . Without loss of generality, we can assume zero-mean errors (we can always subtract  $\mathbf{E}[\mathbf{D}]$  from the measured data matrix  $\mathbf{A}$  if necessary), but the covariance tensor is much more problematic. It is definitely non-trivial to exploit this information in a statistically optimal way.

The TLS solution is widely equated with the right singular vector of  $\mathbf{A}$  corresponding to the smallest singular value. This narrows the view on the potential of TLS-based approaches considerably because it minimizes the squared  $|A\mathbf{p}|^2$  under the constraint  $|\mathbf{p}|^2 = 1$ , i.e. the cost function

$$J_{\text{AD}} = \frac{\mathbf{p}^T \mathbf{A}^T \mathbf{A} \mathbf{p}}{\mathbf{p}^T \mathbf{p}} = \sum_{i=1}^m \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{p}} \quad \text{with} \quad \mathbf{S}_i = \mathbf{a}_i \mathbf{a}_i^T \quad (3)$$

and with  $\mathbf{a}_i^T$  denoting the  $i$ -th row vector of  $\mathbf{A}$ . In section 2.2, we will prove that minimizing  $J_{\text{AD}}$  is statistically optimal if and only if the noise in the elements of the TLS data matrix  $\mathbf{A}$  is independent and identically distributed (iid).

It is important to stress that taking the right singular vector is just one of TLS-based methods; this method will be denoted plain TLS or PTLS from now on. The iid noise assumption connected with PTLS is often not very realistic; therefore, PTLS estimates can be very erroneous (e.g. highly biased in case of fundamental matrix estimation without prior data normalization).<sup>1</sup>

## 2.2 Cost Functions in Parameter Space

In the introduction, we presented the TLS model as general model for estimation of homogeneous vectors. PTLS does not consider covariance information at all, but such information can easily be taken into account, at least if we limit the general model: we will assume that we have a set of measurements  $\mathbf{x}_i$ , each of them can have an arbitrary covariance matrix  $\mathbf{B}_i$ , but they are assumed to be identical.<sup>2</sup> Then a minimum mean squared error (MMSE) estimate can be found by minimizing the cost function

$$J_{\text{ML}} = \sum_{i=1}^m (\hat{\mathbf{x}}_i - \mathbf{x}_i)^T \mathbf{B}_i^{-1} (\hat{\mathbf{x}}_i - \mathbf{x}_i) \quad (4)$$

with  $\hat{\mathbf{x}}_i$  being the estimates for true values of the measurements, i.e. estimates which exactly fulfill the data model (throughout this paper, quantities with a hat symbol on top will always denote estimated values). For Gaussian noise, this is also a maximum likelihood estimate (hence the ML subscript).

Unfortunately, the TLS data matrix  $\mathbf{A}$  is in general non-linear in the measurements  $\mathbf{x}_i$ . This makes minimization of  $J_{\text{ML}}$  intractable and therefore several different schemes were proposed which effectively replace (4) by an iterative solution of simple minimization problems. The most important existing methods of this class are Renormalization [6], Heteroscedastic Errors-In-Variables (HEIV, [7]), and Fundamental Numerical Scheme (FNS, [8]). All these methods are first

---

<sup>1</sup> Note that the iid noise assumption in the TLS data matrix  $\mathbf{A}$  is (in general) even violated if the underlying measurements  $\mathbf{x}_i$  contain iid noise because of the non-linearity of the constraints in (1) with respect to the measurements (e.g. conic fitting: linear in the six homogenous conic parameters, quadratic in the measurements). This effect is known as *heteroscedasticity* [3]. A preprocessing of the data can be used to alleviate the negative effects of heteroscedasticity; this technique is known as *conditioning* in photogrammetry or *data normalization* [4] in computer vision. Data normalization works because it makes the errors in  $\mathbf{D}$  more iid-like.

<sup>2</sup> The full theory including correlated measurements is much more complicated and can be found in the Ph.D. thesis of the first author [5].

order approximations to minimizing  $J_{\text{ML}}$ ; second order errors occur due to the linearization used for covariance propagation from measurements  $\mathbf{x}_i$  to the TLS data matrix  $\mathbf{A}$ .

An approximate maximum likelihood cost function  $J_{\text{AML}}$  for uncorrelated measurements can be defined as

$$J_{\text{AML}} = \sum_i \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}} \quad \text{with } \mathbf{S}_i = \mathbf{a}_i \mathbf{a}_i^T \quad \text{and } \mathbf{C}_i = \text{Cov}[\mathbf{a}_i]. \quad (5)$$

For  $\mathbf{C}_i = c\mathbf{I}$  with some constant  $c$  (iid errors), we find that the algebraic distance  $J_{\text{AD}}$  is statistically optimal. A derivation of  $J_{\text{AML}}$  can be found in [8], but a easier one is found in the appendix of [9]; we will summarize the latter one here. Let  $r_i = \mathbf{a}_i^T \mathbf{p}$  be the residual (deviation from data model) for the  $i$ -th measurement and let  $\mathbf{r} = (r_1, \dots, r_M)^T$  be the vector of residuals. For uncorrelated measurements, the covariance between different residuals is zero and for the variance  $\sigma_i^2$  of the residual  $r_i$ , we find

$$\sigma_i^2 = \left( \frac{\partial(\mathbf{p}^T \mathbf{a}_i)}{\partial \mathbf{x}_i} \right) \text{Cov}[\mathbf{x}_i] \left( \frac{\partial(\mathbf{a}_i^T \mathbf{p})}{\partial \mathbf{x}_i} \right)^T = \mathbf{p}^T \mathbf{C}_i \mathbf{p}. \quad (6)$$

Thus, the covariance matrix of the residual vector is given by  $\boldsymbol{\Sigma}_{\mathbf{rr}} := \text{Cov}[\mathbf{r}] = \text{diag}(\sigma_i^2)$ . Minimizing the Mahalanobis norm of the residual vector now means minimizing

$$J_{\text{AML}} = \mathbf{r}^T \boldsymbol{\Sigma}_{\mathbf{rr}}^{-1} \mathbf{r} = \sum_{i=1}^m (\mathbf{p}^T \mathbf{a}_i) \sigma_i^{-2} (\mathbf{a}_i^T \mathbf{p}) = \sum_{i=1}^m \frac{\mathbf{p}^T (\mathbf{a}_i \mathbf{a}_i^T) \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}} = \sum_{i=1}^m \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}} \quad (7)$$

which proves (5). We stress that all approaches which do not minimize  $J_{\text{ML}}$  are not optimal in the strict sense. However, the degree of approximation errors varies considerably: PTLS gives a coarse and biased estimation, while  $J_{\text{AML}}$  defines a criterion whose only difference to the optimal one is a second order error in covariance propagation.<sup>3</sup>

### 2.3 A General Framework for All Iterative Approaches

Different iterative algorithms are presented in totally different form and with very different reasoning: minimizing geometric distances, unbiasing of matrices by subtraction, solving variational equations. But they all share the same essential mathematical core which is determined by the type of problem: estimation of a vector. All these methods can be summarized under the following framework:

---

<sup>3</sup> For TLS models which are linear in both parameters and measurements, both criterions become identical. In general, however,  $J_{\text{AML}}$  is a first-order approximation to  $J_{\text{ML}}$ . Therefore, an estimation scheme need not necessarily be based on  $J_{\text{AML}}$ . One cannot criticize a scheme for not minimizing  $J_{\text{AML}}$  in these situations (for instance, FNS or our new approach do, Renormalization does not); there may be other first-order approximations to  $J_{\text{ML}}$  which are statistically equivalent.

- All methods compute eigenvectors
- The cost function  $J_{\text{ML}}$  (or an approximation for it like  $J_{\text{AML}}$ ) does not allow closed-form solutions; we therefore have to replace it by “something different” (as defined later) of the general form

$$J = \frac{\mathbf{p}^T \mathbf{X} \mathbf{p}}{\mathbf{p}^T \mathbf{p}} . \quad (8)$$

- The matrix  $\mathbf{X}$  will depend on some estimate  $\mathbf{q}$ .
- The cost functions  $J(\mathbf{p}; \mathbf{q})$  in (8) define a family of cost functions and every iteration step is essentially  $\hat{\mathbf{p}} := \arg \min_{\mathbf{p}} J(\mathbf{p}; \mathbf{q})$ .

The way in which  $\mathbf{X}(\mathbf{q})$  is derived can differ widely, but the general concept behind Sampson method, Fundamental Numerical Scheme, Renormalization, or Heteroscedastic EIV (and later: our new scheme IETLS) is always the same:

1. Set  $\mathbf{q}$  to some initial value, e.g. to  $\mathbf{q} := \arg \min_{\mathbf{p}} J_{\text{AD}} = \arg \min_{\mathbf{p}} \sum_i \mathbf{S}_i$ .
2. Compute matrix  $\mathbf{X}(\mathbf{q})$  using the previous estimate  $\mathbf{q}$ .
3. Compute eigenvector of  $\mathbf{X}$  corresponding to the smallest eigenvalue. Take this as estimate  $\hat{\mathbf{p}}$ .
4. Compare  $\hat{\mathbf{p}}$  and  $\mathbf{q}$ . Terminate if similar enough; otherwise set  $\mathbf{q} := \hat{\mathbf{p}}$  and continue at step 2.

Equation (8) is usually formulated as solving some equation  $\mathbf{X}\mathbf{p} = \mathbf{0}$  (“variational equation”, “renormalization equation” etc). But one actually minimizes an algebraic distance again. From our point of view, (8) emphasizes the structural similarity to  $J_{\text{AD}}$ . Other related papers only speak of a (hopefully converging)

We stress that, prior to computation of estimates, every algorithm defines a family of cost functions  $J(\mathbf{p}; \mathbf{q})$ . These cost functions  $J(\mathbf{p}; \mathbf{q})$  will converge “under favorable conditions”, and in some cases, one can identify their limit (if existent) as  $J_{\text{AML}}$  or any other function which can be expressed in algebraic form. The step from  $J_{\text{AML}}$  to a general model for  $J$  is found by writing

$$J_{\text{AML}} = \frac{\mathbf{p}^T \mathbf{M}(\mathbf{p}) \mathbf{p}}{\mathbf{p}^T \mathbf{p}} \quad \text{with} \quad \mathbf{M}(\mathbf{p}) = \sum_{i=1}^m \left( \frac{\mathbf{p}^T \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}} \mathbf{S}_i \right) . \quad (9)$$

The matrix  $\mathbf{M}(\mathbf{p})$  depends on the sought parameter vector—which makes an optimal closed-form solution impossible. But the assumption that current and previous estimate do not differ much allows the following iterative approach: replace  $\mathbf{M}(\mathbf{p})$  by some other matrix  $\mathbf{X}(\mathbf{q})$  which depends on the current estimate, thus being a function of  $\mathbf{p}$ . Then a new estimate can be computed as eigenvector corresponding to the smallest eigenvalue. In the following subsection, we will present different approaches for homogeneous estimation; the unifying framework presented here links them together.

While being based on a statistically justified cost function, all existing approaches share a common problem: they converge, but their limit has some favorable statistical properties. But the iterative schemes are based on little more than heuristics. As a consequence, these schemes can show severe convergence problems if error levels increase. In section 3, we will present a novel iterative scheme which shows much higher stability to noise.

## 2.4 Sampson's Method, FNS, Renormalization, and HEIV

An early class of approaches is defined by  $\mathbf{X}(\mathbf{q}) := \sum_i \alpha_i^2 \mathbf{S}_i$  such that the weights  $\alpha_i$  are functions of the previous estimate. This approach is known as... . For ellipse fitting, the most obvious choice (cf. (9))

$$\alpha_i^2(\mathbf{q}) = \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \quad \Rightarrow \quad \mathbf{X}(\mathbf{q}) := \sum_i \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \mathbf{S}_i \quad (10)$$

was proposed by Sampson [10]; this defines a first order approximation to the distance between data point and estimated ellipse. Similar approaches were proposed for other computer vision problems like fundamental matrix estimation. These approaches are better than PTLS, but the corresponding series of cost function does not converge to  $J_{\text{AML}}$  or any other first order approximation to  $J_{\text{ML}}$ . Consequently, the estimates are still biased.

In contrast to this, the FNS method [8] has the strong statistical justification of minimizing  $J_{\text{AML}}$  and we will summarize it next. At the minimum, the gradient of  $J_{\text{AML}}$  with respect to the parameter vector must be zero. Computing the derivative and setting it to zero leads to a so-called “variational equation” which essentially means defining

$$\mathbf{X}(\mathbf{q}) := \sum_{i=1}^m \left( \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \mathbf{S}_i - \frac{(\mathbf{q}^T \mathbf{q})(\mathbf{q}^T \mathbf{S}_i \mathbf{q})}{(\mathbf{q}^T \mathbf{C}_i \mathbf{q})^2} \mathbf{C}_i \right) = \sum_{i=1}^m \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \left( \mathbf{S}_i - \frac{\mathbf{q}^T \mathbf{S}_i \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \mathbf{C}_i \right) \quad (11)$$

according to the framework described above. The second summand in (11) can be regarded as a correction term which makes up for inserting  $\mathbf{q}$  in place of  $\mathbf{p}$ .

this algorithm converges,... it converges to the correct solution (up to the second order errors which distinguish  $J_{\text{AML}}$  from  $J_{\text{ML}}$ ). But unfortunately, experiments show that the initial value must be very close to the optimum and the noise level must be low; otherwise, the algorithm is likely to diverge. Additionally, the algorithm is extremely sensitive to prior data transformation / normalization; this problem-specific property limits the generality of an algorithm. The reason for all these negative effects is simple: each iteration step silently assumes iid errors again; otherwise, eigensystem-based methods are not optimal. Therefore, the iteration steps are not solved in a statistically optimal way.<sup>4</sup> Our new algorithm will be free of this drawback.

A third approach, the Renormalization scheme defined by Kanatani [6], has another justification: correction matrices  $\mathbf{D}_i$  are chosen such that

$$\mathbf{X}(\mathbf{q}) := \sum_{i=1}^m \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} (\mathbf{S}_i - \mathbf{D}_i) \quad \text{with} \quad \mathbb{E} \left[ \frac{\mathbf{q}^T \mathbf{S}_i \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \right] = \mathbb{E} \left[ \frac{\mathbf{q}^T \mathbf{D}_i \mathbf{q}}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}} \right] \quad (12)$$

holds, this approach is motivated by subtracting the ‘bias’ in  $\mathbf{S}_i$ . The term ‘bias’ is put in quotes because one can show that this bias does not affect the eigen-

---

<sup>4</sup> One evidence is the need to look for the eigenvalue smallest in *absolute* value in FNS.

A positive definite matrix should not have negative eigenvalues. But the difference in the numerator of  $\mathbf{X}$  is dangerous for  $\mathbf{p} \neq \mathbf{q}$ .

vectors as long as it is proportional to the identity matrix in expectation (then only the eigenvalues are increased). Therefore,  $\mathbf{E}[\mathbf{q}^T \mathbf{S}_i \mathbf{q}] \neq 0$  is not necessarily a bad sign. Nevertheless, the renormalization approach leads to high quality estimation schemes. However, just like FNS we find that the matrix  $\mathbf{X}$  is given as a non-negative definite matrices which is not necessarily non-negative definite as well. Higher errors can lead to convergence problems again.

A further approach is the heteroscedastic errors-in-variables (HEIV) model which was introduced by Leedan [11] and refined by Matei and Meer [7]. A recent paper [12] illustrated the link between FNS and HEIV. Consequently, HEIV shows similar properties and problems (for instance, Nestares et al. report convergence problems in [13]).

The mathematical core of all methods presented so far is very similar: iterative estimation of (generalized) eigenvectors. At the end of this section, we stress that eigenvector-based estimation is not very tolerant to anisotropic (i.e. non-iid) errors. If some eigenvalue  $\bar{\lambda}$ , corresponding to the true solution  $\bar{\mathbf{p}}$ , is close to another eigenvalue  $\lambda$ , corresponding to a wrong estimate  $\tilde{\mathbf{p}}$ , then very small errors can change the order of eigenvalues. As eigenvectors are mutually orthogonal, a small increase in noise can lead to a wrong estimate. This holds for every iteration step. Possible divergence is a severe problem and one should not hope for graceful degradation in homogeneous estimation unless isotropic behavior can be guaranteed for each estimation step.<sup>5</sup> The deeper mathematical understanding of homogeneous estimation presented so far will now allow us to derive a more stable estimation scheme in the following section.

### 3 A Novel Iterative Estimation Approach: IETLS

The  $J_{\text{AML}}$  cost function can be rewritten as

$$J_{\text{AML}} = \sum_i \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}} = \sum_i \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{S}}_i \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{C}}_i \tilde{\mathbf{p}}} \quad (13)$$

with  $\tilde{\mathbf{p}} = \mathbf{W}_R^{-T} \mathbf{p}$  and  $\tilde{\mathbf{S}}_i = \lambda_i^2 \mathbf{W}_R \mathbf{S}_i \mathbf{W}_R^T$  and  $\tilde{\mathbf{C}}_i = \lambda_i^2 \mathbf{W}_R \mathbf{C}_i \mathbf{W}_R^T$ . Instead of minimizing  $J_{\text{AML}}$  directly in the original coordinate system, we have the freedom to choose weights  $\lambda_i$  for each data vector and a weight matrix  $\mathbf{W}_R$  in order to transform the problem of minimizing  $J_{\text{AML}}$  to a more convenient coordinate system. The cost function  $J_{\text{AML}}$  is invariant to transformations of the coordinate system (which is exactly what we can expect from a statistically justified cost function: its result must not depend on the choice of coordinates).

The minimizers  $\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} (J_{\text{AML}})$  (in the original coordinate system) and  $\hat{\tilde{\mathbf{p}}} = \arg \min_{\tilde{\mathbf{p}}} (J_{\text{AML}})$  (in the transformed system) are defined only up to scale;

---

<sup>5</sup> This is also the reason why it is so easy to find limiting statements like “if initialized close to the global optimum” or “under favorable condition” or similar thing written between the lines in algorithmic papers for homogeneous estimation problems.

therefore the back substitution  $\hat{\mathbf{p}} = \mathbf{W}_R^T \hat{\mathbf{p}}$  of the solution in the transformed space is always a solution in the original space. This basic reweighting technique known as [14, 15] can be seen as a tool to make minimizing  $J_{\text{AML}}$  by choosing a different coordinate system.

It is obvious that equilibration changes error metrics but we have not defined yet to choose these weights optimally (it is well-known weight matrices influence the result (e.g. [16]), but the really interesting question is to find optimal weights). We will answer this question now and develop an iterative scheme based on it. First, we rewrite  $J_{\text{AML}}$  as  $J_{\text{AML}} = \sum_i r_i^2 = \sum_i r_i'^2 \mu_i^2$  with

$$r_i^2 = \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{S}}_i \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{C}}_i \tilde{\mathbf{p}}} \quad \text{and} \quad r_i'^2 = \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{S}}_i \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{p}}} \quad \text{and} \quad \mu_i^2 = \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{C}}_i \tilde{\mathbf{p}}}. \quad (14)$$

Here, we introduced the  $r_i'$  that do not consider the  $\mu_i^2$  weights and the  $r_i$  which include the data-dependent denominator. The true residuals are invariant to coordinate transformations (see (13)), but the raw residuals are not. From this point of view, the basic idea behind (we will denote this as IETLS) can be described as follows: taking the singular vector corresponding to the smallest singular value (i.e. a PTLS solution) finds the global minimum of  $J_{\text{IETLS}} = \sum_i r_i'^2$ , i.e. for minimizing the sum of squared residuals. By iteratively updating the equilibration weights (i.e. warping the space in which the estimation is carried out) we can make all  $\mu_i^2$  converge to 1 easily. Then the raw residuals converge to the true residuals, the IETLS cost function  $J_{\text{IETLS}}$  converges to  $J_{\text{AML}}$ , and a simple PTLS solution in equilibrated space is now a minimizer of  $J_{\text{AML}}$ .

Comparing the three functions  $J_{\text{AD}}$ ,  $J_{\text{IETLS}}$ , and  $J_{\text{AML}}$  shows the difference:

$$J_{\text{AD}} = \sum_i \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{p}}, \quad J_{\text{IETLS}} = \sum_i \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{S}}_i \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{p}}}, \quad J_{\text{AML}} = \sum_i \frac{\mathbf{p}^T \mathbf{S}_i \mathbf{p}}{\mathbf{p}^T \mathbf{C}_i \mathbf{p}}.$$

$J_{\text{AD}}$  neglects covariances completely,  $J_{\text{AML}}$  is hardly tractable, but  $J_{\text{IETLS}}$  includes covariances indirectly by appropriate transformation of coordinates while sharing the same mathematical form as  $J_{\text{AD}}$ . It is the formerly missing link between the available methods (i.e. computing singular vectors) and the statistically justified cost function  $J_{\text{AML}}$ . This idea can be used to define an iterative scheme: we first generate initial weights using

$$\lambda_i = \sqrt{\frac{1}{\text{tr}[\mathbf{C}_i]}} \quad \text{and} \quad \mathbf{W}_R = \text{itChol} \left[ \sum_i \lambda_i^2 \mathbf{C}_i \right] \quad (15)$$

where  $\text{itChol}[\cdot]$  stands for inverting and transposing the Cholesky factor of some matrix. Then define the Basic Iterative Equilibrated TLS Scheme (B-IETLS) as follows:

1. Estimate new parameter  $\hat{\mathbf{p}}$  vector in equilibrated space using

$$\hat{\mathbf{p}} = \mathbf{W}_R^T \left( \arg \min_{\tilde{\mathbf{p}}} \frac{\tilde{\mathbf{p}}^T \tilde{\mathbf{S}} \tilde{\mathbf{p}}}{\tilde{\mathbf{p}}^T \tilde{\mathbf{p}}} \right) \quad \text{with} \quad \tilde{\mathbf{S}} = \sum_i \lambda_i^2 \mathbf{W}_R \mathbf{S}_i \mathbf{W}_R^T. \quad (16)$$

2. Compute new equilibration transformations from

$$\lambda_i = \sqrt{\frac{1}{\mathbf{q}^T \mathbf{C}_i \mathbf{q}}} \quad \text{and} \quad \mathbf{W}_R := \text{itChol} \left[ \sum_i \lambda_i^2 r_i^2 \mathbf{C}_i \right]. \quad (17)$$

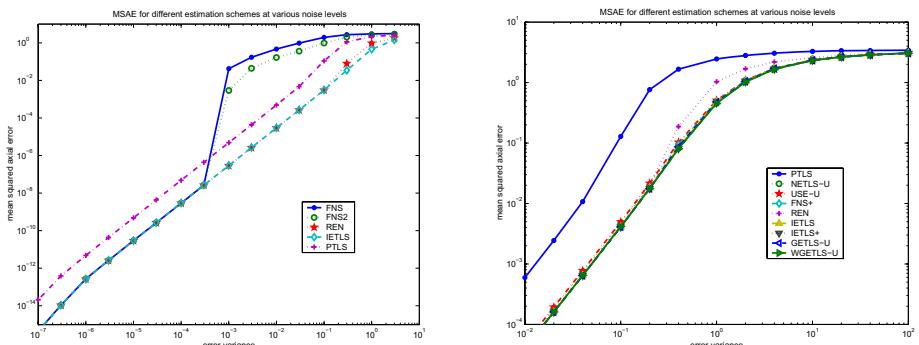
3. Terminate if estimated  $\hat{\mathbf{p}}$  did not change much. Otherwise: next iteration.

It is important to stress that IETLS . . . . . chooses an appropriate coordinate system in . . . . . Other iterative schemes implicitly assume that some data normalization (which is a subset of possible right equilibrations) is carried out as a preprocessing step; the correct type of data transformation has to be known in advance and this transformation is not adaptive during the iterative process. The scheme described above is called . . . IETLS because it can be refined further. We can decrease computational complexity and simultaneously increase numerical stability (e.g. by computing singular vectors instead of eigenvectors of matrix products)—unfortunately at the expense of longer and less intuitive code. The refined version (including MATLAB sources) can be found in [5], but the general idea should also be clear from B-IETLS.

## 4 Experimental Evaluation and Summary

We successfully applied the IETLS scheme to several computer vision applications including fundamental matrix estimation, homography estimation and orientation estimation (which includes motion estimation as a special case for space-time volumes); see [5] for details and examples. In this general and more theoretical paper, however, we will show numerical simulations.

Figs. 1 shows the mean squared axial error for a simulation in which a homogeneous vector in  $\mathbb{P}^5$  was estimated (details can be found in [5]). In the left



**Fig. 1.** Mean squared axial error for different noise levels; complete noise range in left image ( $\rightarrow$  FNS often fails to converge) and medium to higher noise levels only in right image ( $\rightarrow$  REN shows weaknesses at higher noise levels)

image, it is clearly visible that FNS breaks down for medium error levels; its estimates even become worse than PTLS. The right image shows a zoom on the upper right part of the curve: renormalization (REN) is more stable but shows slight weaknesses for higher noise levels. In all simulations, IETLS and some variants of it which were also tested perform best.

Summarizing this paper, we emphasize that iterative estimation of homogeneous vectors can be highly sensitive to noise. Therefore, it is not sufficient to define an iterative scheme which converges to some statistically optimized solution—provided that it converges at all. We showed that the essential mathematical core of all homogeneous estimation approaches is some eigensystem analysis. This allowed to identify some underlying general framework and in this framework, we could derive a novel scheme in which the estimation process is carried out in a statistically optimal way. Experimental evaluation (in this paper and extended evaluation including several computer vision examples in [5]) showed the superiority of our novel approach.

## References

1. Mardia, K.V., Jupp, P.E.: *Directional Statistics*. Wiley (2000)
2. van Huffel, S., Vandewalle, J.: *The Total Least Squares problem: Computational aspects and analysis*. SIAM, Philadelphia (1991)
3. Mardia, K.V., Kent, J., Bibby, J.: *Multivariate Analysis*. Academic Press (1992)
4. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. First edn. Cambridge University Press (2000)
5. Mühlbach, M.: *Estimation of Homogeneous Vectors and Applications in Computer Vision*. PhD thesis, J. W. Goethe-Universität Frankfurt (2004)
6. Kanatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier (1996)
7. Matei, B., Meer, P.: A general method for errors-in-variables problems in computer vision. In: IEEE Conf. CVPR. (2000) 18–25
8. Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: On the fitting of surfaces to data with covariances. *IEEE Trans. PAMI* **22** (2000) 1294–1303
9. Förstner, W.: On Estimating 2D Points and Lines from 2D Points and Lines. In: *Festschrift anlässlich des 60. Geburtstages von Prof. Dr.-Ing. Bernhard Wrobel*. Technische Universität Darmstadt (2001) 69 – 87
10. Sampson, P.D.: Fitting conic sections to ‘very scattered’ data: An iterative refinement of the Bookstein algorithm. *Computer Vision Graphics and Image Processing* **18** (1982) 97–108
11. Leedan, Y., Meer, P.: Heteroscedastic regression in computer vision: problems with bilinear constraint. *Int. J. Computer Vision* **37** (2000) 127–150
12. Chojnacki, W., Brooks, M., van den Hengel, A., Gawley, D.: From FNS to HEIV: A link between two vision parameter estimation methods. *IEEE Trans. PAMI* **26** (2004) 85–91
13. Nestares, O., Fleet, D.J.: Error-in-variables likelihood functions for motion estimation. In: *IEEE International Conference on Image Processing*, Barcelona. (2003)

14. Mühlich, M., Mester, R.: The role of total least squares in motion analysis. In: Proc. Europ. Conf. Comp. Vision. (1998)
15. Mühlich, M., Mester, R.: Subspace methods and equilibration in computer vision. Technical Report XP-TR-C-21, J.W.G.University Frankfurt (1999)
16. Golub, G.H., van Loan, C.F.: Matrix Computations. 2nd edition edn. The John Hopkins University Press, Baltimore and London (1989)

# Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction

Zhijian Yuan and Erkki Oja

Neural Networks Research Centre,  
Helsinki University of Technology,  
P.O.Box 5400, 02015 HUT, Finland  
[{zhijian.yuan, erkki.oja}@hut.fi](mailto:{zhijian.yuan, erkki.oja}@hut.fi)

**Abstract.** In image compression and feature extraction, linear expansions are standardly used. It was recently pointed out by Lee and Seung that the positivity or non-negativity of a linear expansion is a very powerful constraint, that seems to lead to sparse representations for the images. Their technique, called Non-negative Matrix Factorization (NMF), was shown to be a useful technique in approximating high dimensional data where the data are comprised of non-negative components. We propose here a new variant of the NMF method for learning spatially localized, sparse, part-based subspace representations of visual patterns. The algorithm is based on positively constrained projections and is related both to NMF and to the conventional SVD or PCA decomposition. Two iterative positive projection algorithms are suggested, one based on minimizing Euclidean distance and the other on minimizing the divergence of the original data matrix and its non-negative approximation. Experimental results show that P-NMF derives bases which are somewhat better suitable for a localized representation than NMF.

## 1 Introduction

For compressing, denoising and feature extraction of digital image windows, one of the classical approaches is Principal Component Analysis (PCA) and its extensions and approximations such as the Discrete Cosine Transform. In PCA or the related Singular Value Decomposition (SVD), the image is projected on the eigenvectors of the image covariance matrix, each of which provides one linear feature. The representation of an image in this basis is  $\dots \dots$  in the sense that typically all the features are used at least to some extent in the reconstruction.

Another possibility is a  $\dots \dots$  representation, in which any given image window is spanned by just a small subset of the available features [1, 2, 6, 10]. This kind of representations have some biological significance, as the sparse features seem to correspond to the receptive fields of simple cells in the area V1 of the mammalian visual cortex. This approach is related to the technique of Independent Component Analysis [3] which can be seen as a nongaussian extension of PCA and Factor Analysis.

Recently, it was shown by Lee and Seung [4] that a linear expansion is a very powerful constraint that also seems to yield sparse representations. Their technique, called Non-negative Matrix Factorization (NMF), was shown to be a useful technique in approximating high dimensional data where the data are comprised of non-negative components. The authors proposed the idea of using NMF techniques to find a set of basis functions to represent image data where the basis functions enable the identification and classification of intrinsic “parts” that make up the object being imaged by multiple observations. NMF has been typically applied to image and text data [4, 9], but has also been used to deconstruct music tones [8].

NMF imposes the non-negativity constraints in learning the basis images. Both the values of the basis images and the coefficients for reconstruction are all non-negative. The additive property ensures that the components are combined to form a whole in the non-negative way, which has been shown to be the part-based representation of the original data. However, the additive parts learned by NMF are not necessarily localized.

In this paper, we start from the ideas of SVD and NMF and propose a novel method which we call Projective Non-negative Matrix Factorization (P-NMF), for learning spatially localized, parts-based representations of visual patterns. First, in Section 2, we take a look at a simple way to produce a positive SVD by truncating away negative parts. Section 3 briefly reviews Lee’s and Seung’s NMF. Using this as a baseline, we present our P-NMF method in Section 4. Section 5 gives some experiments and comparisons, and Section 6 concludes the paper.

## 2 Truncated Singular Value Decomposition

Suppose that our data<sup>1</sup> is given in the form of an  $m \times n$  matrix  $\mathbf{V}$ . Its  $n$  columns are the data items, for example, a set of images that have been vectorized by row-by-row scanning. Then  $m$  is the number of pixels in any given image. Typically,  $n > m$ . The Singular Value Decomposition (SVD) for matrix  $\mathbf{V}$  is

$$\mathbf{V} = \mathbf{U}\hat{\mathbf{D}}\hat{\mathbf{U}}^T, \quad (1)$$

where  $\mathbf{U}$  ( $m \times m$ ) and  $\hat{\mathbf{U}}$  ( $n \times m$ ) are orthogonal matrices consisting of the eigenvectors of  $\mathbf{V}\mathbf{V}^T$  and  $\mathbf{V}^T\mathbf{V}$ , respectively, and  $\hat{\mathbf{D}}$  is a diagonal  $m \times m$  matrix where the diagonal elements are the ordered singular values of  $\mathbf{V}$ .

Choosing the  $r$  largest singular values of matrix  $\mathbf{V}$  to form a new diagonal  $r \times r$  matrix  $\hat{\mathbf{D}}$ , with  $r < m$ , we get the compressive SVD matrix  $\mathbf{X}$  with given rank  $r$ ,

$$\mathbf{X} = \mathbf{U}\hat{\mathbf{D}}\hat{\mathbf{U}}^T. \quad (2)$$

---

<sup>1</sup> For clarity, we use here the same notation as in the original NMF theory by Lee and Seung

Now both matrices  $\mathbf{U}$  and  $\hat{\mathbf{U}}$  have only  $r$  columns corresponding to the  $r$  largest eigenvalues. The compressive SVD gives the best approximation  $\mathbf{X}$  of the matrix  $\mathbf{V}$  with the given compressive rank  $r$ .

In many real-world cases, for example, for images, spectra etc., the original data matrix  $\mathbf{V}$  is non-negative. Then the above compressive SVD matrix  $\mathbf{X}$  fails to keep the nonnegative property. In order to further approximate it by a non-negative matrix, the following truncated SVD (tSVD) is suggested. We simply truncate away the negative elements by

$$\hat{\mathbf{X}} = \frac{1}{2}(\mathbf{X} + \text{abs}(\mathbf{X})). \quad (3)$$

However, it turns out that typically the matrix  $\hat{\mathbf{X}}$  in (3) has higher rank than  $\mathbf{X}$ . Truncation destroys the linear dependences that are the reason for the low rank. In order to get an equal rank, we have to start from a compressive SVD matrix  $\mathbf{X}$  with lower rank than the given  $r$ . Therefore, to find the truncated matrix  $\hat{\mathbf{X}}$  with the compressive rank  $r$ , we search all the compressive SVD matrices  $\mathbf{X}$  with the rank from 1 to  $r$  and form the corresponding truncated matrices. The one with the largest rank that is less than or equal to the given rank  $r$  is the truncated matrix  $\hat{\mathbf{X}}$  what we choose as the final non-negative approximation. This matrix can be used as a baseline in comparisons, and also as a starting point in iterative improvements. We call this method truncated SVD (t-SVD).

Note that the tSVD only produces the non-negative low-rank approximation  $\hat{\mathbf{X}}$  to the data matrix  $\mathbf{V}$ , but does not give a separable expansion for basis vectors and weights as the usual SVD expansion.

### 3 Non-negative Matrix Factorization

Given the nonnegative  $m \times n$  matrix  $\mathbf{V}$  and the constant  $r$ , the Nonnegative Matrix Factorization algorithm (NMF) [4] finds a nonnegative  $m \times r$  matrix  $\mathbf{W}$  and another nonnegative  $r \times n$  matrix  $\mathbf{H}$  such that they minimize the following optimality problem:

$$\min_{\mathbf{W}, \mathbf{H} \geq 0} \|\mathbf{V} - \mathbf{WH}\|. \quad (4)$$

This can be interpreted as follows: each column of matrix  $\mathbf{W}$  contains a basis vector while each column of  $\mathbf{H}$  contains the weights needed to approximate the corresponding column in  $\mathbf{V}$  using the basis from  $\mathbf{W}$ . So the product  $\mathbf{WH}$  can be regarded as a compressed form of the data in  $\mathbf{V}$ . The rank  $r$  is usually chosen so that  $(n+m)r < nm$ .

In order to estimate the factorization matrices, an objective function defined by the authors as Kullback-Leibler divergence is

$$\mathbf{F} = \sum_{i=1}^m \sum_{\mu=1}^n [\mathbf{V}_{i\mu} \log(\mathbf{WH})_{i\mu} - (\mathbf{WH})_{i\mu}]. \quad (5)$$

This objective function can be related to the likelihood of generating the images in  $\mathbf{V}$  from the basis  $\mathbf{W}$  and encodings  $\mathbf{H}$ . An iterative approach to

reach a local maximum of this objective function is given by the following rules [4, 5]:

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \sum_{\mu} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}} \mathbf{H}_{a\mu}, \mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia}}{\sum_j \mathbf{W}_{ja}} \quad (6)$$

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \sum_i \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}. \quad (7)$$

The convergence of the process is ensured<sup>2</sup>. The initialization is performed using positive random initial conditions for matrices  $\mathbf{W}$  and  $\mathbf{H}$ .

## 4 The Projective NMF Method

### 4.1 Definition of the Problem

The compressive SVD is a projection method. It projects the data matrix  $\mathbf{V}$  onto the subspace of the eigenvectors of the data covariance matrix. Although the truncated method t-SVD outlined above works and keeps nonnegativity, it is not accurate enough for most cases. To improve it, for the given  $m \times n$  nonnegative matrix  $\mathbf{V}$ ,  $m < n$ , let us try to find a subspace  $\mathcal{B}$  of  $R^m$ , and an  $m \times m$  projection matrix  $\mathbf{P}$  with given rank  $r$  such that  $\mathbf{P}$  projects the nonnegative matrix  $\mathbf{V}$  onto the subspace  $\mathcal{B}$  and keeps the nonnegative property, that is,  $\mathbf{PV}$  is a nonnegative matrix. Finally, it should minimize the difference  $\|\mathbf{V} - \mathbf{PV}\|$ . This is the basic idea of the Projective NMF method.

We can write any symmetrical projection matrix of rank  $r$  in the form

$$\mathbf{P} = \mathbf{WW}^T \quad (8)$$

with  $\mathbf{W}$  an orthogonal  $(m \times r)$  matrix<sup>3</sup>. Thus, we can solve the problem by searching for a nonnegative  $(m \times r)$  matrix  $\mathbf{W}$ . Based on this, we now introduce a novel method which we call  $\dots$  (P-NMF) as the solution to the following optimality problem

$$\min_{\mathbf{W} \geq 0} \|\mathbf{V} - \mathbf{WW}^T \mathbf{V}\|, \quad (9)$$

where  $\|\cdot\|$  is a matrix norm. The most useful norms are the Euclidean distance and the divergence of matrix  $\mathbf{A}$  from  $\mathbf{B}$ , defined as follows: The Euclidean distance between two matrices  $\mathbf{A}$  and  $\mathbf{B}$  is

---

<sup>2</sup> The matlab program for the above update rules is available at <http://journalclub.mit.edu> under the "Computational Neuroscience" discussion category.

<sup>3</sup> This is just notation for a generic basis matrix; the solution will not be the same as the  $\mathbf{W}$  matrix in NMF.

$$\|\mathbf{A} - \mathbf{B}\|^2 = \sum_{i,j} (\mathbf{A}_{ij} - \mathbf{B}_{ij})^2, \quad (10)$$

and the divergence of  $\mathbf{A}$  from  $\mathbf{B}$

$$D(\mathbf{A} \parallel \mathbf{B}) = \sum_{i,j} (\mathbf{A}_{ij} \log \frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}} - \mathbf{A}_{ij} + \mathbf{B}_{ij}). \quad (11)$$

Both are lower bounded by zero, and vanish if and only if  $\mathbf{A} = \mathbf{B}$ .

## 4.2 Algorithms

We first consider the Euclidean distance (10). Define the function

$$\mathbf{F} = \frac{1}{2} \|\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}\|^2. \quad (12)$$

Then the unconstrained gradient of  $\mathbf{F}$  for  $\mathbf{W}$ ,  $\frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}}$ , is given by

$$\frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}} = -2(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}. \quad (13)$$

Using the gradient we can construct the additive update rule for minimization,

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} - \eta_{ij} \frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}} \quad (14)$$

where  $\eta_{ij}$  is the positive step size.

However, there is nothing to guarantee that the elements  $\mathbf{W}_{ij}$  would stay non-negative. In order to ensure this, we choose the step size as follows,

$$\eta_{ij} = \frac{\mathbf{W}_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \quad (15)$$

Then the additive update rule (14) can be formulated as a multiplicative update rule,

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \quad (16)$$

Now it is guaranteed that the  $\mathbf{W}_{ij}$  will stay nonnegative, as everything on the right-hand side is nonnegative.

For the divergence measure (11), we follow the same process. First we calculate the gradient

$$\frac{\partial D(\mathbf{V} \parallel \mathbf{W}\mathbf{W}^T\mathbf{V})}{\partial \mathbf{w}_{ij}} = \sum_k \left( (\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj} \mathbf{V}_{ik} \right) \quad (17)$$

$$- \sum_k \mathbf{V}_{ik} (\mathbf{W}^T\mathbf{V})_{jk} / (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} \quad (18)$$

$$- \sum_k \mathbf{V}_{ik} \sum_l \mathbf{W}_{lj} \mathbf{V}_{lk} / (\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk}. \quad (19)$$

Using the gradient, the additive update rule becomes

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} + \zeta_{ij} \frac{\partial D(\mathbf{V} || \mathbf{WW}^T \mathbf{V})}{\partial \mathbf{w}_{ij}} \quad (20)$$

where  $\zeta_{ij}$  is the step size. Choosing this step size as following,

$$\zeta_{ij} = \frac{\mathbf{W}_{ij}}{\sum_k \mathbf{V}_{ik} [(\mathbf{W}^T \mathbf{V})_{jk} / (\mathbf{WW}^T \mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj} \mathbf{V}_{lk} / (\mathbf{WW}^T \mathbf{V})_{lk}]} \quad (21)$$

we obtain the multiplicative update rule

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\sum_k ((\mathbf{W}^T \mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj} \mathbf{V}_{ik})}{\sum_k \mathbf{V}_{ik} ((\mathbf{W}^T \mathbf{V})_{jk} / (\mathbf{WW}^T \mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj} \mathbf{V}_{lk} / (\mathbf{WW}^T \mathbf{V})_{lk})} \quad (22)$$

It is easy to see that both multiplicative update rules (16) and (22) can ensure that the matrix  $\mathbf{W}$  is non-negative.

### 4.3 The Relationship Between NMF and P-NMF

There is a very obvious relationship between our P-NMF algorithms and the original NMF. Comparing the two optimality problems, P-NMF (9) and the original NMF (4), we see that the weight matrix  $\mathbf{H}$  in NMF is simply replaced by  $\mathbf{W}^T \mathbf{V}$  in our algorithms. Both multiplicative update rules (16) and (22) are obtained similar to Lee and Seung's algorithms [5]. Therefore, the convergence of these two algorithms can also be proved following Lee and Seung [5] by noticing that the coefficient matrix  $\mathbf{H}$  is replaced by  $\mathbf{WV}$ .

### 4.4 The Relationship Between SVD and P-NMF

There is also a relationship between the P-NMF algorithm and the SVD. For the Euclidean norm, note the similarity of the problem (9) with the conventional PCA for the columns of  $\mathbf{V}$ . Removing the positivity constraint, this would become the usual finite-sample PCA problem, whose solution is known to be an orthogonal matrix consisting of the eigenvectors of  $\mathbf{VV}^T$ . But this is the matrix  $\mathbf{U}$  in the SVD of eq. (1). However, now with the positivity constraint in place, the solution will be something quite different.

## 5 Simulations

### 5.1 Data Preparation

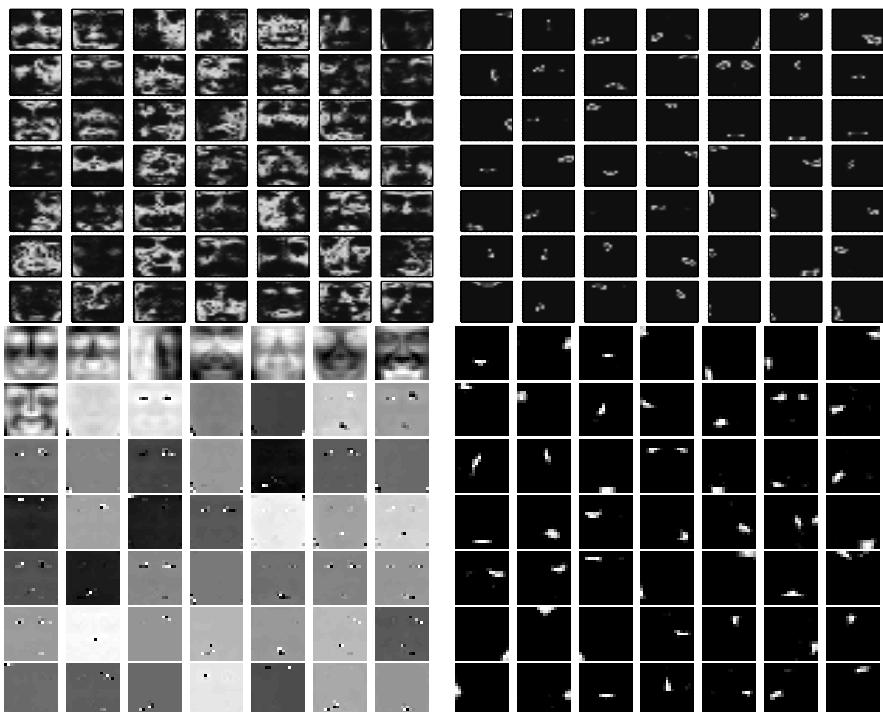
As experimental data, we used face images from the MIT-CBCL database and derived the NMF and P-NMF expansions for them. The training data set contains 2429 faces. Each face has  $19 \times 19 = 361$  pixels and has been histogram-equalized and normalized so that all pixel values are between 0 and 1. Thus

the data matrix  $\mathbf{V}$  which now has the faces as columns is  $361 \times 2429$ . This matrix was compressed to rank  $r = 49$  using either t-SVD, NMF, or P-NMF expansions.

## 5.2 Learning Basis Components

The basis images of tSVD, NMF, and P-NMF with dimension 49 are shown in Figure 1. For NMF and P-NMF, these are the 49 columns of the corresponding matrices  $\mathbf{W}$ . For t-SVD, we show the 49 basis vectors of the range space of the rank-49 nonnegative matrix  $\hat{\mathbf{X}}$ , obtained by ordinary SVD of this matrix. Thus the basis images for NMF and P-NMF are truly non-negative, while the t-SVD only produces a non-negative overall approximation to the data but does not give a separable expansion for basis vectors and weights.

All the images are displayed with the matlab command "imagesc" without any extra scale. Both NMF and P-NMF bases are holistic for the training set. For this problem, the P-NMF algorithm converges about 5 times faster than NMF.



**Fig. 1.** NMF (top, left), t-SVD (bottom, left) and the two versions of the new P-NMF method (right) bases of dimension 49. Each basis component consists of  $19 \times 19$  pixels

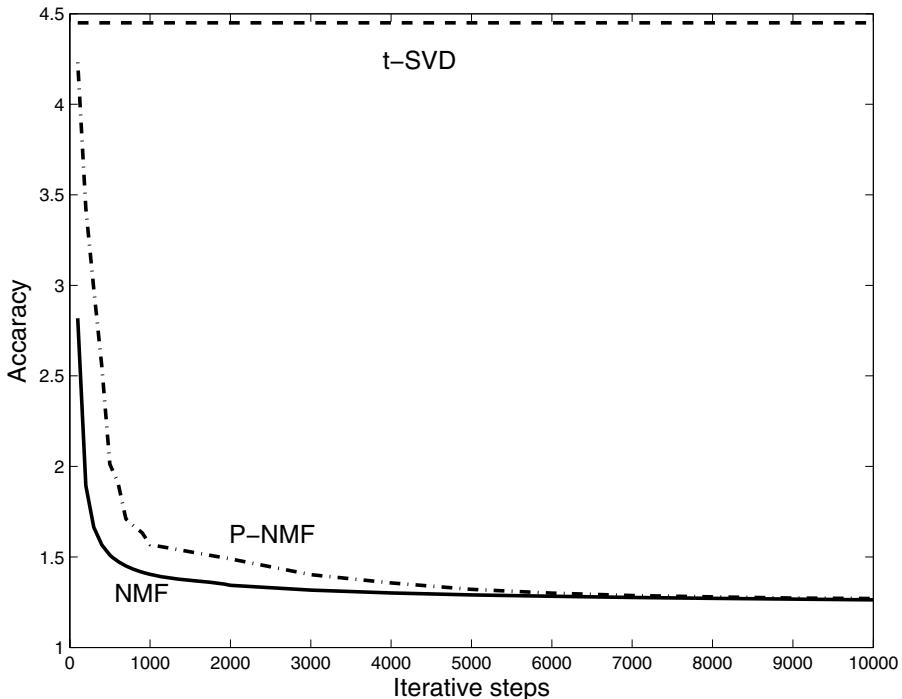


**Fig. 2.** The original face image (left) and its reconstructions by NMF (top row), the two versions of the new P-NMF method under 100 iterative steps (second and third rows), and t-SVD (bottom row). The dimensions in columns 2, 3, and 4 are 25, 49 and 81, respectively

### 5.3 Reconstruction Accuracy

We repeated the above computations for ranks  $r = 25, 49$  and  $81$ . Figure 2 shows the reconstructions for one of the face images in the t-SVD, NMF, and P-NMF subspaces of corresponding dimensions. For comparison, also the original face image is shown. As the dimension increases, more details are recovered. Visually, the P-NMF method is comparable to NMF.

The recognition accuracy, defined as the Euclidean distance between the original data matrix and the recognition matrix, can be used to measure the performance quantitatively. Figure 3 shows the recognition accuracy curves of P-NMF and NMF under different iterative steps. NMF converges faster, but when the number of steps increases, P-NMF works very similarly to NMF. One thing to be noticed is that the accuracy of P-NMF depends on the initial values. Although the number of iteration steps is larger in P-NMF for comparable error with NMF, this is compensated by the fact that the computational complexity for one iteration step is considerably lower for P-NMF, as only one matrix has to be updated instead of two.



**Fig. 3.** Recognition accuracies (unit:  $10^8$ ) versus iterative steps using t-SVD, NMF and P-NMF with compressive dimension 49

## 6 Conclusion

We proposed a new variant of the well-known Non-negative Matrix Factorization (NMF) method for learning spatially localized, sparse, part-based subspace representations of visual patterns. The algorithm is based on positively constrained projections and is related both to NMF and to the conventional SVD decomposition. Two iterative positive projection algorithms were suggested, one based on minimizing Euclidean distance and the other on minimizing the divergence of the original data matrix and its approximation. Compared to the NMF method, the iterations are somewhat simpler as only one matrix is updated instead of two as in NMF. The tradeoff is that the convergence, counted in iteration steps, is slower than in NMF.

One purpose of these approaches is to learn localized features which would be suitable not only for image compression, but also for object recognition. Experimental results show that P-NMF derives bases which are better suitable for a localized representation than NMF. It remains to be seen whether they would be better in pattern recognition, too.

## References

1. A. Bell and T. Sejnowski. The "independent components" of images are edge filters. *Vision Research*, 37: 3327–3338, 1997.
2. A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 13: 1527–1558, 2001.
3. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.
4. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
5. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.
6. B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Network*, 7: 333–339, 1996.
7. P. Paatero and U. Tapper. Positive Matrix Factorization: A non-negative factor model with optimal utilization of error estimations of data values. *Environmetrics*, 5, 111-126, 1997.
8. T. Kawamoto, K. Hotta, T. Mishima, J. Fujiki, M. Tanaka and T. Kurita. Estimation of single tones from chord sounds using non-negative matrix factorization. *Neural Network World*, 3, 429-436, July 2000.
9. L.K. Saul and D.D. Lee. Multiplicative updates for classification by mixture models. In *Advances in Neural Information Processing Systems* 14, 2002.
10. J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. London B*, 265: 2315–2320, 1998.

# A Memory Architecture and Contextual Reasoning Framework for Cognitive Vision\*

J. Kittler, W.J. Christmas, A. Kostin, F. Yan, I. Kolonias, and D. Windridge

Centre for Vision, Speech and Signal Processing,  
University of Surrey , Guildford, GU2 7XH, UK  
[w.christmas@surrey.ac.uk](mailto:w.christmas@surrey.ac.uk)

**Abstract.** One of the key requirements for a cognitive vision system to support reasoning is the possession of an effective mechanism to exploit context both for scene interpretation and for action planning. Context can be used effectively provided the system is endowed with a conducive memory architecture that supports contextual reasoning at all levels of processing, as well as a contextual reasoning framework. In this paper we describe a unified apparatus for reasoning using context, cast in a Bayesian reasoning framework. We also describe a modular memory architecture developed as part of the VAMPIRE\* vision system which allows the system to store raw video data at the lowest level and its semantic annotation of monotonically increasing abstraction at the higher levels. By way of illustration, we use as an application for the memory system the automatic annotation of a tennis match.

## 1 Introduction

Over the last decade, the subject of visual perception has made considerable progress. Novel learning algorithms (Support Vector Machines, AdaBoost, Multiple Classifier Systems) have been developed and combined with geometric invariance, picture formation and noise models to enhance the capability of machine perception systems. These have now reached the stage of maturity that makes them deployable in constrained or semi-constrained environments to perform tightly specified visual tasks.

However, in spite of the progress, the state of the art systems which rely heavily on machine learning lack many features characteristic of human cognition, in particular the ability to understand and reason about the perceived environment, learn new concepts, develop through interaction with other systems and users, define its own perceptual goals as well as to control the perception process, and use the perceived information to support action.

One of the key requirements for a cognitive vision system to support reasoning is to possess an effective mechanism to exploit context both for scene

---

\* This work was funded by EC projects IST-34401 “VAMPIRE”, IST-004176 “COSPAL” and IST-507752 “MUSCLE”.

interpretation and for action planning. Context is conveyed by the sensory data itself, but also by the model of the imaged scene and the a priori model of the universe in which the perception system operates. By context we understand relationships between the (intrinsic and extrinsic) properties of objects appearing in the scene. Context is omnipresent and can aid vision processing at all levels. At the lowest level it is conveyed by pixels neighbouring in space and time. At high levels it pertains to objects or sets of objects that define more abstract semantic entities in a particular application domain.

Context can be used effectively provided the system is endowed with a conducive memory architecture that supports contextual reasoning at all levels of processing, as well as a contextual reasoning framework. In this paper we describe a memory architecture developed as part of the VAMPIRE vision system which allows the system to store raw video data at the lowest level and its semantic annotation of monotonically increasing abstraction at the higher levels. The memory mechanisms include forgetting processes with the rate of forgetting being inversely related to the level of interpretation. Thus the memory of the raw input data decays rapidly whereas the high levels retain the ability to provide symbolic description of the scene over relatively long term. The architecture supports forward and feedback interaction between levels.

We also describe a unified apparatus for reasoning in context. It is cast in the Bayesian framework of evidential reasoning. The use of both the memory structure and the reasoning engine is illustrated on the problem of interpreting tennis videos. We show that the contextual reasoning can be applied to the problem of foreground/background separation at the lowest level of the system, through tennis ball and player tracking, to high level reasoning about score points.

The paper is organised as follows. In the next section we develop the unified Bayesian framework for contextual reasoning. In Section 3 we summarise the proposed memory architecture. The following section (Section 4) describes the individual processing modules. Section 5 describes some experiments in which we ran the complete system on some of the material from the women's final of the 2003 Australian Open Tennis Tournament. In the final section we review the progress made on the annotation system, highlighting some of the remaining problems, and outlining the future direction of this work.

## 2 Theoretical Framework

In order to develop a general framework for interpreting video data we need to introduce an appropriate mathematical notation. Let  $v_i^t$ ,  $i = 1, \dots, N_t$  be a set of video objects to be interpreted at time  $t$ . The nature of these objects will very much depend on the interpretation task and the level of video content representation. At the lowest level the objects may be  $\dots$ , whereas at higher levels they may be regions, groups of regions, or visual events.

The inherent spatial relation of these objects is best represented in terms of an attributed relational graph  $\mathbf{G}^t = (\mathbf{V}^t, \mathbf{E}^t, \mathbf{X}^t, \mathbf{B}^t)$  where  $\mathbf{V}^t$  is a set of vertices

(nodes) constituted by objects  $v_i^t$  and  $\mathbf{E}^t$  is the set edges  $e_{ij}^t, i = 1, \dots, N_t, j \in \mathfrak{N}_i$  connecting object  $v_i^t$  with  $v_j^t$  neighbouring to  $v_i^t$ .  $\mathfrak{N}_i$  defines the index set of neighbours to node  $i$  within a particular neighbourhood system. For instance, at the lowest level, the neighbourhood system could be a 2D lattice where as at higher levels the neighbourhood system could be a general, fully connected graph.  $\mathbf{X}^t$  and  $\mathbf{B}^t$  denote unary and binary relation information characterising the nodes of the graph, i.e.

$$\begin{aligned}\mathbf{X}^t &= \{\mathbf{x}_i | i = 1, \dots, N_t\} \\ \mathbf{B}^t &= \{\mathbf{b}_{ij} | i = 1, \dots, N_t, j \in \mathfrak{N}_i\}\end{aligned}\quad (1)$$

where  $\mathbf{x}_i$  is a vector of unary attributes relating to node  $i$  and  $\mathbf{b}_{ij}$  is a vector of binary relations between objects  $i$  and  $j$ .

Each object is assumed to have a unique identity, determined by its intrinsic properties (shape, colour, texture) and extrinsic properties (pose, position, motion) that are the basis for its symbolic grounding. The measurement of these properties may be subject to sensor transfer function and imaging transformation. Let us denote the identity of object  $v_i^t$  by  $\theta_i^t$ . Then the problem of video interpretation can be formulated as one of finding the most probable labelling  $\theta_i^t = \omega_{\theta_i^t}$  of objects  $v_i^t, i = 1, \dots, N_t$ , given the measurement information conveyed by the attributed relational graphs at all the time frames up to time  $t$  as well as the interpretation of all the objects in the previous frames. Adopting a shorthand notation  $\Theta^k = (\theta_1^k, \dots, \theta_{N_k}^k)$  as a label set and  $\Omega^k = (\omega_{\theta_1^k}, \dots, \omega_{\theta_{N_k}^k})$  as the set of specific identities assumed by labels in  $\Theta^k$ , the interpretation problem can be stated

$$\begin{aligned}&\text{assign } v_i^t \rightarrow \omega_{\theta_i^t}, \forall i \text{ if} \\ &P(\Theta^t = \Omega_{\Theta^t} | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1) = \\ &= \max_{\Omega} P(\Theta^t = \Omega | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1)\end{aligned}\quad (2)$$

Note that in equation (2) the a posteriori probability can be expressed as

$$\begin{aligned}&P(\Theta^t = \Omega | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1) = \\ &= \frac{p(G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1 | \Theta^t = \Omega) P(\Theta^t = \Omega)}{\sum_{\Lambda} p(G^t, \dots, G^1, \Theta^t = \Lambda, \Theta^{t-1}, \dots, \Theta^1)}\end{aligned}\quad (3)$$

This can be further rearranged as follows:

$$\begin{aligned}&P(\Theta^t = \Omega | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1) = \\ &= \frac{p(G^t, \dots, G^1 | \Theta^t = \Omega, \Theta^{t-1}, \dots, \Theta^1) P(\Theta^t = \Omega, \Theta^{t-1}, \dots, \Theta^1)}{\sum_{\Lambda} p(G^t, \dots, G^1 | \Theta^t = \Lambda, \Theta^{t-1}, \dots, \Theta^1) P(\Theta^t = \Lambda, \Theta^{t-1}, \dots, \Theta^1)}\end{aligned}\quad (4)$$

As the denominator in (4) is independent of labelling  $\Omega$  it will not affect the decision making process and can be ignored. The contextual interpretation of the objects at time instant  $t$  is a function of the likelihood of observing measurements  $G^t, G^{t-1}, \dots, G^1$  given jointly the hypothesis  $\Omega$  as well as the interpretation of the objects in the past frames, and the prior probability of the interpretation up to time  $t$ . Under the assumption that both the measurement and label processes

are Markovian, i.e. the past history is captured by the measurements and labels at the previous time frame, we can simplify (4) as

$$\begin{aligned} P(\Theta^t = \Omega | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1) &= \\ &= \frac{p(G^t | G^{t-1}, \Theta^t = \Omega, \Theta^{t-1}) P(\Theta^t = \Omega | \Theta^{t-1}) p(G^1 | \Theta^1) P(\Theta^1)}{\sum_{\Lambda} p(G^t, \dots, G^1 | \Theta^t = \Lambda, \Theta^{t-1}, \dots, \Theta^1) P(\Theta^t = \Omega, \Theta^{t-1}, \dots, \Theta^1)} \times \\ &\quad \times \prod_{k=1}^{t-2} p(G^{t-k} | G^{t-k-1}, \Theta^{t-k}, \Theta^{t-k-1}) P(\Theta^{t-k} | \Theta^{t-k-1}) \end{aligned} \quad (5)$$

or more intuitively as

$$\begin{aligned} P(\Theta^t = \Omega | G^t, \dots, G^1, \Theta^{t-1}, \dots, \Theta^1) &= \\ &= \frac{p(G^t | G^{t-1}, \Theta^t = \Omega, \Theta^{t-1}) P(\Theta^t = \Omega | \Theta^{t-1}) P(\Theta^{t-1} | G^{t-1}, \dots, G^1, \Theta^{t-2}, \dots, \Theta^1)}{\text{norm}} \end{aligned} \quad (6)$$

which shows that the interpretation of video objects at time  $t$  is a function of the interpretation deduced at time  $t - 1$ , the probability of transition from state  $\Theta^{t-1}$  to  $\Theta^t$  and the likelihood of observing measurements  $G^t$  given measurements  $G^{t-1}$  and labels. The latter is an objective function of attributed relational graph matching which realises spatial contextual reasoning under temporal consistency constraints. If there is no temporal context (for example when performing camera calibration in our system) the problem reduces to a standard graph matching one [?, ?, 6]. Conversely if the only context is temporal, we can employ the standard techniques of Kalman or particle filters [3], or hidden Markov models [8].

The spatio-temporal Bayesian reasoning embodied by (6) provides a unifying framework for combining evidence in a wide range of video interpretation tasks at various levels of video representation. In our tennis video analysis test bed these include foreground / background separation, tennis ball tracking, player tracking, tennis court detection, event detection, and high level analysis. For each task the framework has been further developed to take into account the pertinent spatio-temporal constraints. In Section 4 we will give a more detailed account of a few of these tasks to illustrate the generality of the framework. First, the adopted memory architecture supporting the reasoning process will be described in the next section.

### 3 The Video Memory System

Working memory is effectively a capacity-limited audio-visual buffer that has been empirically demonstrated to exist in human subjects via the techniques of cognitive psychology. In visual terms its capacity is taken to be analogous to the extent and resolution of a spatial buffer, and is hence considered the location of our 'mental imagery' processing [5]. However, according to Just [4], it is possible utilise this buffer for visual processing as well as representation. Thus, it is possible in working memory to envisage sequences of geometric transformations

of relatively simple images as well as relatively complex but static images, as the current cognitive task demands. Working memory is hence a finite computational resource that is allocated on demand by a central executive.

In typical cognitive task (one such as sentence comprehension that is balanced equally between representation and relationship) short term working memory has an effective capacity of around seven distinct 'chunks' of data [7], a chunk being a pattern of audio-visual data that has been committed to long term memory. The discrete chunks in working memory are stored only very briefly, but may be recalled by the central executive instantly and in full detail (that is, with all of their associated attributes). Storage in the long-term memory, on the other hand, is primarily associative, relating very large numbers of differing items to one another in terms of their co-occurrences, rather than via their inherent attributes. Recall from the long term memory is thus a slow process, and (lacking immediate access to the attribute data) completely reliant on a sufficiency of retrieval cues related by association to the item currently under consideration in order to be appropriately recalled.

The discrete patterns represented in long term memory are hence high level abstractions of the underlying audio-visual sensory representations existing at the level of the short-term working memory, and are originally allocated on the basis of the number of times that that particular set of attributes has occurred within the working memory [1]. There is hence an inverted relationship between memory retention and interpretative level amongst human subjects, with the lowest, least generalised level of sensory memory subject to the fastest decay times (in the short-term memory), and the highest-levels of associative and relational data retained for the longest times (in the long-term memory).

In the system we have constructed, the memory system is in two separate parts: short-term and long-term, together with a graphical browser to access the memory contents. The short-term memory analyses the video, in a bottom-up fashion, making available relevant results to the long-term memory, and "forgetting" the rest. The long-term memory stores only the information that might be useful in the longer term. An analogy with human memory might be that the short-term memory is active when the system is watching the tennis match, working out what is going on, and remembering the exciting bits.

### 3.1 The Short-Term Memory

The short-term memory provides the infrastructure to run the particular set of modules selected for the application (although the short-term memory system itself is designed to be independent of the application). Thus it includes:

- self-assembly of the system
- the means of starting the modules as individual threads
- communication between the modules
- storage for the module data
- the re-use of previously computed partial results
- forgetting data that is no longer wanted

The system is run by launching a target module (or modules); this module then launches those modules needed to generate its input data, in a recursive fashion, until all of the modules needed for the application have been assembled.

### 3.2 The Long-Term Memory

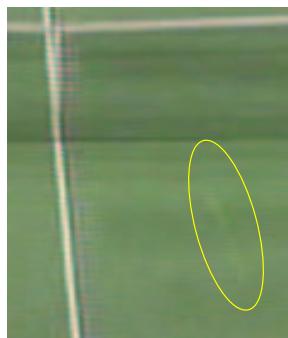
The long-term memory becomes active once the short-term memory has generated annotation, recalling exciting moments in the game, and remembering the general outcome, such as: who won, how close it was, how many sets, maybe who won each set. The long-term memory is thus relatively static once it is populated: it is the short-term memory that is actively and progressively refining the video data to extract meaning from it. Unlike the short-term memory design, it is (currently) specific to the application.

## 4 Processing Modules for the Short-Term Memory

The system is intended for use with low-quality, off-air video from a single camera (unlike for example [?]). The ball tracking in particular is a challenging task under these conditions: the movement of the tennis ball is so fast that sometimes it is blurred into the background, and is also subject to temporary occlusion and sudden change of motion direction. Two example frames are shown in Fig. 1. In Fig. 1(a), the tennis ball is over the player's head, with a size of only about five pixels. In Fig. 1(b), the tennis ball velocity is very high, and is almost completely blurred into the background.



(a) Far player serving



(b) Fast moving tennis ball

**Fig. 1.** Frames in which the ball is difficult to track

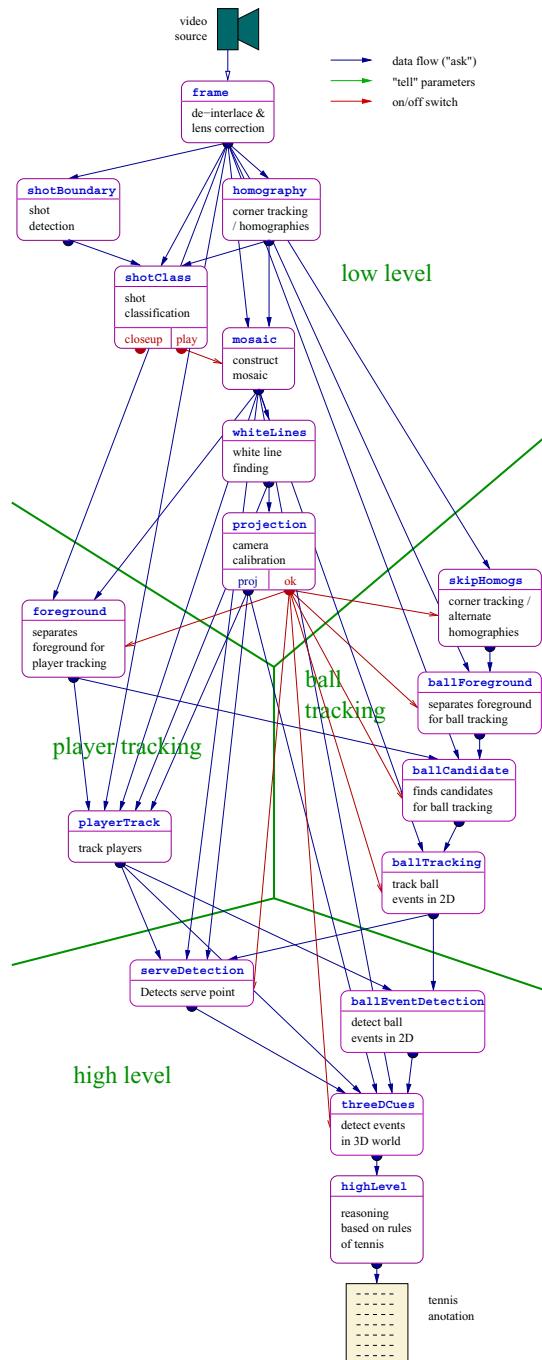


Fig. 2. Module set and data flow for tennis annotation

The complete set of modules that make up the tennis annotation system is shown in Fig. 2, which also shows the data flow between the modules. Most of the low-level modules are run on the whole video. The remaining modules are only run on play shots.

#### 4.1 Low-Level Modules

Many of these modules do not require state-of-the-art algorithms for this application. Accordingly they were mainly implemented using simple, fast, established algorithms (with exception of the mosaic module).

**Module ‘Frame’.** The ‘frame’ module reads a stream of video frames, de-interlaces them into a stream of fields and applies a preset amount of lens distortion correction.

**Module ‘Homography’.** The camera position on the court is assumed to be fixed, so that the global transformation between consecutive pairs of fields can be assumed to be a homography. The homography is found by: tracking corners through the sequence; applying RANSAC to the corners to find a robust estimate of the homography, and finally applying a Levenberg-Marquardt optimiser to improve the homography.

**Module ‘ShotBoundary’.** This module uses the colour histogram intersection between adjacent fields to detect shot boundaries.

**Module ‘ShotClass’.** A linear classifier is used to classify the shots into “play” and “non-play” using a combination of colour histogram mode and corner point continuity. Some shots are incorrectly classified (for our purposes) as “play”, such as replays. However these false positives are generally eliminated later on by the module ‘projection’, which rejects the shot if it is unable to find the court markings.

**Module ‘Mosaic’.** A mosaic is generated for each “play” shot as follows:

- The inter-field homographies are used to approximately warp each field into the mosaic coordinate system.
- The warped field is re-registered with the current mosaic to improve the homography.
- The warped fields are subsampled — typically 1 in 10 fields are retained for the mosaic.
- The mosaic is constructed by applying a median filter to each pixel of the set of these remaining warped fields.

**Module ‘WhiteLines’.** The white lines are located using a combination of edge detector and Hough transform.

**Module ‘Projection’.** At present, some basic assumptions are made about the camera position in order to label the white lines. A court model is then used to set up a set of projective linear equations in order to solve for the camera homography. If the module fails, the shot is relabelled as “non-play”.

## 4.2 Player Tracking Modules

**Module ‘Foreground’.** The foreground is determined for each field by subtracting the warped mosaic from the field, low-pass filtering the difference, and applying a threshold to the filtered difference image. The result is then filtered to reduce the noise.

**Module ‘PlayerTrack’.** Firstly an attempt is made to locate the players from the foreground images from the ‘foreground’ module. This sometimes fails, e.g. because one of the players in a short shot does not move enough to be removed from the mosaic image. If this is the case, a second algorithm is invoked, in which the shot is segmented into regions, and the dominant colour established. Candidate regions are identified that are (a) significantly different from the dominant colour and (b) in a plausible position within the image.

## 4.3 Ball Tracking Modules

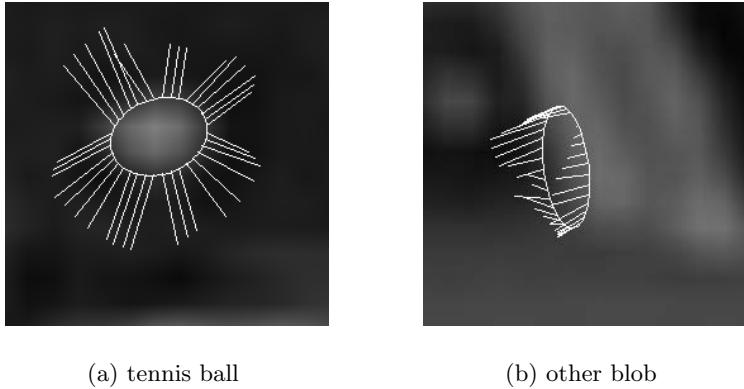
**Module ‘SkipHomogs’.** Because of the spatial aliasing present in the fields, a second homography module was implemented for the ball tracking branch, in order to generate more precise foreground separation. This module operates in a broadly similar fashion to the ‘homography’ module. The difference is that the homographies are computed between corresponding fields of consecutive frames: i.e. field  $n$  is compared with field  $n + 2$ . Thus two separate corner trackers are needed, for the odd and even fields.

**Module ‘BallForeground’.** For the  $i^{th}$  field, six fields, with sequence numbers  $i - 8, i - 6, i - 4, i + 4, i + 6, i + 8$ , are chosen as reference fields. Each motion-compensated reference field is subtracted from the  $i^{th}$  field. The subtraction results are then thresholded to get 6 foreground images. A pixel is classified as a foreground pixel only if it appears as a foreground pixel on all the six foreground images. A morphological opening operation is then applied, to further remove noise.

**Module ‘BallCandidates’.** The foreground image may contain blobs from the rackets or the players, as well as from the true tennis ball. In a single frame, the ball is distinguished by its colour and shape. However we concluded that, due to the relatively poor colour rendering in off-air video, colour was of little practical use in classifying the blobs. On the other hand, the ball shape is consistently oval (circular if the ball is stationary).

The blob shape is analysed as follows. An ellipse is fitted to the blob boundary. The image gradient on the ellipse boundary is measured. If the blob is a ball, the gradient is assumed to be normal to the ellipse and positive in the direction of the ellipse centre (Fig. 3). The blobs are then classified (using the AdaBoost method) on this basis.

**Module ‘BallTracking’.** The proposed tennis ball tracking algorithm is based on a Kalman filter and data association technique [2]. The Kalman filter works well when the linear assumption of tennis ball motion is satisfied, that is, except when the tennis ball bounces or is hit. The position, velocity and acceleration of



**Fig. 3.** Gradient vectors superimposed on candidate blobs

the tennis ball in both  $x$  and  $y$  directions are modelled in the filter state vector. To enhance the basic filter, the following measures are taken:

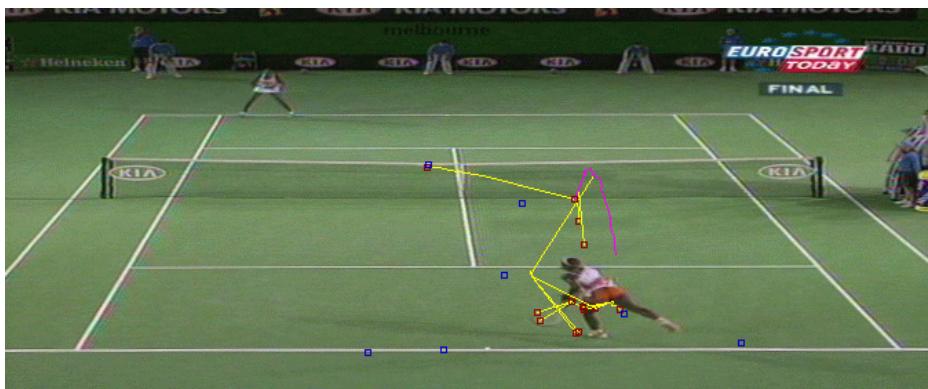
- Because the ball motion experiences a discontinuity when it bounces or is hit which is not modelled by the Kalman filter, the method permits the generation of multiple hypotheses at each new field. A “lifetime” parameter, which represents the number of “no observation fields” a track can tolerate before it is discarded, is defined for each track. The life time is associated with the cumulative likelihood of the track: the higher the cumulative likelihood, the longer the life time. Consequently, a track that is more likely to be the true trajectory has a better chance of surviving the failure of the object detection. Finally, at each observation the track that has greatest cumulative likelihood is selected (magenta track in Fig 4).
- Since the track is generally reliable before a motion discontinuity, but is less reliable afterwards, the tracking is performed in both backward and forward directions. The two tracks are merged field-by-field, on the basis of maximum likelihood.

#### 4.4 The High-Level Modules

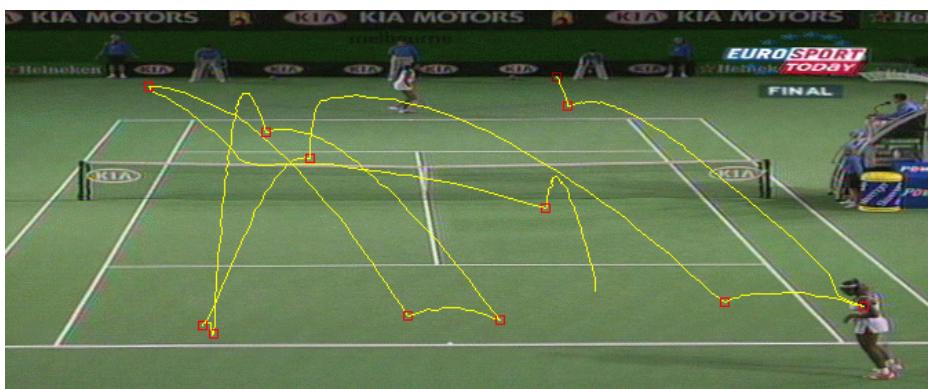
These modules combine the results from the ball and player tracking to generate an analysis of the evolution of each play shot.

**Module ‘BallEventDetection’.** The progress of the tennis match is described by key events such as: the tennis ball being hit, bouncing on the court or hitting the net. Such events can be spotted by detecting the motion discontinuities of the tennis ball. A point on the trajectory is marked as an event when the change in both orientation and motion magnitude exceed specific thresholds.

An example of the final tracking result and event detection result is shown in Fig. 5. In this example there are no false positive events; however 3 events



**Fig. 4.** Making multiple hypotheses



**Fig. 5.** Final tracking result (with event detection)

(1 bounce, 2 hits) are not detected. It is also left to higher level modules to recover from wrong decisions made here.

**Module ‘ServeDetection’.** The system performs these three operations on each player:

- process player tracking data to see whether any of the players is located in a possible serving position, and create a contour of each player,
- detect whether any of the players has the body pose we would normally associate with a serve hit,
- verify that the tennis ball is directly above the player.

The process is repeated for each field from the beginning of the shot, and terminates if, for ... of the players .. of the above hold. At this point a serve is deemed to have occurred.

**Module ‘3DCues’.** The events from the preceding modules are reinterpreted in 3D space:

- Hit / bounce discrimination is performed on all events recognised by the ball tracking module. If a player is close to the ball and the ball is located where a hit can reasonably be made, the event is labelled as a hit; otherwise it is labelled as a bounce.
- The court coordinates of the players and of ball events are projected onto the court surface model. We use the position of the players' feet, and assume that when the ball is hit it is roughly above the player.
- The events are labelled according to the region of the court in which they occurred.
- To avoid multiple instances of the same event in close succession, no event labels are permitted within 3 frames of each other.

**Module ‘HighLevel’.** The rules of the game of tennis provide us with a good guideline as to what events we will have to be capable of tracking efficiently, so as to follow the evolution of a tennis match properly. Such events would include:

- the tennis ball being hit by the players
- the ball bouncing on the court
- the players' positions and shapes (that is, body poses)

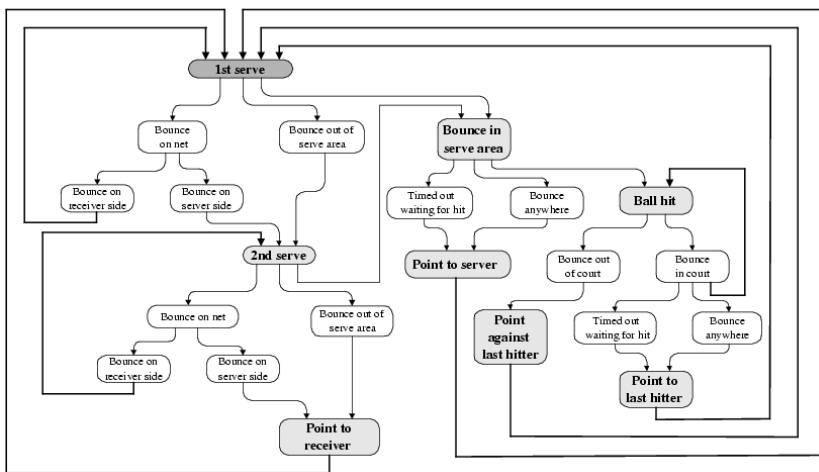
These events are used to perform reasoning about events at a higher level, like awarding a play outcome. The model for the evolution and award of a point in a tennis match is illustrated in Fig. 6(a).

As we can see, the model contains a number of loops; the state transitions drawn with **bold** lines indicate where these loops close. In order to simplify the design, we propose to replace the original scene evolution model with a set of sub-models, each one illustrating a certain scenario of the match evolution. This set of sub-graphs is illustrated in Figure 6(b).

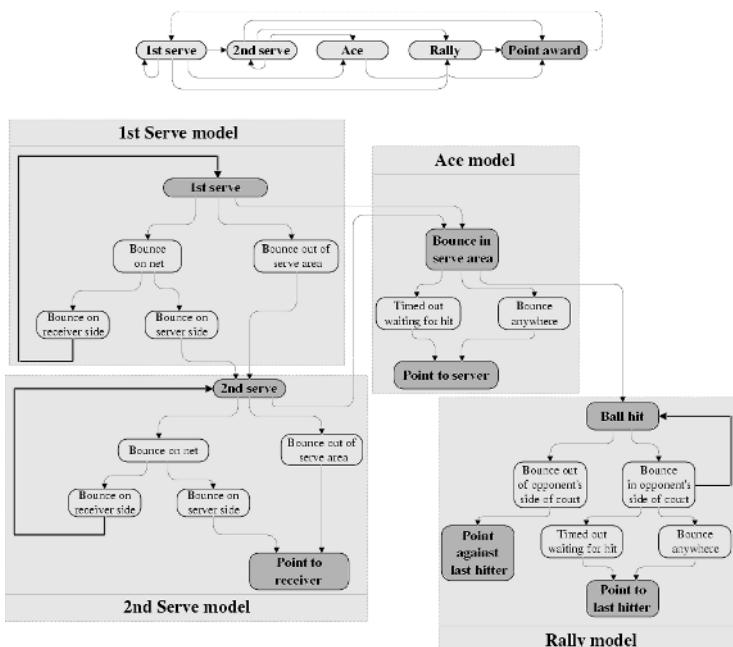
As we can see from the set of sub-models, we have opted for a more ‘perceptual’ way of selecting the set of event chains that will form the model. Moreover, choosing to decompose the initial graph down into sub-graphs and train each subgraph separately will be beneficial in other ways:

- Probabilistic reasoning tools (like HMMs) will enable correction of wrongly detected elementary events within the game, so a point can be correctly awarded even if not all events are accurately detected.
- Since the models are simpler, we can use a smaller training set.
- In some cases, we can use prior knowledge to speed up the training process. For example, statistics available for events occurring in a tennis match helps us get a very good initial estimate for the HMM parameters without the need for an explicit training process.
- It will be easier to determine which sub-models need to be improved to improve the whole system.

Since the system is dealing with ambiguous and noisy data, we are bound to encounter event detection errors in the process. Hence, another issue is that the reasoning engine should be robust to false input events. To address this, the



(a) Model for awarding a point in a tennis match



(b) Switching model and its set of sub-models

**Fig. 6.** State models for a tennis point

system used an HMM with a look-ahead decision mechanism, thereby allowing us to take into account events that occur ... the current one. The length of the look-ahead window has been limited to one event, thus allowing us to correct isolated errors. There are two reasons for this choice:

- If the length of the look-ahead window is too large, short shots with errors in them may be wrongly interpreted. That can happen as, at the end of the chain, the Viterbi algorithm will not have enough evidence to correct errors.
- Of the events used as input to the HMM, the ones most susceptible to errors are the ball bounces. However, since only one bounce can occur between two successive hits if the ball is still in play, detecting a player hit (... a serve) automatically removes the chance of a point being awarded, even if the bounce point is wrongly detected (or not detected at all).

## 5 Experiments on the Shot-Level Tennis Annotation System

The scheme described above has been tested on approximately half an hour's play from the Women's Final of the 2003 Australian Tennis Open. The sequence included a total of 45 shots that were "play" shots, all of which were correctly recognised as such.

**Ball Events.** To measure the performance of the ball tracking modules, recall and precision of the event detection are used. Recall is defined as the ratio of true positives in detected events to the ground truth events; the precision is defined as the ratio of true positives in detected events to all detected events. Since an event is a motion discontinuity point in the trajectory, it corresponds to a point in X-Y-Time 3D space. A detected event is regarded as correct when it is within  $\sqrt{3}$  pixels and 1 field of the ground truth position.

The experiments were carried out on 45 manually classified play shots, which contain 320 events. The results are as follows:

Ground truth events	Precision	Recall	Performance
320	0.85	0.81	0.65

The main cause of error is that it is sometimes difficult to detect that the ball has bounced — i.e. the deviation in the trajectory is very small. We can increase the sensitivity to such events, but at the cost of introducing false positives.

**Play Events.** If a shot contains an entire play sequence, there are 5 possible correct outcomes: no play; faulty serve by either player; point awarded to either player. The model also contains other possible situations, such as a good serve,

but where the shot terminated before the play was complete. Out of 45 play shots, 18 were awarded correct outcomes. The causes of the erroneous shot outcomes can be roughly broken down as follows:

**Table 1.** Error summary

Type of error	no. of affected shots
High-level reasoning engine	14
Ball event detection	11
Projecting events into 3D	3
Player tracking	1

These results reflect the current state of development of the system. A more detailed analysis indicated a number of problems:

- Since we do not have enough data to train the HMM parameters in the high-level reasoning, the model had to be constructed by hand, based on the author's experience as a tennis enthusiast. The model is complex, and could undoubtedly be improved.
- It can be difficult to identify ball bounce events — sometimes the deflection of the ball is barely perceptible.
- The system relies on the success of the serve detection algorithm in order to start processing a shot. If it fails, the sequence is not analysed further.
- If a shot contains more than one serve, the system only picks up the last one.
- Currently the reasoning engine gets re-initialised for each shot, thus eliminating any earlier information that could help us recover from errors encountered on the current shot.

## 6 Conclusions

We have created an integrated system that enables us to analyse tennis video material up to the level of a single shot. The system can also store partial results for subsequent experiments, thus avoiding repeated running of the time-consuming lower-level modules. The above results indicate that there is much to be done to improve the accuracy of the system, but they reflect the fact that the results come from one of the first experiments to be performed on the complete shot annotation system. Now that we have an integrated system, it is easier to identify the weaknesses, such as:

- Ball event detection: ball events can be hard to detect, even by eye. Also at present there is no explicit model of the ball events within the filter.
- Currently shots are analysed individually: combining information over wider time scales than a single shot, using higher-level models of the tennis rules, should improve the accuracy.

- The HMM interpreting the play rules in the ‘highLevel’ module is currently using as input “hard” decisions made in other modules. Using “soft” decisions, or combining decisions with confidence information, should improve matters.

However we feel that the basic approach is sound, and we emphasise that the results are preliminary. The memory model works reliably, and is readily adaptable to a wide range of cognitive tasks that require analysis at a number of different semantic levels.

## References

1. J.R. Anderson. *The architecture of cognition*. Harvard University Press, 1983.
2. Y. Bar-Shalom and T. E. Forman. *Tracking and Data Association*. Academic Press INC, 1988.
3. M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal on Computer Vision*, 1998.
4. M.A. Just, P.A. Carpenter, and D.D. Hemphill. Constraints on processing capacity: Architectural or implementational. In D.M. Steier and Mitchell T.M, editors, *Mind matters: A tribute to Allen Newell*. Erlbaum, 1996.
5. R.H. Logie. *Visuo-spatial working memory*. Lawrence Erlbaum Associates, 1995.
6. David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
7. G. A Miller. The magical number seven, plus or minus two: Some limits of our capacity for processing information. *Psychological Review*, 63:81–97, 1956.
8. L. R. Rabiner and B. H. Juang. An introduction to Hidden Markov Models. *IEEE Signal Processing Magazine*, 61(3):4–16, June 1986.

# Synthesizing the Artistic Effects of Ink Painting

Ching-tsorng Tsai<sup>1</sup>, Chishyan Liaw<sup>1</sup>, Cherng-yue Huang<sup>1</sup>, and Jiann-Shu Lee<sup>2</sup>

<sup>1</sup> Department of Computer Science and Information Engineering, Tunghai University ,  
181 Sec. 3, Taichung-Kung Rd., Taichung 407, Taiwan  
`{cttsai, liaw}@mail.thu.edu.tw`

<sup>2</sup> Department of Information and Learning Technology, National University of Tainan,  
33 Sec. 2, Shu-Lin St., Tainan 412, Taiwan

**Abstract.** A novel method that is able to simulate artistic effects of ink-refusal and stroke-trace-reservation in ink paintings is developed. The main ingredients of ink are water, carbon particles, as well as glue. However, glue is not taken into account in other researches, although it plays an important role in ink diffusion. In our ink-diffusion model, we consider the number of fibers and the quantity of glue as parameters of the structure of paper. We simulate the physical interaction among water, carbon particles, glue, and fiber mesh of paper. The realistic renderings created from our models have demonstrated that our models are successful, and are able to imitate the special artistic effects of ink painting.

## 1 Introduction

Ink painting has been in the Orient for thousands of years. It is a kind of remarkable non-photorealistic rendering; with a few strokes in gray tones, it is still able to depict its implicit spirit and meanings. The ancient ink painting techniques can be simulated and imitated by using computer software. Many researchers have demonstrated their models and results, but none of them have found a simple and yet effective model to achieve the special effects of ink-refusal and stroke-trace-reservation, that appear in ink paintings. Ink painting involves a complex interaction of its art mediums, and is very difficult to be imitated by using a simple model. Strassmann [1] applied texture-mapping techniques to simulate the diffusion of ink painting. Some [2,3,4] tried to render the silhouette with stylized strokes, while others, like Small [5] and Curtis [6] thought the ink was similar to the material used in watercolor and applied the watercolor diffusion model on ink diffusion situations. However, it takes a lot of effort to calculate, and is difficult to effective diffusion.

Guo [7] presented the diffusion speed function and classified the carbon particles in ink and the fiber density of paper for four levels. Kunii [8] constructed a multidimensional diffusion model to describe the phenomena of ink diffusion. Water spreads in the paper due to microscopic effects, but the movement of carbon particles, which are much bigger than water molecules, is based on Brownian motion. Lee [9] observed that the displacement of diffused ink front in still water agreed with the theory of diffusion proposed by Kunii. Based on Kunii's diffusion equations, Wang

[10] also took the structure of paper and gravity into account. Huang [11] divided the paper into many 2D Papels that have their own base according to the kinds of fiber. By adding the number of fibers to the base and multiplying a random number, one can calculate the absorbency of paper. Zhang [12] presented a diffusion model based on a 2D cellular automaton, a 2D grid array of tanks linked with their neighbors by pipes, computational model. He developed models of the transfer and diffusion between cells. Guo [13] defined each point of the fiber mesh as fiber structure data and capillary structure data, which includes the number of fibers and capillary tubes connected to eight neighboring points. Liquid ink flows along the capillary tubes between interlacing fibers from one point to others. All of the above researches focus on the interaction between water and carbon particles in ink, and fiber distributions in paper. However, another important factor, glue, to diffusion was neglected. The ingredients of ink are water, carbon particles, and glue. The rate of carbon to glue varies from 100:60 to 100:120, so the glue in the ink must play an important role of ink diffusion on absorbent painting paper.

This paper discusses the diffusion rendering of ink painting and focuses on synthesizing artistic effects of ink-refusal and stroke-trace-reservation, which make the ink paintings remarkable, unpredictable, and implicitly beautiful. We derive the diffusion model from the physical interaction between ink and paper. Unlike those previous researches, glue is just as important as the water, carbon, and paper, which are all being taken into account in this paper.

## 2 The Characteristics of Ink Painting

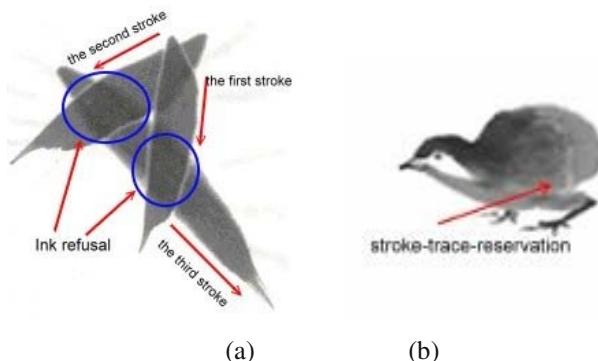
Ink painting mainly uses the ink permeation on special absorbent paper as well as skill with some strokes to express the painter's style and imagination. The ink is composed of water, carbon particles, and glue. Generally, artists use water to control the intensity of color and diffusion. More water in the ink means more diffusion and less intensity in color. Water also serves as the carrier of carbon particles. After the ink carried in the brush touches absorbent paper, carbon particles move along with water into the fiber mesh of the paper. Those particles will be stuck on the fiber mesh with glue after the water dries out; also, higher concentration of glue in ink reduces the diffusion zone of brush strokes.

There are several kinds of absorbent paper used in ink painting. All of them are thin, textured and with high absorbency, which allow the liquid ink to flow and diffuse easily. The paper, like fiber mesh, consists of lots of fibers in random positions and orientations, with some space among them. The phenomenon of ink diffusion on paper is dependent on the surface, thickness, density, impurity, kind of fiber, absorbency, and so on.

After ink is applied to the surface of paper, the space among the fibers serves as capillaries to carry ink away from its initial zone. Interactions among the ingredient molecules of the ink can also cause diffusion. When ink seeps into the fiber mesh, the carbon particles flow with water. The border of the stroke becomes blurry and feathery beauty. The diffusion stops when the amount of water left less than the space among ambient fibers.

The amount of ink permeated into paper is also related to the order of stroke in the overlap area. It is difficult for the ink of next stroke to seep into the fiber mesh if it has been filled up on the first stroke. That means the stroke cannot be covered by another stroke. If there are overlapping areas between two strokes, the second stroke doesn't usually darken that same area. That is because ink has permeated and diffused into paper mesh on the former stroke. The structure of its fiber mesh has been changed, filled up with carbon particles and glue after water dried out, and therefore, the quantity of more ink that can seep into the former stroke area is limited. This is the effect of ink-refusal. Fig.1 shows the effects of ink-refusal and stroke-trace reservation in ink-painting.

When ink permeates and diffuses in the fiber mesh on the former stroke, the water and glue molecules move faster and farther than carbon particles. Glue sticks on the fiber mesh near edge of diffusion area after water dries. There will not have more space for the ink of later stroke. Consequently, it cannot accept more ink on the border of the former stroke where later stroke applied, so a lighter border appears. This is the effect of stroke-trace-reservation. The white border will be bigger if there is enough time for the glue of the former stroke to dry. In ink painting, the first stroke looks like floating on the later overlapped ones.



**Fig. 1.** The effects of ink-refusal and stroke-trace reservation in ink-painting;(a) bamboo leaves and (b) a chicken

### 3 The Proposed Diffusion Model

In this paper, we describe that the absorbent paper is similar to grid array of fiber tanks linked with their neighbors by pipes. In our model, glue is as important as water and carbon particles. The glue affects the structure of absorbent paper after water evaporates. The interactions among fiber, water, carbon particles, and glue are modeled and all the special effects of ink-refusal and stroke-trace-reservation are well synthesized.

### 3.1 The Structures of Absorbent Paper and Ink

The absorbent paper consists of lots of fibers in random positions and directions. It is divided into many cells, and each cell relates to its eight neighbor cells. The paper cell, the basic element of paper, is called Papel, which corresponds to a pixel in rendering.  $B_i$  is the base capacity of Papel  $i$  and  $C_i$  represents the capacity of water contained in Papel  $i$ .  $G_i$  is the quantity of glue and it is set to 0 before ink starts to permeate.  $W_i$  represents the quantity of water permeated into Papel  $i$ . There is a tube connected between Papel  $i$  and its neighbor Papel  $k$ .  $TH_k^i$  stands for the minimum quantity of water molecules needed in order to diffuse from Papel  $i$  to Papel  $k$ , where  $TH_k^i = B_i + C_i + G_i$ .

It is important to construct a data structure of the absorbent paper that is based on its physical properties since different structures generate different diffusion textures. The structure of the paper is defined as:

```
structure Paper{
    int Fibers; //number of fibers
    float W; //the capacity of water permeated into papel
    float C; //the maximum capacity of water contained
              in fiber
    float I; //the quantity of carbon particle contained
    float G; //the capacity of glue
    float B; //the capacity of base
}
```

Fibers are generated by Bezier function and distributed in random orientation. The parameter Fibers is added by one if a fiber passed. The capacity of base  $B$  and the maximum capacity of water contained in fiber  $C$  can be estimated. In The parameter  $H$  represents for the thickness of the paper. The thickness of the paper and number of fibers provide space for water molecules. The thicker and more fibers, the more ink and water can be contained in Papel and therefore, diffusion zone is less for the same quantity of ink.

Besides,  $B_i$  and  $C_i$  are the quantity of base of impurity such as CaCo<sub>3</sub> and the quantity of water contained in fiber, respectively. They are not identical in different Papels because the fibers are distributed randomly. When a fiber passes Papel  $i$ ,  $\Delta B$  is subtracted from  $B_i$ . Meanwhile,  $C_i$ , the capacity of water can be held by fiber, increases the quantity of  $\Delta C$ . Therefore,

$$(B_i, C_i) := (B_i - \Delta B, C_i + \Delta C). \quad (1)$$

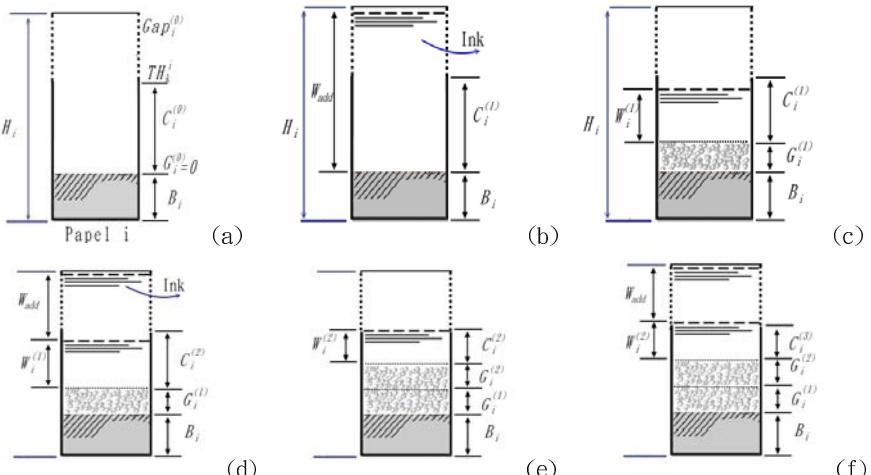
Finally, the minimum quantity of water molecules required in order to overflow on pipe can be figured out.

The quantity of ink seeped into paper when the bristles of the brush touch the surface mainly relates to the speed of the brush movement, the quantity of ink in the brush, and concentration of ink. According to this, we define the structure of ink as

```
structure Ink {
    int x,y;           //the coordinate of stroke center
    float S;           //the speed of brush movement
    float Wquantity;   //the quantity of ink in the brush
    float Ic;          //the concentration of ink
    float Gc;          //the concentration of glue in ink
}
```

### 3.2 Ink Diffusion

Fig.2 depicts the process of drying after ink seeped into Papel. The descending of carbon particles and glue change the space among fibers when water evaporates. Therefore, it affects the ink permeation of next stroke on that stroke trace. Fig. 2a shows that there is only base capacity  $B_i$  and no any carbon particles and glue in Papel i. Fig. 2b represents the condition after first stroke.  $W_{add}$ , the quantity of ink added, is more than  $TH_k^i$  and it begins to diffuse to neighbor Papels. Fig. 2c shows the situation after the first stroke diffused and dried. Glue,  $G_i(1)$ , diffused along with water and later clogs the space among fibers. Thus,  $C_i$ , the capacity of water can be held by fiber, decreases by  $\Delta C$ . Consequently,  $TH_k^i$ , the minimum quantity of water molecules needed in order to diffuse from Papel i to Papel k, changes. Fig. 2d represents when the second stroke is added. Obviously,  $W_{add}$ , the quantity of ink can be added, decreases. This is the cause of the ink-refusal effect. Fig. 2e shows that the solidified glue,  $G_i(2)$ , clogs more space after the second stroke dry. Fig. 2f shows that the third stroke is added; the capacity of the Papel,  $C_i(3)$ , decreases more and the quantity of ink that can be added,  $W_{add}$ , is even less.



**Fig. 2.** The Papel status; (a)before ink seeps into Papel, (b) when the first stroke is applied (c) after the first stroke diffused and dried, (d) when the second stroke is added,(e) the glue solidified, (f)when the third stroke is applied

## (a)The Diffusion of Water Molecular

Since water molecules are the carriers of carbon particles and glue, carbon particles and glue diffuse only whenever water molecules diffuse. Finally, the carbon particles descend into fibers and are solidified by glue after water molecules evaporate. Fibers, acting like capillaries, carry the ink into the paper mesh when the bristles of the brush touch the surface of the paper. The quantity of ink that seeps into the paper mainly depends on the quantity of ink carried by brush, the speed of brush movement and number of fibers. The ink in the bristles of the brush has plenty of time to seep into the Papel, and diffuse to its neighbor Papels if the brush moves slower. Conversely, there is less or even no ink filled into Papel if the brush moves very fast or the ink runs out in a stroke. The quantity of ink seeped, Wadd, is as following,

$$W_{add} = (H - B - W) * W_{quantity}, \quad (2)$$

where  $W_{quantity}$  is the quantity of ink carried by the brush. After the ink permeated into the fibers, the water molecules begin to diffuse to its eight neighbor Papels according to the structure of Morre Neighborhood System. The quantity of water permeated into Papel i is

$$W_i := W_i + \sum_{k=0}^{k=7} (\Delta W_i^k - \Delta W_k^i), \quad (3)$$

where  $\Delta W_i^k$  represents for the quantity of water flows from Papel k to Papel i and  $\Delta W_k^i$  is the quantity of water flows from Papel i to Papel k.  $\Delta W_i^k$  and  $\Delta W_k^i$  are defined as

$$\Delta W_i^k := \max\{0.0, 0.125 \cdot a \cdot \min[(B_k + G_k + W_k) - (B_i + G_i + W_i), (B_k + G_k + W_k) - TH_i^k]\} \quad (4)$$

and

$$\Delta W_k^i := \max\{0.0, 0.125 \cdot a \cdot \min[(B_i + G_i + W_i) - (B_k + G_k + W_k), (B_i + G_i + W_i) - TH_k^i]\}, \quad (5)$$

where  $a$  is the coefficient of diffusion,  $TH_i^k$  and  $TH_k^i$  represent the minimum quantity of water required in order to flow from Papel k to Papel i and Papel i to Papel k, respectively. The minimum quantities are calculated as  $TH_i^k := \max\{B_i + G_i, B_k + G_k + C_k\}$  and  $TH_k^i := \max\{B_k + G_k, B_i + G_i + C_i\}$ , respectively.

In real ink paintings, the water molecules evaporate gradually. Therefore, some quantity of water is subtracted in each diffusion cycle as the following formula:

$$W_i := W_i - \Delta W. \quad (6)$$

The diffusion stops when the quantity of water molecules in Papel is less than its minimum value of TH, unless there is more ink added.

(b) The Diffusion of Carbon Particles

The carbon particles diffuse along with water molecules. The more water diffused, the more carbon particles carried to their neighbor Papers. The diffusion of carbon particles is described in the following equations. Let  $I_i$  and  $I_k$  denote the quantity of the carbon molecules in Papel i and Papel k, respectively, and  $I_i$  is defined as:

$$I_i := I_i + \sum_{k=0}^{k=7} (\Delta I_i^k - \Delta I_k^i), \quad (7)$$

where  $\Delta I_k^i$  and  $\Delta I_i^k$  are the quantity of carbon particles flow from Papel i to Papel k, and flow from Papel k to Papel i, respectively.  $\Delta I_k^i$  and  $\Delta I_i^k$  are defined as:

$$\Delta I_k^i := \Delta W_k^i \left( \frac{I_i}{W_i} \right) \quad (8)$$

and

$$\Delta I_i^k := \Delta W_i^k \left( \frac{I_k}{W_k} \right). \quad (9)$$

(c) The Diffusion of Glue

The glue in the ink can be dissolved in water. It diffuses, along with water, to its neighbor Papers and then its concentration decreases. The glue sticks on fibers after the water dries. The glue contained in Papel i is

$$G_i := G_i + \sum_{k=0}^{k=7} (\Delta G_i^k - \Delta G_k^i), \quad (10)$$

where  $\Delta G_k^i$  and  $\Delta G_i^k$  represent the quantity of glue flows from Papel k to Papel i, and the quantity of glue flows from Papel i to Papel k, respectively.  $\Delta G_i^k$  and  $\Delta G_k^i$  are defined as

$$\Delta G_k^i := \Delta W_k^i \left( \frac{G_i}{W_i} \right) \quad (11)$$

and

$$\Delta G_i^k := \Delta W_i^k \left( \frac{G_k}{W_k} \right), \quad (12)$$

where  $G_k$  and  $G_i$  are the quantities of glue contained in Papel k and Papel, respectively.

The glue sticks the fibers and clogs the fiber gaps after water dry out. The ink of the latter stroke will be rejected by the stroke trace because of the effect of ink-refusal. Thus, a white border is formed on the edge of the first stroke and it appears in

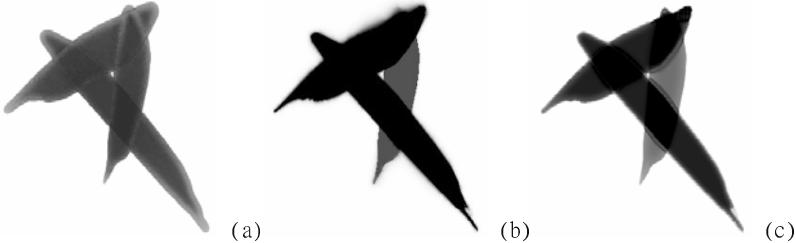
contrast if another stroke overlaid on it. The effect of stroke-trace-reservation makes the first stroke floating on the later strokes. To simulate these phenomena, we have to find the space stuck by glue,  $G_i$ , in Papel i. The formula is

$$G_i = G_i + \alpha \cdot \Delta G_i \cdot D_{rate}, \quad (13)$$

where  $Drate$  is the solidification rate of glue, and  $0 \leq Drate \leq 1$ . The larger of  $Drate$  is, the more glue solidifies.

## 4 Results

To imitate the permeation of ink and calculate the capacity of water in paper as well as the quantity of carbon particles, we input the stroke and find its boundary. The gray-scale intensity of the area inside the boundary is converted to the amount of carbon particles. The quantities of water and glue are also inputted in order for it to seep into the corresponding Papels. The first bamboo leave is loaded into the model to simulate the diffusion and the effect of ink-refusal. The drying time is approximately forty steps. To simulate the real ink painting, fifteen diffusion steps are taken as the interval between two strokes. Similarly, the second and the third strokes are inputted and simulated. Fig.1a is the image of the real ink painting using thinner ink, whose intermission is about one second. According to the intensity of these three strokes, we set the quantity of water to 1.0 and the quantity of carbon to 0.05 to imitate the real ink image. The rendering is shown in Fig. 3a which in the intersection area shown the result of ink-refusal effect as the real ink image in Fig.1a. Fig.3b and Fig.3c show that in the renderings created from the models presented by Kunii[8] and Zhang[12], no any ink-refusal effect appeared.



**Fig. 3.** The bamboo leaves;(a) our proposed rendering result, (b) Kunii's result, and (c) Zhang's result

We drew a little chicken on the absorbent paper as shown on Fig.1b. Then, we imitate the image by using our model as illustrated in Fig.4a. Compared with both Fig.4b and Fig.4c, which are created by using the models presented by Kunii[8] and Zhang[12], our rendering is more realistic and is able to present the effect of stroke-trace-reservation.



**Fig. 4.** A chicken for the effect of stroke-trace-reservation; (a) our proposed rendering result (b) Kunii's result, and (c) Zhang's result

## 5 Conclusions

In this paper we have developed a method that is able to simulate artistic effects such as ink-refusal and stroke-trace-reservation of ink painting. Our method is based on the physical interaction between ink and paper. The main ingredients of ink are water, carbon particles, as well as glue. However, glue is not taken into account in other researches except in ours, although it plays an important role in ink diffusion. We simulate the physical interaction among water, carbon particles, glue, and fiber mesh of paper. The realistic renderings created from our models have demonstrated that our models are successful, and are able to imitate the special artistic effects of ink painting that cannot be rendered from other existing models.

## Acknowledgement

This work was partially supported by National Science Council of Taiwan, R.O.C. under the Grant NSC93-2213-E-029-013.

## References

- [1] S. Strassmann, "Hairy brushes," ACM SIGGRAPH (1986) 20(3), 225-232
- [2] M.P. Salisbury, S.E. Anderson, R. Barzel, and D.H. Salesin, "Orientable texture for image-based pen-and-ink illustration," 24th annual conference on Computer graphics & interactive techniques, ACM SIGGRAPH, ACM Press (1997) 401-406
- [3] P. Litwinowicz, "Processing images and video for an impressionist effect," Proc. of ACM SIGGRAPH (1997)
- [4] K. Perlin, "An image synthesizer," Computer Graphics Proceedings, ACM SIGGRAPH, ACM Press (1985) 199-201
- [5] D. Small, "Modeling watercolor by simulating diffusion, pigment, and paper fibers," In Proceedings of SPIE (1991)
- [6] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, and D.H. Salesin, "Computer-generated watercolor," Computer Graphics Proceedings, Annual Conference Series (1997) 421-429
- [7] Q. Guo and T.L. Kunii, "Modeling the diffusion painting of sumie," Modeling in Computer Graphics (Proceedings of the IFIP WG5.10), Berlin: (1991) 329-338
- [8] T.L. Kunii, G.V. Nosovskij, and H. Takafumi, "A diffusion model for computer animation of diffuse ink painting," Computer Animation ' proceeding (1995) 98-102

- [9] S.Lee, H.Y. Lee, I.F. Lee, and C-Y Tseng, "Ink diffusion in water," European Journal of Physics (2004) 331-336
- [10] C.M. Wang, J.S. Lee, and R.J.Wang, "On realistic ink diffusion synthesis for a calligraphic learning system," International Journal of Computer Processing of Oriental Languages (2003) 16(2), 105-118
- [11] S.W. Huang, D.L. Way, and Z.C. Shih, "Physical-based model of ink diffusion in Chinese ink paintings," Journal of WSCG'(2003)
- [12] Q.Zhang, Y.Sato, J.Takahashi, K.Muraoka, and N.Chiba, "Simple cellular automaton-based simulation of ink behavior and its application to Suibokuga-link 3D rendering of trees," The Journal of Visualization and Computer Animation (1999) 10, 27-37
- [13] Q. Guo and T.L. Kunii, "Nijimi rendering algorithm for creating quality black ink painting", Int. conf. on IEEE Computer Graphics-CGI (2003)

# **Application of Spectral Information to Investigate Historical Materials**

## **- Detection of Metameric Color Area in Icon Images -**

Kimiyoshi Miyata<sup>1</sup>, Hannu Laamanen<sup>2</sup>, Timo Jaaskelainen<sup>2</sup>,  
Markku Hauta-Kasari<sup>3</sup>, and Jussi Parkkinen<sup>3</sup>

<sup>1</sup> Museum Science Division, Research Department,  
The National Museum of Japanese History,  
117, Jonai-cho, Sakura-shi, Chiba 285-8502, Japan  
[miyata@rekihaku.ac.jp](mailto:miyata@rekihaku.ac.jp)  
<http://www.rekihaku.ac.jp>

<sup>2</sup> Color Research Group, Department of Physics, University of Joensuu,  
P.O. Box 111, 80110 Joensuu, Finland  
[{hannu.laamanen, Timo.Jaaskelainen}@joensuu.fi](mailto:{hannu.laamanen, Timo.Jaaskelainen}@joensuu.fi)  
<http://spectral.joensuu.fi/>

<sup>3</sup> Color Research Group, Department of Computer Science, University of Joensuu,  
P.O. Box 111, 80110 Joensuu, Finland  
[{mhk, jussi.parkkinen}@cs.joensuu.fi](mailto:{mhk, jussi.parkkinen}@cs.joensuu.fi)  
<http://spectral.joensuu.fi/>

**Abstract.** The spectral reflectance of Icons is estimated from RGB digital images taken by a digital camera, and it is applied to detect metameric color areas in the Icons. In this paper, two detection methods are proposed and examined by using a test chart and ten Icons painted on wooden plates. The first method is based on the definition of metamericism that two stimuli can match in color while having a disparate spectral reflectance. The second method is based on a phenomenon that the variation of the color difference between two colors is changed by replacing the illuminant if the colors are metamer to each other. The experimental results can be used to consider which parts of the Icons have been repainted as restoration treatments. Despite the necessity of further consideration and improvement, the experimental results demonstrate that the proposed methods have the basic ability to detect metameric color areas.

## **1 Introduction**

Historical materials tend to be very fragile, and thus, they need to be preserved from further degradations in future. Photography of materials is one of the tools to achieve both conservation and utilization of the objects. However, information recorded in photographs is not sufficient to analyze the materials for historical research and advanced investigation purposes since they have only three color channels, which have device dependency. Spectral reflectance has a variety of objective and device-independent information on the materials. In fact, many researches have been con-

ducted on the measurement and estimation of spectral reflectance, and its application to various fields of study. Not only does spectral reflectance encourage the investigation of materials, but it also forecasts restoration treatments and natural fading processes [1]-[3]. It also provides opportunities to produce precise reproductions with highly accurate color for various lighting conditions. This study introduces methods with the use of spectral reflectance to investigate authentic historical artifacts.

In this paper, spectral reflectance is used to detect metameristic color areas in the Icons. Metamerism is a well known phenomenon of color that is defined that metameristic color stimuli are color stimuli with the same tristimulus values but different spectral radiant power distributions [4]. In the conservation and restoration of historical materials, evidence on restoration treatments, which have been applied to the objects, is important information. In such restoration treatments, color dyes or pigments having metamerism relation to the original pigments would be used. Thus, metameristic color area could tell us which parts of an object have been possibility repainted. In this study, it is examined by using color and spectral information on authentic Icons. Two detection methods will be proposed and their details will be described in the following sections.

## 2 Obtaining Spectral Reflectance of Historical Materials

There is a variety of techniques to obtain the spectral reflectance of objects that can be classified primarily into two categories – the measurement techniques with the use of measuring equipments and the estimation techniques based on some mathematical models. One of the measurement techniques was presented in our previous study [5], while an estimation technique is applied in this study.

### 2.1 Estimation of Spectral Reflectance

Many recent studies have been addressing the estimation of spectral reflectance for a variety of materials [6], [7]. However, there are problems with the measurement of spectral reflectance, including the long measurement time, sensitivity of the photo detector, low spatial resolution compared to recent digital cameras, and large quantity of measured spectral data. In order to overcome these issues, the Wiener estimation technique is introduced to obtain the spectral reflectance of materials from images taken by a digital camera having RGB color channels. In this study, this technique is applied to estimate the spectral reflectance of Icon images. The Wiener estimation matrix  $M$  is determined as follows [8]:

$$\mathbf{v} = [d_R, d_G, d_B, d_R^2, d_G^2, d_B^2, d_R d_B, d_R d_G, d_R d_G d_B]^T \quad (1)$$

$$\mathbf{R}_{rv} = \langle \mathbf{r} \mathbf{v}^T \rangle \quad (2)$$

$$\mathbf{R}_{vv} = \langle \mathbf{v} \mathbf{v}^T \rangle \quad (3)$$

$$\mathbf{M} = \mathbf{R}_{rv} \mathbf{R}_{vv}^{-1} \quad (4)$$

where vector  $\mathbf{r}$  is the measured spectral reflectance of the Macbeth Color Checker and vector  $\mathbf{v}$  consists of the sensor response  $d_R$ ,  $d_G$  and  $d_B$  including higher order terms when the Checker is taken as a digital image. Matrix  $R_{rv}$  is a cross-correlation matrix between vector  $\mathbf{r}$  and  $\mathbf{v}$ , while Matrix  $R_{vv}$  is an auto-correlation matrix of vector  $\mathbf{v}$ . The symbol  $\langle \cdot \rangle$  represents the ensemble average, and  $t$  is the transpose of the vector. Spectral image data  $f(x,y,\lambda)$  is calculated from vector  $\mathbf{v}$  and the matrix  $M$  as:

$$f(x,y,\lambda) = M\mathbf{v} \quad (5)$$

The Icons used in the estimation of spectral reflectance are shown in Figure 1. Natural pigments have been applied on wooden plates of these mid-19<sup>th</sup> century Icons. The minimum size of the Icon is approximately 240 mm in height and 165 mm in width, while the maximum size is about 430 mm in height and 330 mm in width. The images of these Icons are taken by a consumer-type digital camera.



(a) Icon No.1



(b) Icon No.2



(c) Icon No.3



(d) Icon No.4



(e) Icon No.6



(f) Icon No.7



(g) Icon No.8



(h) Icon No.9



(i) Icon No.10

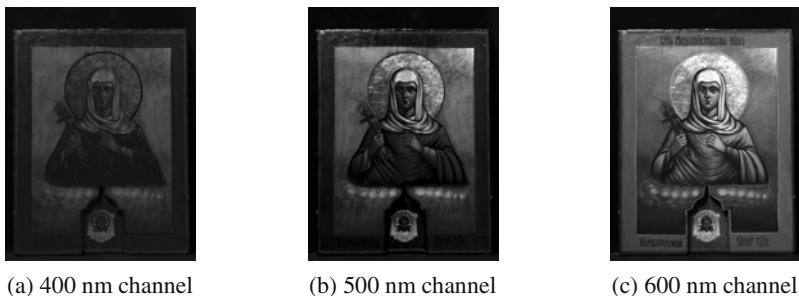


(j) Icon No.18

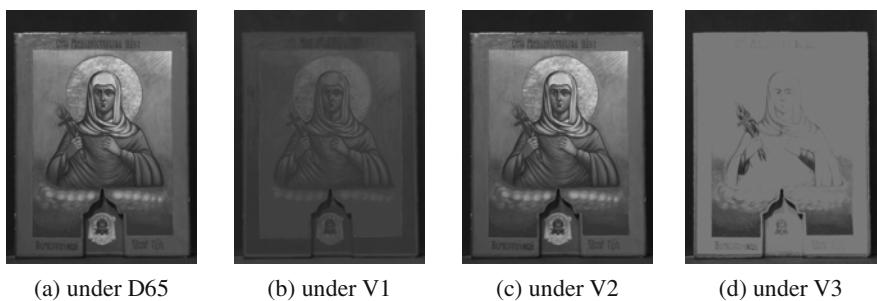
**Fig. 1.** Icons used in the study. These RGB images are taken by a digital camera

## 2.2 Applications of Spectral Reflectance

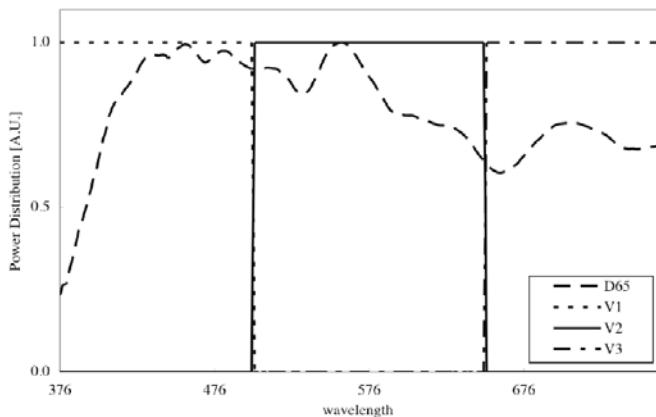
The estimated image of spectral reflectance for each Icon can be used for a wide range of applications. For example, Figure 2 shows spectral color separation, which



**Fig. 2.** Spectral channel separation using the estimated spectral reflectance



**Fig. 3.** Simulation of color changes under different illuminants shown in Figure 4



**Fig. 4.** Spectral power distributions of the D65 and three virtual illuminants, V1, V2 and V3

helps consider the types of pigments used to paint the Icons. Figure 3 illustrates the simulation of color changes under different illuminants. This simulation is useful to test the color changes of a material under various illuminants, such as those in galleries or research laboratories.

In Figure 3, (a) is a RGB image calculated from the estimated spectral reflectance image under the D65 illuminant, and (b), (c), and (d) are simulated color reproductions under three different kinds of virtual illuminants. The spectral power distributions of virtual illuminants are rectangular in shape that corresponds to the domain of short, middle, and long wavelengths as shown in Figure 4. These virtual illuminants are also used in the experiment to detect the metameristic color areas described in the next section.

## 3 Detection of Metameristic Color Area

### 3.1 Purpose to Detect Metameristic Color Area

In the investigation of historical materials, the records of restoration treatments applied to an artifact provide valuable information on how the object has been restored. This information contributes to historical researches, and further renovation and conservation of the object. Restoration records could be found by scientific or chemical analysis. However, these analyses often require fragments removed from the materials. One of the ideas to detect restoration records without any destruction to the objects is the use of color and spectral information. When historical materials were restored in the past, color dyes or pigments used in the restoration would be selected to be observed the same color to the original materials, but not identical in the sense of spectral reflectance. This is matched to the definition of the metamerism. Thus, the detection of metameristic color areas in an artifact suggests which parts of the object have been restored.

In this article, two methods to detect metameristic color area are proposed and applied to the Icons. The first method is based on the definition of metamerism that two stimuli can match in color while having disparate spectral reflectance. The second method is based on a phenomenon that the variation of color difference between two colors will be changed if a different illuminant is used and the colors are metameric each other. The details of the detection methods will be described in the following sections.

### 3.2 Proposed Method

#### 3.2.1 Method 1

The metameristic color area has the same tristimulus value but different spectral reflectance. The first detection method is based on this definition of metamerism. In Method 1, the color information is evaluated by using  $\Delta E$  and spectral information is

evaluated by using the RMSE (root mean square error). The  $\Delta E$  and  $RMSE$  are calculated by the following equations:

$$\Delta E(x,y) = \sqrt{\{L^*(x_r, y_r) - L^*(x, y)\}^2 + \{a^*(x_r, y_r) - a^*(x, y)\}^2 + \{b^*(x_r, y_r) - b^*(x, y)\}^2} \quad (6)$$

$$RMSE(x,y) = \frac{1}{N_\lambda} \sqrt{\sum_{j=1}^{N_\lambda} \{f(x_r, y_r, \lambda_j) - f(x, y, \lambda_j)\}^2} \quad (7)$$

In the equations,  $f(x, y, \lambda)$  is the estimated spectral reflectance image,  $N_\lambda$  is the number of spectral dimension,  $(x, y)$  is the pixel position, and  $(x_r, y_r)$  is the reference pixel position.  $L^*(x, y)$ ,  $a^*(x, y)$ , and  $b^*(x, y)$  correspond to CIE  $L^*a^*b^*$ . First in Method 1, we select an arbitrary pixel position in the Icon image. The selected pixel is referred to the reference pixel to detect the metamer color. The procedure of Method 1 is listed as follows:

- Step 1: Set the position of the reference pixel.
- Step 2: Observe the spectral reflectance at the pixel position, and use it as the reference spectral reflectance.
- Step 3: Check every pixel in the image whether the color difference between the current pixel and reference pixel is less than a predetermined threshold  $T_c$ .
- Step 4: Check every pixel in the image whether the RMSE between the current pixel and reference pixel is greater than a predetermined threshold  $T_{RMSE}$ .
- Step 5: Mark the pixels that satisfy both Step 4 and Step 5 as detected metamer color.

### 3.2.2 Method 2

We proposed another detection method as Method 2. As in Method 1, we first select an arbitrary pixel position in the Icon image that is referred to the reference pixel. Four kinds of illuminants shown in Figure 4 are used, and  $\Delta E_i$  ( $i = 1, 2, 3$  and  $4$ ) under the each illuminant  $i$  is calculated from Equation (8). The difference between color differences calculated from two illuminants by using Equation (9). One of them is the color difference calculated for D65 illuminant as the reference illuminant ( $i = 1$ ) in this study. The difference is compared with thresholds to determine whether the current pixel is metamer color to the reference pixel.

$$\Delta E_i(x,y) = \sqrt{\{L_i^*(x_r, y_r) - L_i^*(x, y)\}^2 + \{a_i^*(x_r, y_r) - a_i^*(x, y)\}^2 + \{b_i^*(x_r, y_r) - b_i^*(x, y)\}^2} \quad (8)$$

$$\begin{aligned} |\Delta E_1(x,y) - \Delta E_2(x,y)| &> T_2 \\ |\Delta E_1(x,y) - \Delta E_3(x,y)| &> T_3 \\ |\Delta E_1(x,y) - \Delta E_4(x,y)| &> T_4 \end{aligned} \quad (9)$$

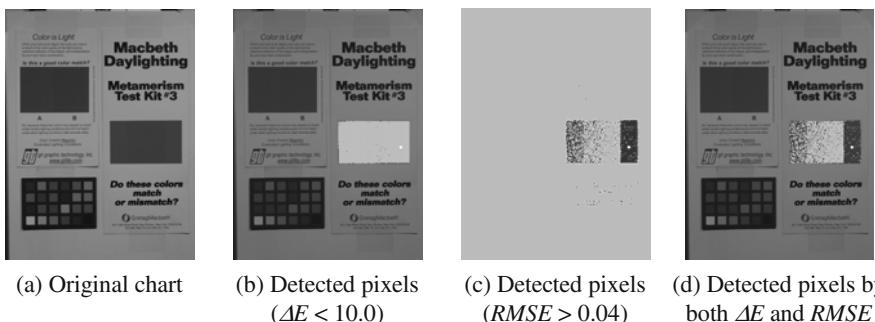
The metamer color areas have to be in small color difference. This is evaluated by using  $\Delta E$ . Three virtual illuminants are used to emphasize the change of

the color difference cause of metamerism effect because the color difference between the current and reference pixel is kept if the both pixels are in metamerism relation. All of pixels in the image are checked based on the sequence listed as follows:

- Step 1: Set the position of a reference pixel.
- Step 2: Calculate  $\Delta E_i$  for four kinds of illuminants from Equation (8).
- Step 3: Select pixels with  $\Delta E_1$  less than predetermined threshold  $T_1$  as metameric color candidates.
- Step 4: Check  $\Delta E_2$ ,  $\Delta E_3$ , and  $\Delta E_4$  for all of the candidates in Step 3.
- Step 5: Select pixels satisfying any one of the Equations (9).
- Step 6: Mark the pixels that satisfy both Step 3 and Step 5.

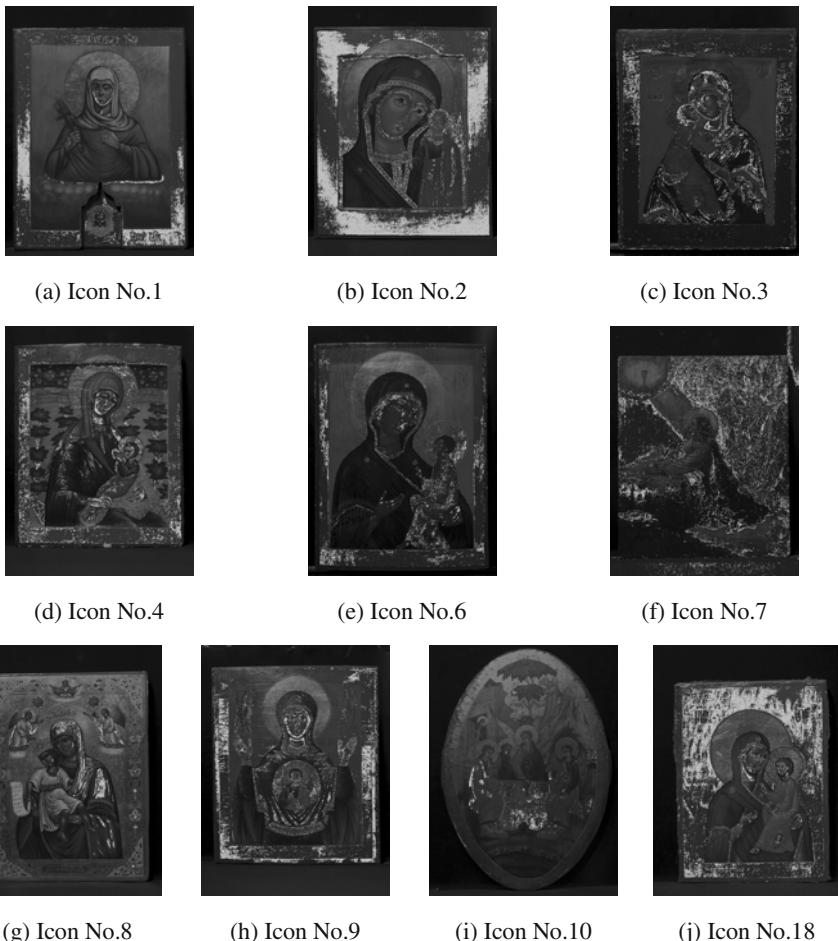
### 3.3 Experimental Results

First, Method 1 is conducted with the use of a test chart, which has metamerism color patches, as shown in Figure 5. In the figure, (a) shows the original chart, and (b) and (c) show the detected area by using only  $\Delta E$  and by using  $RMSE$  to the reference pixel respectively. The white dot in the figure is the predetermined reference pixel, and the gray area is a collection of pixels detected as metamerism color area to the reference pixel. Figure 5 (d) shows the final result of the detected metamerism color area to the reference pixel that is determined by both  $\Delta E$  and  $RMSE$ . Method 1 is also applied to the authentic Icon images, and the results are shown in Figure 6. Each Icon in the figure has a single reference pixel, which can be set at an arbitrary position in the image.



**Fig. 5.** Results of Method 1 by using a metamerism test chart

Subsequently, Method 2 is applied to the test chart and Icon images. Figure 7 (a) shows the original chart while (b) shows the detected metamerism color area. The results of the experiment on Icon images are shown in Figure 8.



**Fig. 6.** Results of the detection of metameric color area by using Method 1



**Fig. 7.** Results of Method 2 tested for the metamerism test chart



**Fig. 8.** Results of the detection of metamer color areas ( $T_1 = T_2 = T_3 = T_4 = 10.0$ )

## 4 Discussions

In both methods, a reference pixel is set first, and then metamer colors to the reference pixel are searched by pixel-wise procedure. A reference pixel is set in each Icon, and its position ( $x_r, y_r$ ) remains constant for Method 1 and Method 2. The reference pixel can be set arbitrary in the image, but only one position is used in this study to show the basic performance of the proposed methods.

In the comparison of the detection results, there are no significant differences between Method 1 and Method 2 for the metamerism chart as shown in Figure 5 (d) and Figure 7 (b). However, there are remarkable distinctions in the Icon images as shown in Figure 6 and 8. One of the reasons is that the test chart has a flat surface with the application of homogenous mixture of pigments on the colored areas, while the authentic Icons have uneven surfaces with the use of complicated mixtures of color pigments. Surface reflection property and color mixture model are necessary to en-

hance accuracy of the detection. Furthermore, methods with scientific analyses are required to conclude whether the detected metameric areas are correct.

## 5 Conclusions

In this study, two kinds of methods are proposed and tested for a test chart and authentic Icons to detect metameric color areas. The proposed methods demonstrate sufficient performance for the test chart, but further improvements are necessary for the experiments on the authentic Icons. The evaluations are discussed as follows:

- (1) Consideration of the density fluctuation and mixture of different pigments used in the painting of the Icons.
- (2) Consideration of appropriate light sources in the calculation of the tristimulus value.
- (3) Comparison between the experimental results from the proposed methods and the results of scientific analyses are necessary to conclude whether the detection by the proposed methods is reliable.
- (4) Development of combined investigation methods of measured and estimated spectral reflectance are required for more accurate investigation of materials.

## References

1. Hardeberg, J. Y., Schmitt, F., Brettel, H. H., Crettez, J., Maître, H.: Spectral Imaging in Multimedia, Proc. CIM'98 Colour Imaging in Multimedia (1998) 75-89
2. König, F., Praefcke, W.: A Multispectral Scanner, Proc. CIM'98 Color Imaging in Multimedia (1998) 63-73
3. König, F., Praefcke, W.: Practice of Multispectral Image acquisition, Proc. Electronic Imaging: Processing, Printing and Publishing in Color, SPIE 3409 (1998) 34-41
4. Wyszecki, G., Stiles, W. S.: Color Science 2nd Edition (1982) 180
5. Laamanen, H., Jaaskelainen, T., Hauta-Kasari, M., Parkkinen, J., Miyata, K.: Imaging Spectrograph Based Spectral Imaging System, CGIV 2004 -- Second European Conference on Color in Graphics, Imaging and Vision (2004) 427-430
6. Tsumura, N., Sato, H., Hasegawa, T., Haneishi, H., Miyake, Y.: Limitation of color samples for spectral estimation from sensor responses in fine art painting, Optical Review 6 (1999) 57-61
7. Nishibori, M., Tsumura, N., Miyake, Y.: Why Multispectral Imaging in Medicine?, Journal of Imaging Science and Technology 48 (2004) 125-129
8. Tsumura, N et al: Estimation of spectral reflectance from multi-band images by multiple regression analysis, Japanese Journal of Optics 27 (1998) 384-391 (in Japanese)

# An Approach to Digital Archiving of Art Paintings

Shoji Tominaga

Department of Engineering Informatics,  
Osaka Electro-Communication University,

Neyagawa, Osaka 572-8530, Japan  
[shoji@tmlab.osakac.ac.jp](mailto:shoji@tmlab.osakac.ac.jp)

[http://www.osakac.ac.jp/labs/shoji/English/index\\_e.html](http://www.osakac.ac.jp/labs/shoji/English/index_e.html)

**Abstract.** This paper describes an approach to digital archives of art paintings by considering the surface properties and the perceptual effect. A multi-band imaging system with six spectral channels is used for observing the painting surfaces. Multiple images of a painting are acquired with different illumination directions. Algorithms are presented for estimating the surface properties of surface normals, surface spectral reflectance, and reflection model parameters. All the estimates are combined for rendering realistic images of the painting under a variety of illumination and viewing conditions. Moreover, a chromatic adaptation transform is described for predicting appearance of the painting under incandescent illumination and producing the full color image on a display device. The feasibility of the method is demonstrated for an oil painting.

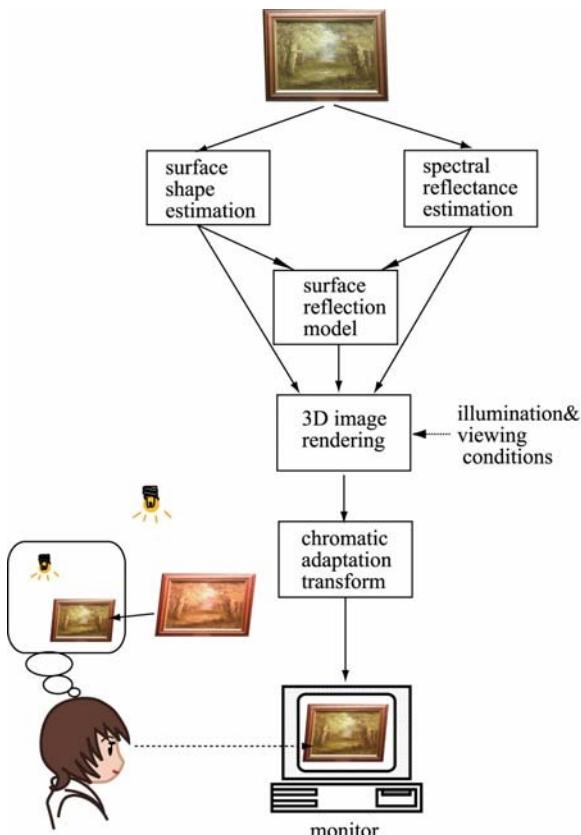
## 1 Introduction

Digital archiving of art paintings was originally based on making color-calibrated images from the paintings [1], where the surface of a painting was assumed to be a flat plane and was photographed under diffuse light sources. Practical digital archiving of art paintings including oil paintings and water paintings, however, should be based on both shape information for describing surface geometries and spectral information for color reproduction. Moreover, we have to consider human perceptual effects when the images are rendered on a display device.

A direct way to acquire the shape data of object surfaces is to use a laser range-finder [2]. This device, however, make unavoidable errors in measuring colored paintings that include specularity. Moreover laser beams may be harmful to the painting surface. Without using the 3D range data, the surface shape of a painting can be imitated using a set of surface normal vectors of small facets [3]. Next, it is well known that the spectral reflectance information is more useful than color information [4]-[5]. In fact, an RGB image is device-dependent and valid for only the fixed conditions of illumination and viewing.

Concerning the perceptual effects, we should note the viewing conditions for art paintings. Most art paintings are hung on the wall indoors, which are often illuminated with incandescent lamps. In this situation, we cannot neglect the effect of chromatic adaptation that is the most important color-appearance phenomenon of the human visual system. Colorimetric image rendering pixel-wise is not enough to produce a realistic appearance of art paintings on any display device.

The present paper describes an approach to digital archives of art paintings by considering the surface properties and the perceptual effect. Figure 1 depicts the flow for the total digital archives of art paintings. We estimate the surface properties, including surface normal, surface spectral reflectance, and reflection model parameters. A painting surface is observed with a multi-band camera system with six spectral channels. Multiple images of the painting are acquired with different illumination directions. All the estimated surface data are combined for image rendering. Moreover, the image rendered in a colorimetric way is transformed with the chromatic adaptation effect. This chromatic adaptation transform is useful for predicting appearance of the paintings under the illumination of an incandescent lamp and producing the full color images on a monitor.

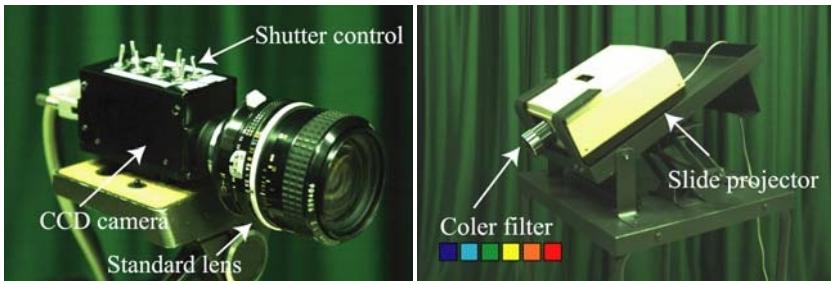


**Fig. 1.** Flow for the total digital archives of art paintings

## 2 Multi-band Measuring System

Our multi-band measuring system is decomposed into a monochrome camera and a multi-spectral lighting system, as shown in Figure 2. The camera is a CCD camera

with the image size of 1636x1236 pixels. The bit depth of the image is 10 bits. The lighting system consists of a slide projector and six color filters. Combining the camera and the light source provides a stable imaging system with six-spectral bands in the visible wavelength range of 400-700 nm. The image acquisition of the same object surface is repeated for different illumination directions. The camera aims at the object surface from vertically above the painting. The position of a light source is changed around the optical axis. In practice, multi-band images are captured under nine illumination directions.



**Fig. 2.** Multi-band imaging system with six-spectral channels

### 3 Modeling of Painting Surfaces

#### 3.1 Geometry Model

##### (A) 3D Model

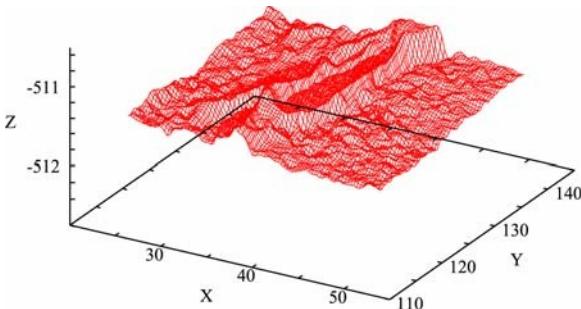
The laser range finder is used for 3D measurement of the surface shape of a painting. In this system, a horizontal stripe laser beam is emitted to the surface. This light produces a bright spot on the surface that is imaged by a calibrated CCD camera. The 3D position of the spot is then reconstructed by triangulation. For example, Figure 3 shows the surface shape constructed by regular meshes for a small area of an oil painting. The range data in coordinates (X, Y, Z) are depicted in the unit of mm. These 3D measurements by the laser range finder are available for matte surfaces with only diffuse reflection component. However this system makes errors for colored surfaces with gloss and specular highlight.

##### (B) 2D Model

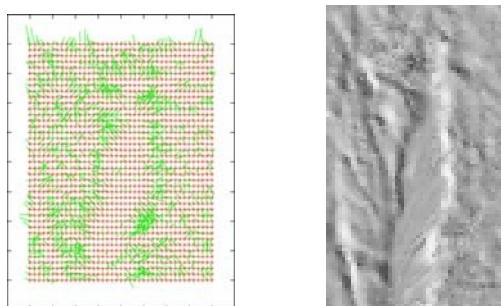
The surface of an art painting can be considered as a rough plane rather than a 3D curved surface. Therefore it is not necessary to reconstruct the 3D surface for digital archiving. An array of surface normal vectors can be used for render the image of a rough plane. The surface normal vector at each pixel point is estimated from a change in shading.

The left figure in Figure 4 illustrates the needle map of the surface normals estimated for a small area of the oil painting. The normal estimation is based on a photometric stereo, which uses the camera data at several illumination directions

without any specular reflection component. The right figure shows an image rendering the array of the surface normals.



**Fig. 3.** Surface shape by regular meshes for a small area of an oil painting



**Fig. 4.** Array of surface normals by a needle map (left) and by an image (right)

### 3.2 Reflection Model

Let us suppose an oil painting. The surface material of this object consists of a thick oil layer. Sometimes the surface is covered with vanish for the protection or decoration of the painting surface. Therefore, gloss and highlight appear on the surface. This type of object is regarded as an inhomogeneous dielectric material. Light reflected from the object surface is decomposed into two additive components, the specular reflection and the diffuse reflection.

The Torrance- Sparrow model [6] is used for describing the reflection properties of an oil painting. Then the spectral radiance  $Y(\lambda)$  from a surface is expressed as

$$Y(\lambda) = (\mathbf{N} \cdot \mathbf{L})S(\lambda)E(\lambda) + \beta \frac{DFG}{\mathbf{N} \cdot \mathbf{V}} E(\lambda), \quad (1)$$

where the first and second terms represent, respectively, the diffuse and specular components. Vectors  $\mathbf{N}$ ,  $\mathbf{V}$ , and  $\mathbf{L}$  represent the global surface normal vector, the view vector, and the illumination directional vector.  $S(\lambda)$  is the surface-spectral

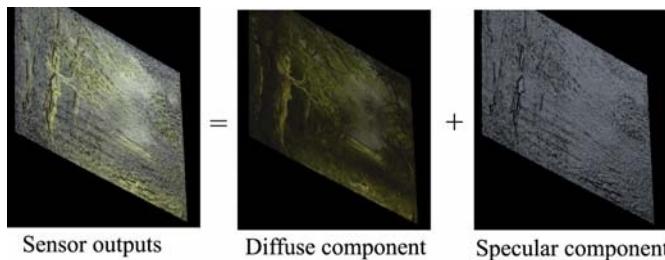
reflectance, and  $E(\lambda)$  is the illuminant spectrum. The specular component in Eq.(1) consists of three terms:  $D$  is a function providing the index of surface roughness defined as  $\exp\{-\ln(2)\varphi^2 / \gamma^2\}$ , where  $\varphi$  is the angle between the normal vector  $\mathbf{N}$  and the bisector vector of  $\mathbf{L}$  and  $\mathbf{V}$ .  $G$  is a geometrical attenuation factor.  $F$  represents the Fresnel reflectance.  $\beta$  represents the intensity of the specular component.

Therefore,  $\mathbf{N}$ ,  $S(\lambda)$ ,  $\gamma$ , and  $\beta$  are the unknown parameters to be estimated.

## 4 Estimation of Surface Reflection

### 4.1 Pixel Classification

The observed image of an oil painting by the multi-band imaging system is decomposed into two additive components of the diffuse reflection and the specular reflection, as shown in Figure 5. The surface-spectral reflectance function is estimated from the diffuse reflection component. The reflection parameters are estimated from the specular reflection component. We have devised a procedure to detect the diffuse reflection component at each pixel point of the multiple images acquired under different illumination directions.



**Fig. 5.** Decomposition of the Observed image into reflection components

### 4.2 Spectral Reflectance Estimation

The sensor outputs for a diffuse object are described as

$$\rho_k = \int_{400}^{700} S(\lambda)E(\lambda)R_k(\lambda)d\lambda + n_k, \quad (k = 1, 2, \dots, 6), \quad (2)$$

where  $R_k(\lambda)$  is the spectral sensitivity of the  $k$ -th sensor, and  $n_k$  is noise. Let  $\mathbf{p}$  be a six-dimensional column vector representing the camera outputs, and  $\mathbf{s}$  be a  $n$ -dimensional vector representing the spectral reflectance  $S(\lambda)$ . Moreover, define a  $6 \times n$  matrix  $\mathbf{H} (\equiv [h_{ki}])$  with the element  $h_{ki} = E(\lambda_i)R_k(\lambda_i)\Delta\lambda$ . Then the above imaging relationships are summarized in the matrix form

$$\mathbf{p} = \mathbf{H}\mathbf{s} + \mathbf{n}. \quad (3)$$

When the signal component  $\mathbf{s}$  and the noise component  $\mathbf{n}$  are uncorrelated, the Wiener estimator gives an optimal solution as

$$\hat{\mathbf{s}} = \mathbf{R}_{ss}^{-1} \left[ \mathbf{H} \mathbf{R}_{ss} \mathbf{H}^t + \sigma^2 \mathbf{I} \right]^{-1} \mathbf{p}, \quad (4)$$

where  $\mathbf{R}_{ss}$  is an  $n \times n$  matrix representing a correlation among surface spectral reflectances. We assume white noise with a correlation matrix  $\sigma^2 \mathbf{I}$ .

### 4.3 Model Parameter Estimation

The sensor output with the maximal intensity among the set of outputs under different illumination directions has the possibility of including the specular reflection component. To detect the specular component, we first calculate the output vector  $\mathbf{p}_D$  for the diffuse component by using the estimated reflectance  $\hat{S}(\lambda)$ . Next, select the maximal sensor output  $\mathbf{p}_M$  among all observations, and define the difference  $\mathbf{p}_S = \mathbf{p}_M - \mathbf{p}_D$ .

The specular function of the Torrance-Sparrow model is fitted to the specular data extracted from the entire pixel points. Since the specular component at any pixel has the same spectrum as the light source, the parameters are estimated based on the statistical distribution of the specular component. Define the intensity of the specular component as  $\|\mathbf{p}_S\|$ , and normalize the specular component as  $\rho_S = \|\mathbf{p}_S\| (\cos(\theta_i) \cos(\theta_r)) / (GF)$ . We minimize the sum of the fitting error

$$e = \sum_x \left\{ \rho_{Sx} - \beta D(\varphi_x, \gamma) \right\}^2, \quad (5)$$

where  $\rho_{Sx}$  and  $\varphi_x$  are the specular intensity and angle at different pixel point  $x$ . The parameters  $\gamma$  and  $\beta$  minimizing the error are solved as a solution of the nonlinear fitting problem.

## 5 Image Rendering

### 5.1 Prediction of Chromatic Adaptation

There are many kinds of chromatic-adaptation and color-appearance models (e.g., see [7]-[9]). These models consist of complicated equations with many parameters. Therefore, the previous models are difficult to apply to the problem of image rendering of art paintings in the present study.

We have proposed a color prediction method for incomplete chromatic adaptation that is based on an extended version of the von Kries model using correlated color temperature scale [10]. We use Illuminant D65 as reference and Illuminant A as test. The adaptation process is considered incomplete adaptation along the color temperature scale. Then, an incomplete adaptation index  $d$  ( $0 \leq d \leq 1$ ), representing the

degree of chromatic adaptation, is introduced on the color temperature scale between the test illuminant A and the reference illuminant D65.

Because the color temperature scale in *kelvin K* is not correlated to perceived color differences, we use a *reciprocal megakelvin* temperature scale. The unit of this scale is the reciprocal megakelvin ( $MK^{-1}$ ), and a given small interval in this scale is approximately perceptible. Let  $[MK^{-1}]_T$  and  $[MK^{-1}]_R$  be the reciprocal temperatures of the test illuminant and the reference illuminant, respectively. Then, the color temperature of adaptation illumination corresponding to the index  $d$  is determined as

$$T_d = 10^6 / (([MK^{-1}]_R - [MK^{-1}]_T)d + [MK^{-1}]_T). \quad (7)$$

A proper value of the adaptation index  $d$  is determined on matching experiments between real paintings under Illuminant A and the images on the monitor.

## 5.2 Rendering Algorithm

Images of the target paintings are created using the estimated surface normals and spectral reflectances at all pixel points, and the above determined reflection model. A ray-casting algorithm is adopted for the image rendering under parallel rays from the light source.

For the purpose of accurate color image rendering, the color images of art paintings are not represented by RGB values, but represented by the CIE tristimulus values XYZ. We calculate the XYZ values at each pixel by using the spectral radiance  $Y(\lambda)$  and the CIE standard color-matching functions.

Therefore, the image of an art painting under a specified illuminant condition is represented as an array of the tristimulus values XYZ. The color values of each pixel are then transformed by taking the chromatic adaptation effect into account. A computational procedure for this color transformation is given in Ref. [10]. It uses a von Kries-type transformation. The entire transformation process is summarized as

$$\begin{bmatrix} X_p(D) \\ Y_p(D) \\ Z_p(D) \end{bmatrix} = \mathbf{M}^{-1} \mathbf{W} \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (8)$$

where  $\mathbf{M}$  is a 3x3 transformation matrix into the cone responses, and  $\mathbf{W}$  is a diagonal matrix with the elements of gain coefficients determined by the incomplete adaptation index  $d$ .

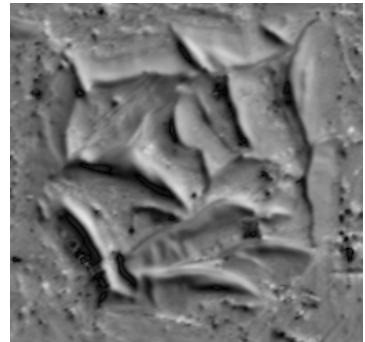
## 6 Application Results

Figure 6 shows the oil painting of “Flowers.” The surface of this painting is covered with a transparent oil vanish. This object was measured with the six spectral channels in the image size of 1070x 1503 pixels. The surface normals at all pixel points were estimated using the photometric stereo method to the image intensity data by the diffuse reflection. Figure 7 shows an image rendering the estimated surface normals for

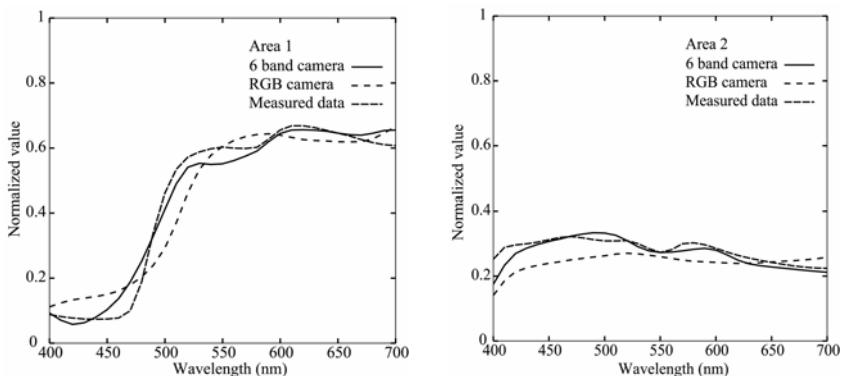
the rectangular area of “Flowers” in Figure 6. We can see big roughness at the petals. These results are much more precise than the measurement by the laser range finder.



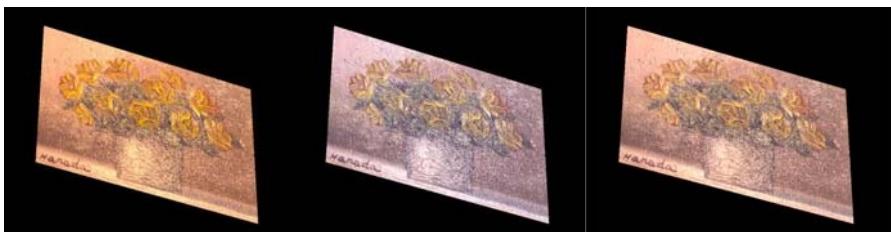
**Fig. 6.** Oil painting “Flowers”



**Fig. 7.** Images of the estimated surface normals



**Fig. 8.** Estimation results of surface-spectral reflectances for Area 1 and Area 2



**Fig. 9.** Image rendering results. (Left: colorimetric rendering under an incandescent illumination, Middle: image rendered under a fluorescent light source of D65, Right: Image rendered with the incomplete chromatic adaptation effect under an incandescent illumination)

Next, the surface spectral reflectances were estimated at all pixel points. Figure 8 shows the estimation results for Area 1 and Area 2 shown by white rectangles in Figure 6, where the bold curve, the dashed curve, and the broken curve represent, respectively, the estimates by the present method, the direct measurements, and the estimates by the previous method using an RGB camera with three spectral channels. We can see that the present method using the multi-spectral imaging system recovers the detailed shape of spectral reflectance.

We have executed the visual experiment for predicting incomplete chromatic adaptation to several oil paintings. This experiment was conducted by using the method of memory matching between the target painting under a real Illuminant A and the images on a calibrated monitor. The most proper prediction was performed with the index value of about  $d=0.5$ .

All the estimates of the surface properties were combined for creating computer graphics images of the oil painting under different illumination and viewing conditions. Figure 9 shows the image rendering results. The left picture represents the image under an incandescent light source that is based on the colorimetric rendering without any chromatic adaptation effect. The middle picture represents the image under a fluorescent light source with the color temperature of D65. The right picture represents the image created with the incomplete chromatic adaptation effect.

## 7 Conclusion

The present paper has described an approach to digital archives of art paintings by considering the surface properties and the perceptual effect. A multi-band imaging system with six spectral channels was used for observing the painting surfaces. We have presented algorithms for estimating the surface properties from the image data, which includes surface normals, surface spectral reflectance, and reflection model parameters. All these estimates were combined for rendering realistic images of the original painting under a variety of illumination and viewing conditions.

We have described a chromatic adaptation transform for predicting appearance of the painting under incandescent illumination and producing the full color image on a display device. The incomplete adaptation index representing the degree of chromatic adaptation was defined on the color temperature scale. The von Kries' framework was extended to the algorithm of incomplete chromatic adaptation.

This paper has shown the feasibility of the proposed method for oil paintings. As discussed in 3.2, the surface of a water painting is rough and reflects no specular component that is essentially different from any oil painting. We have found that the surface of a water painting is not Lambertian. The digital archiving problem for water paintings is left for a future work.

## References

- [1] K. Martinez et al.: Ten years of art imaging research, *Proceedings of the IEEE*, vol. 90, No. 1, 2002.
- [2] S. Tominaga and N. Tanaka: 3D Recording and Rendering of Art Paintings, *Proc. Ninth Color Imaging Conf.*, pp.337-341, 2001.

- [3] N. Tanaka and S. Tominaga: Measuring and Rendering Art Paintings Using an RGB Camera, *Proc. of EUROGRAPHICS*, pp.299-306, 2002.
- [4] H. Maitre et al.: Spectrophotometric image analysis of fine art paintings, *Proc. Fourth Color Imaging Conf.*, pp.50-53, 1996.
- [5] Y. Miyake, et al.: Development of multiband color imaging systems for recording of art paintings, *Proc. SPIE: Color Imaging*, Vol.3648, pp.218-225, 1999.
- [6] K.E.Torrance and E.M.Sparrow: Theory for off-specular reflection from roughened surfaces, *J. of Optical Society of America A*, Vol.57, No.9, pp.1105-1114, 1967.
- [7] Y. Nayatani: A Simple Estimation Method for Effective Adaptation Coefficient, *Color Res. Appl.* Vol.22, pp.259-268, 1997.
- [8] M. D. Fairchild : *Color Appearance Models*, Addison-Wesley, 1998.
- [9] N. Moroney, M. D. Fairchild, R. W. G. Hunt, C. Li, M. R. Luo and T. Newman: The CIECAM02 Color Appearance Model, *Proc. Tenth Color Imaging Conf.*, pp.23-27, 2002.
- [10] Tominaga, M. Nakagawa, and N. Tanaka: Image Rendering of Art Paintings, *Proc. Twelfth Color Imaging Conf.*, pp.70-75, 2004

# Preferential Spectral Image Quality Model

D. Kalenova, P. Toivanen, and V. Bochko

Spectral Image Processing and Analysis Group, Laboratory of Information Processing,  
Department of Information Technology, Lappeenranta University of Technology,

P.O.Box 20, 53851 Lappeenranta, Finland

{Diana.Kalenova, Pekka.Toivanen, Vladimir.Botchko}@lut.fi

**Abstract.** In this paper a novel method of spectral image quality characterization and prediction, preferential spectral image quality model is introduced. This study is based on the statistical image model that sets a relationship between the parameters of the spectral and color images, and the overall appearance of the image. It has been found that standard deviation of the spectra affects the colorfulness of the image, while kurtosis influences the highlight reproduction or, so called vividness. The model presented in this study is an extension of a previously published spectral color appearance model. The original model has been extended to account for the naturalness constraint, i.e. the degree of correspondence between the image reproduced and the observer's perception of the reality. The study shows that the presented preferential spectral image quality model is efficient in the task of quality of spectral image evaluation and prediction.

## 1 Introduction

The nature and the scope of imaging have been undergoing dramatic changes in the recent years. The latest trend is the appearance of multiprimary displays and printers that reproduce images with a closer spectral match to the original scene captured [1,2]. Appearance of such devices gives rise to a problem already existing for conventional tools - assessment of perceptual image quality given only physical parameters of the image. The demand for a quantitative analysis of image quality has dramatically increased. Preferential spectral image quality model, presented in this work is intended to create a paradigm that would allow description of the quality of spectral images in terms of objectively measurable parameters of spectral images in connection with the subjective quality metrics. The preferential spectral image quality model, presented in this paper, can be used for spectral image quality evaluation and prediction in tasks of e.g. imaging device production and calibration, printing industry and in some other industrial applications.

The model described in this paper is based on the results of a previously published spectral color appearance model. The model, introduced in [3], has been used for color image quality estimation, with a color image, reproduced through the spectral image. Spectral color appearance model has demonstrated that there is a close connection between the quality judgments of the observers and the parameters of the model, i.e. vividness and colorfulness, which in turn have been proven to depend upon statistical characteristics of spectral images, in particular, standard deviation and

kurtosis. This corresponds with the results obtained by other researchers [4]. Fedorovskaya and de Ridder in [4] suggest that scaling of the perceived strength of colorfulness impression is linearly related to the average and standard deviation of the chroma variations in CIELUV color space.

In every set of images produced through variation of both parameters of the spectral color appearance model, one image has been found to be of maximal quality, meaning that it had the highest quality judgment given by the observers. Moreover, there has been found a significant difference between the quality judgments of the scenes, meaning that part of the images exhibited a clear maximum at points close to the original scenes, whilst the others have had a significant shift in the highest quality judgment position. This model has already proven to be effective in the task of image quality evaluation, however it lacks universality, in a sense, that units of quality used are artificial and have weak mathematical basis, which, in turn, does not allow comparison with analogous reference systems. Another serious drawback is that modeling of a combined effect of the parameters of the model on the overall quality impression has been reproduced via Fuzzy Inference System, and has resulted in serious error rates in some cases [3].

In general, the spectral color appearance model and the preferential spectral image quality model, introduced in this paper, can be attributed to a class of preferential quality models. A number of publications exist on the topic of preference in color and tone reproduction in the framework of image quality investigation [5, 6, 7]. Among the whole range of preferential characteristics, contrast, saturation and memory color reproduction are the most common ones. These features change is evident in the image and is highly dependent upon the observer and the content of the scene. Normally, such image attributes have an optimal value where the overall impression is most pleasing [7].

Spectral color appearance model has been created upon the assumption that colorfulness and vividness can efficiently describe image quality, with colorfulness including both contrast and saturation. Previous research is extended in this work to account for memory color reproduction or as it will further be called naturalness. The naturalness constraint imposed upon the image quality stems from an intuitive assumption that high quality images should at least be perceived as “natural”. At base, this assumption rests on the psychological model of the quality judgment constitution. Accordingly, the impression of an image is formed as a result of comparison of the output of the process of visual perception and the internal representation of an object, which, in turn, is based on the previous experience (i.e. memory representation or prior knowledge of reality). That is, a high quality image complies with the ideas and expectations of the scene captured in the image at hand. Several works exploring the influence of naturalness on color image quality exist at the moment [4, 8], particularly, in the field of color photography. A direct dependence between the naturalness constraint and the quality judgments has been experimentally found in these, with memory colors being relatively consistent among different observers. However, for the case of colorfulness variation a discrepancy between the naturalness judgments and the perceived quality has been found, i.e. observers perceived more colorful images as being of higher quality, at the same judging these images as unnatural. This phenomenon can be explained from the information-processing point of view, a high degree of naturalness is a necessary, but not a sufficient condition for

quality perception, a usefulness condition has to be satisfied as well, which, in turn, leads to a discriminability principle. In other words, a highly saturated image is perceived to be of high quality, despite being unnatural, due to an increased possibility of discerning certain features in an image [4, 8]. In this study we are trying to establish a connection between quality judgments of spectral images, spectral image attributes and the naturalness constraint with regard to the principles mentioned.

## 2 Statistical Model

A generalized statistical model, characterizing the behavior of statistical characteristics of natural spectral images  $\mathbf{f}(\mathbf{x})$ , presented as n-dimensional vector random field, is described by the following equation [9]:

$$\mathbf{f}(\mathbf{x}) = \boldsymbol{\mu} + \mathbf{D}\mathbf{g}(\mathbf{x}) \quad (1)$$

where  $\mathbf{x}$  is a vector with each element being spatial dimension;  $\boldsymbol{\mu}$  is a mean vector,  $\mathbf{g}(\mathbf{x})$  is a normalized vector image with zero mean and unit standard deviation for each component,  $\mathbf{D}=\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ , where  $\sigma_i$  is standard deviation in the component [3]. The following parameters:  $\alpha$ ,  $\beta$  and  $k_{\max}$  are used for modifying the colorfulness and vividness change in the experiment.

Vector  $\boldsymbol{\sigma}$  is presented in the following form:

$$\boldsymbol{\sigma} = \alpha\beta\boldsymbol{\sigma}_v + (1-\alpha)\boldsymbol{\sigma}_c \quad (2)$$

where  $\alpha = (\sigma_{\max} - \sigma_{\min})/\sigma_{\max}$  is the relationship between constant and variable parts of standard deviation, affecting the saturation of colors in an image, and  $\beta$  is a contrast variation coefficient,  $\boldsymbol{\sigma}_v$  is a variable component vector of  $\boldsymbol{\sigma}$ ,  $\boldsymbol{\sigma}_c$  is a constant component vector of  $\boldsymbol{\sigma}$  [3].

$\mathbf{g}(\mathbf{x})$  is defined through gamma-Charlier histogram transform of  $\mathbf{f}_s(\mathbf{x})$  and a kurtosis vector  $\mathbf{k}$  as follows [3]:

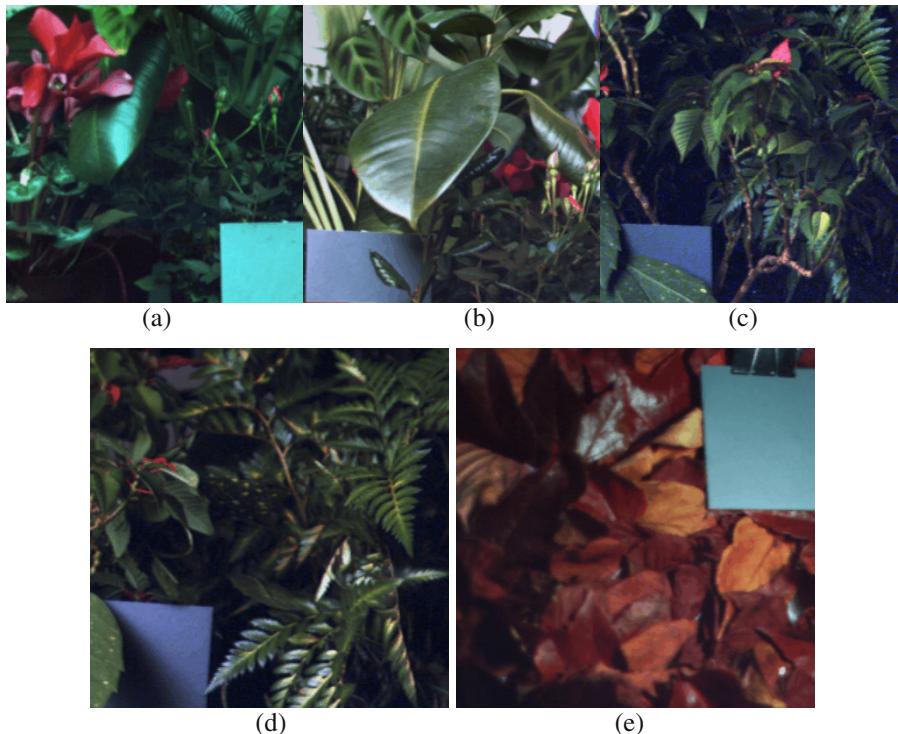
$$\mathbf{g}(\mathbf{x}) = \mathbf{H}(\mathbf{f}_s(\mathbf{x}), \mathbf{k}) \quad (3)$$

where  $\mathbf{f}_s(\mathbf{x})$  is a normalized image of  $\mathbf{f}(\mathbf{x})$ , with zero mean and unit standard deviation for each component. To affect the image appearance through histogram transform, all kurtosis elements are proportionally modified according to the given maximum of the kurtosis value  $k_{\max}$  [3].

The task of quality manipulation is a complicated task that requires significant computational resources. Spectral images contain large amounts of information, which have to be manipulated in order to influence the overall impression of the display. Usually, some implicit assumptions are made in order to limit the amount of computations. The assumption underlying this study is that only global variations are taken into account, which, in turn, originates from the fact that all parts of the image have been captured under the same illuminant or belong to the same object. Thus the same modifications are applied to all pixels of the image irrespective of the content [4]. Based upon this principle the generalized statistical model is applied to spectral images in this study.

### 3 Experiment

Experiments were performed on spectral images of natural scenes from [10]. Five images – *inlab1*, *inlab2*, *inlab5*, *jan13am* and *rleaves* were selected (see Fig.1). Images have the following dimensions: 256x256 pixels, and 31 spectral components per each pixel. For the purpose of the experiments the area of 128x128 pixels were selected. Images were captured by a CCD (charge coupled device) camera in a 400-700 nm wavelength range at 10 nm intervals. The images selected were taken indoor (in a controlled environment, i.e. dark-lab or glass-house).



**Fig. 1.** Color reproduction of original spectral images used in the experiments (a) *inlab1*, (b) *inlab2*, (c) *inlab5*, (d) *jan13am*, (e) *rleaves*

Experimental settings, which include the number and the criteria for selection of observers, test stimuli, instructions and viewing conditions requirements, were chosen to comply with [11]. According to this standard relative quality values should be obtained from at least ten observers and three scenes, and all of the observers have to be tested for normal vision and visual acuity. Thus, we chose twenty observers to participate in the experiments. They had normal or corrected-to-normal vision without color deficiencies. To prevent the loss of quality of judgments due to fatigue the duration of the experimental sessions was limited to one hour, in case of more time needed the experiments continued after a break. Viewing conditions followed the

requirements given in [12]. Therefore, the general room lighting was shaded and was set so that it neither directly nor indirectly influenced the viewing surface. When viewing slides, the frames were darkened to a 10% brightness level for a width of 75 mm.

Although original images were presented, it should be emphasized that they were not explicitly identified to observers as such. First, a set of test images was produced using the colorfulness parameter. The term includes both contrast and color saturation. Thus, colorfulness was varied through standard deviation, using Eq.2. By changing  $\alpha$  and  $\beta$  coefficients it was possible to receive new values for constant and variable parts of standard deviation. This procedure was applied to the images with values of  $(\alpha, \beta)$  equal to  $(0.55, 1)$ ,  $(0.75, 1)$ ,  $(1, 1.3)$ ,  $(1, 1.6)$  consequently. The second set of tests was produced through variation of the vividness parameter, closely related to highlight reproduction in an image. As the highlight was modified through kurtosis change, the test images were produced with the help of Eq.3 (with  $k_{max}$  equal to 5, 10, 30, 60). The effect of the change of both of parameters on the overall appearance had been shown in [3]. Both test sets were presented to the subjects, who had to rate the naturalness of the images on a ten-point numerical category scale ranging from one (unnatural) to ten (real life). The following instructions for the experiments were given to the observers [4]:

"You will be presented a series of images. Your task is to asses the naturalness of images, using an integer number from one to ten, with one corresponding to the lowest degree of naturalness and ten to the highest. Naturalness is defined in this case as the degree of correspondence between the image reproduced on the screen and reality, i.e. the original scene as you picture it."

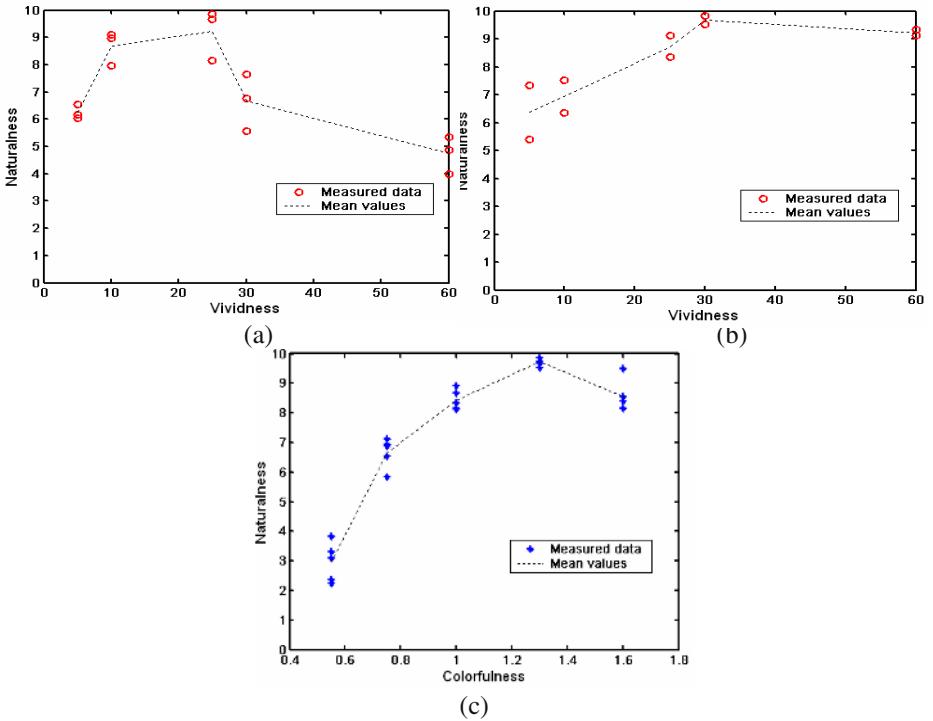
The results of the tests are given in Table 1, where each cell corresponds to an averaged naturalness evaluation score, with outliers being excluded from consideration. The columns denoted as 1 present results of the tests produced through the colorfulness change, and 2 with the vividness change respectively. In Table 1 Image 1 and Image 2 have parameters  $(\alpha, \beta, k_{max})$ :  $(0.55, 1, 5)$ ,  $(0.75, 1, 10)$ , Image 3 is the original, Image 4 and Image 5 have respectively parameters  $(\alpha, \beta, k_{max})$  equal to  $(1, 1.3, 30)$ ,  $(1, 1.6, 60)$ . Note that either  $(\alpha, \beta)$  (for colorfulness change) or  $k_{max}$  (for vividness change) were varied, while the rest of the parameters were kept constant.

**Table 1.** Mean values of naturalness evaluation scores

Quality	Image1		Image 2		Image 3		Image 4		Image 5	
	1	2	1	2	1	2	1	2	1	2
Inlab1	2.37	5.41	5.83	6.34	8.12	8.34	9.67	9.83	8.15	9.12
Inlab2	2.25	6.03	6.93	7.98	8.93	9.67	9.87	5.56	8.41	3.98
Inlab5	3.84	7.34	6.55	7.53	8.34	9.12	9.85	9.53	8.56	9.34
Jan13AM	3.10	6.56	7.12	8.96	8.67	9.87	9.73	6.76	8.17	4.87
Rleaves	3.32	6.17	6.86	9.10	8.16	8.17	9.54	7.65	9.50	5.35

Looking at Table 1 we can state that the peaks of the naturalness judgments do not lie within the original image area, which in turn brings us to a conclusion that users generally prefer slightly modified images. Fig. 2 illustrates the connection between

the naturalness constraint and statistical parameters of the spectral images varied at the experiments.



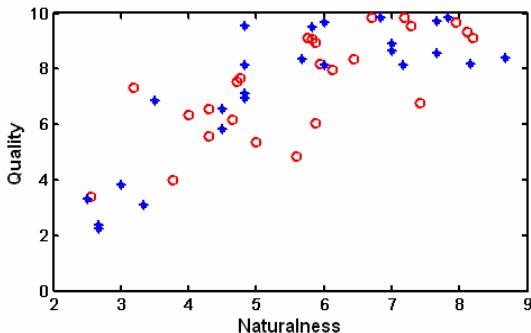
**Fig. 2.** Averaged naturalness estimations vs. statistical parameters of spectral images. Vividness (a) of *inlab2*, *jan13am*, *rleaves*; vividness (b) of *inlab1* and *inlab5*; colorfulness (c) of all of the images

Fig. 2 demonstrates that there is positive correlation between the attributes of spectral images and the naturalness constraint. Fig. 2 (a) and (b) specifically present the relation between the vividness spectral image attribute and the naturalness constraint. It is clearly visible that the naturalness maximum lies at points close to the original image. We have separated the vividness parameter versus naturalness plots onto two parts due to a different form of dependency between the two. Fig. 2(a) contains a plot of the *inlab2*, *jan13am*, *rleaves* image judgments, and Fig. 2(b) - *inlab1* and *inlab5*. In the first case images exhibit a sharper drop in the naturalness judgments than in the second one, in fact, in the second case the decrease in naturalness is such that the naturalness remains approximately close to the maximal value. Such discrepancy in the image judgments has also been obtained when evaluating the quality of the images [3]. Both of the phenomena can be attributed to a fact that images *inlab1* and *inlab5* contain objects that attract the most of the observers' attention, compared to the objects situated at the background. Moreover, these objects lie in the red area of the spectrum, which assumes that observers are not

susceptible to minor variations in these areas due to the properties of the human visual system. Thus, the drop in quality and in naturalness is less definitive.

Fig. 2(c) demonstrates the connection between the colorfulness parameter and the naturalness constraint. It can be stated that observers perceive slightly more colorful than original images as being the most natural ones. This effect is consistent with the results obtained in the experiments with color images, stating that there is a tendency for memory colors to be slightly more saturated compared with actual object colors [13]. Moreover, considering the fact that observers have previously rated the images with higher colorfulness values as being of higher quality [3] we can state that memory color reproduction influences the preferred color reproduction of the objects [14].

Another important characteristic of image naturalness is correlation with the quality perception. For this purpose the quality judgment values have been taken from the previous study [3] and plotted against the naturalness obtained in this study. Fig. 3 demonstrates a plot of the quality judgments versus the naturalness constraints for both vividness (red circles) and colorfulness (blue asterisk) test sets. Such a comparison is possible due to the fact that experimental settings (number of observers, number and contents of scenes, viewing conditions, etc.) are similar in both of the experiments, moreover the algorithm of modification and values of the statistical parameters of spectral images are the same.



**Fig. 3.** Averaged naturalness estimations vs. perceptual quality estimations of images with the vividness (red circles) and colorfulness (blue asterisk) change

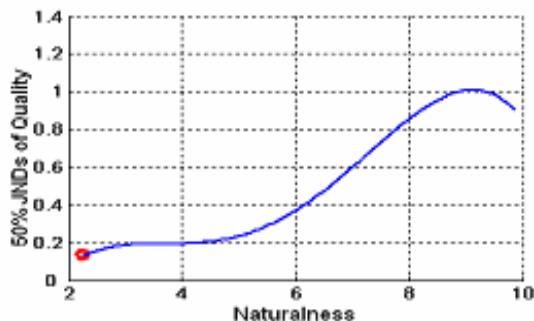
Fig. 3 illustrates the interdependence between the quality judgments and the naturalness constraint. It is clear that there is a strong correlation between these notions. In fact, the correlation coefficient between the quality judgments and naturalness estimations computed over all of the scenes equals to 0.8196 for colorfulness, 0.7018 in the case of vividness test sets, and the overall correlation is equal to 0.7752.

With the increase of naturalness the quality also increases, which proves the preliminary assumption made, stating that in order for the image to be of good quality it should at least be perceived natural. The spread in the plot can be attributed to a lack of test images and a rough scale of spectral image attributes accepted in this

study. However, even such a small test set was enough to prove a connection between the attributes of spectral images, naturalness and overall image quality impression.

Even though we can see from the plot in Fig. 3 that there is a connection between the naturalness constraint and the quality judgments of the users, it is relatively difficult to predict what would be the effect of the naturalness change on image quality, and how fast does the quality decrease with the decrease in naturalness, which in turn can be varied through variation of any of the attributes of spectral images. Accordingly naturalness could serve as a universal image attribute that would allow modeling both image quality and the joint effect of attributes of spectral images on the overall perception of the image reproduction. In order to model the effect of naturalness on quality a preference distribution function of naturalness expressed in terms of JNDs has been constructed. According to [11] JND is a stimulus difference that yields a 75%:25% proportion in a forced-choiced paired comparison. In other words, JND is the smallest possible difference between two stimuli, noticeable by observers. The standard distinguishes two types of JND units, attribute and quality JNDs. In our study we have used the quality JNDs, which is a measure of the significance of perceived differences on overall image quality. Essentially all of the observers could detect the difference and identify which of the samples have had higher naturalness.

The preference data was obtained with the use of the rank order technique. The observers were presented small sets of stimuli (5 images at a time) and had to rank the images according to the quality of these. The result of such experiments is noisier than in a paired-comparison technique, however it significantly reduces sample handling. To construct the quality preference function value of each of the attributes of the highest rated image was identified for each scene and observer, and the fraction of times each position had been preferred was computed. To convert each fraction to a probability density, it was divided by the width of an attribute interval including the sample position. The resulting preference distribution function is presented in Fig. 4.



**Fig. 4.** Averaged preference distribution function of naturalness

A conclusion conforming to the preliminary assumption imposed upon the naturalness constraint can be drawn based on Fig. 4: with the increase of naturalness the quality of the images increases. However, at a certain point naturalness starts to decrease. Thus, high degree of naturalness is a necessary, but not a sufficient

condition for quality perception. A usefulness condition has to be satisfied as well, which in turn leads to a discriminability principle. Meaning that, even if the image gives an impression of being unnatural, it might be perceived as being of high quality, due to the fact that the information in the image is easily discriminable. Therefore, naturalness has a strong connection with the statistical characteristics of spectral images and image quality perception on the whole, being a necessary, but not a determinative factor.

## 4 Conclusions

In this paper, a preferential spectral image quality model has been presented. The model sets a relationship between the statistical characteristics of spectral images, overall quality, and perceived naturalness. The model described in this paper is an extension of a previously published spectral color appearance model. The original study has been extended to account for the naturalness constraint, i.e. the degree of correspondence between the image reproduced and the observers' perception of the reality.

Several conclusions can be drawn upon this study. One of the important inferences drawn from this work is that a strong connection between the statistical parameters of spectral images and the naturalness perception does exist. Particularly, not only there is a strong correlation between the colorfulness parameter, but it can also be said that there is a tendency for memory colors to be slightly more saturated compared with actual object colors, meaning that observers perceive slightly more colorful than original images to be more natural [13]. Moreover, considering the fact that observers generally discern more colorful images as being of higher quality, it can be stated that memory color reproduction influences the preferred color reproduction of the objects [4].

The connection between the vividness parameters and naturalness is twofold. Part of the images used in the experiment exhibit a sharper drop in the naturalness judgments than the rest, in fact, in the second case the decrease in naturalness is such that the naturalness remains approximately close to the maximal value. A similar phenomenon has been found in the spectral color appearance model [3] concerning image quality judgments. Both of the phenomena can be attributed to a fact that images *inlab1* and *inlab5*, that exhibit an insignificant drop in both of the characteristics compared with the rest of the images, contain objects that attract the most of the observers' attention, in comparison with the objects situated at the background. Moreover, these objects lie in the red area of the spectrum, which assumes that observers are not susceptible to minor variations in these areas due to the properties of the human visual system. Thus, the drop in quality and in naturalness is less definitive.

The last conclusion is the connection between the naturalness constraint and the overall perceived image quality. Although with the increase in naturalness the quality of the images increases, naturalness of the image is a necessary, but not a sufficient condition for the high quality judgments. A usefulness condition has to be satisfied as well. Thus the peak of the quality judgments does not lie at the highest naturalness value, meaning that observers knowing that the image is unnatural would still

perceive the image as being of high quality. For the purpose of modeling the naturalness influence upon image quality a preference distribution function, describing the impact of the naturalness on the quality judgments in terms of JNDs has been constructed. The function can be used for spectral image quality prediction in terms of image naturalness.

In general, both the preferential spectral image quality and the spectral color appearance models can be attributed to a class of preferential image quality models and can serve as an efficient tool of image quality characterization and prediction.

## References

1. Hardeberg, J. and Gerhardt, J.: Characterization of an Eight Colorant Inkjet System for Spectral Color Reproduction, in Procs. Second European Conf/ on Colour Graphics, Imaging and Vision, Aachen, Germany (2004) 263-267.
2. Rosen, M., Hattenberger, E. and Ohta, N.: Spectral Redundancy in a 6-ink Inkjet Printer, in Procs. of The Dig. Phot. Conference, Rochester, NY, USA (2003) 236-243.
3. Kalenova, D., Botchko, V., Jaaskelainen, T. and Parkkinen, J.: Spectral Color Appearance Modeling, in Proc. Dig. Phot. Conference, Rochester, NY, USA (2003) 381-385.
4. Fedorovskaya, E.A., de Ridder, H. and Blommaert, F.J.J.: Chroma Variations and Perceived Quality of Colour Images of Natural Scenes, *J. Color res. and appl.* 22 (1997) 96-110.
5. Buhr, J.D. and Franchino, H. D.: Color Image Reproduction of Scenes with Preferential Tone Mapping, U.S. Patent #5 447 (1995) 811.
6. de Ridder, H.: Saturation and Lightness Variation in Color Images of Natural Scenes, *J. Imaging Sci. and Techn.* 6(40) (1996) 487-493.
7. Janssen, R.: Computational Image Quality, (2001) 20-35.
8. de Ridder, H.: Naturalness and Image Quality: Saturation and Lightness Variation in Color Images of Natural Scenes, *J. Imaging Sci. and Techn.* 40 (1996) 487-498.
9. Botchko, V., Kälviäinen, H. and Parkkinen, J.: Highlight Reproduction Using Multispectral Texture Statistics, Proc. Third Int. Conf. on Multispectral Color Science, Joensuu, Finland (2001) 61-65.
10. Parraga, A., Brelstaff, G. and Troscianko, T.: Color and Luminance Information in Natural Scenes, *J. of Opt. Soc. of America A* 15 (1998) 3-5.
11. ISO/DIS 20462-1, Psychophysical Experimental Method to Estimate Image Quality – Part 1: Overview of Psychophysical Elements, International Organization for Standardization (2003).
12. ISO 3664, Graphic Technology and Photography: Viewing conditions, International Organization for Standardization (2000).
13. Newhall, S. M., Burnham, R. W. and Clark, J. R.: Comparison of Successive with Simultaneous Color Matching, *J. of Opt. Soc. of America* 47 (1957) 43-56.
14. Siple, P., and Springer, R. M.: Memory and Preference for the Color of Objects, *Perception and Psychophysics* 33 (1983) 363-370.

# Three-Dimensional Measurement System Using a Cylindrical Mirror

Yuuki Uranishi, Mika Naganawa, Yoshihiro Yasumuro, Masataka Imura,  
Yoshitsugu Manabe, and Kunihiro Chihara

Graduate School of Information Science, Nara Institute of Science and Technology,  
8916-5 Takayama Ikoma, Nara, Japan

{yuuki-u, naganawa, yasumuro, imura, manabe, chihara}@is.naist.jp  
<http://chihara.naist.jp/>

**Abstract.** We propose a novel method for measuring a whole three-dimensional shape of an object with a simple structured system. The proposed system consists of a single CCD camera and a cylindrical mirror. A target object is placed inside the cylindrical mirror and an image is captured by the camera from above. There are two or more points that have the same origin in the captured image: one is observed directly, and the other is observed via the mirror. This situation means that a point is observed by a real camera and virtual cameras at the same time. Therefore, the three-dimensional shape of the object can be obtained using stereo vision. We simulated an actual experimental situation and measured the three-dimensional shape of the object from the simulated image, and the results have demonstrated that the proposed method is useful for measuring the whole three-dimensional shape.

## 1 Introduction

Whole three-dimensional shape measuring and model reconstructing have a wide area of applications, including virtual museums [1] and digital archiving of relics [2]. Most of whole three-dimensional models are built from multiple range images. Registration and integration of the images are cumbersome procedures. Therefore, it is highly desirable that a whole shape of the object is easily measured in a single shot. The three-dimensional measurement methods are classified into two groups: one is an active measurement method, and the other is a passive measurement method.

The active measurement method, such as Time-of-Flight [3] and Structured Lighting [4], obtains a shape of the target object by projecting a light or some kind of energy and by measuring its reflection. This method provides a precise shape of the object as a dense point cloud. However, it is impossible to obtain the whole shape data at once due to the limitation of the viewpoint. Therefore, in order to reconstruct the whole shape data, measurement from multiple viewpoints is required by moving a turntable or the sensor itself.

There are several existing methods for passive measurement. For example, Visual Hull [5][6] constructs approximation of a whole object shape from silhou-

ettes observed from multiple viewpoints using a volume intersection technique. This method is fast and robust enough to implement real-time applications. Although increasing the number of cameras improves the accuracy of an approximation of the actual object shape, a complete multi-camera calibration becomes inevitable. The accurate three-dimensional shape of the object can be measured using stereo vision [7], which is one of the passive methods. The geometric structure is acquired from images shoot from different viewpoints based on triangulation. It is difficult to force all cameras to be calibrated and synchronized. To overcome the problem, a catadioptric stereo system is proposed [8][9]. The catadioptric stereo system consists of a single camera and one or more mirrors. Placing mirrors produces multiview observations that are included in a single shot image. There is the one that comes directly into the lens and there are those that are reflected by the mirrors. Conventional catadioptric stereo system involves a complex arrangement of mirrors and the object. Therefore, it becomes more difficult to arrange the system appropriately for whole shape measuring.

In this paper, we propose a simple structured method for measuring a whole three-dimensional shape with a single CCD camera and a cylindrical mirror. We have previously proposed a concept of our system [10]. An image of the target object, which is placed inside the cylindrical mirror, is captured from above. The captured image includes sets of points observed from multiple viewpoints. The three-dimensional shape of the object can be obtained using stereo vision. The proposed method is suitable for measuring the shape of moving objects such as insects and small animals, because the proposed method can measure a whole three-dimensional shape of the object in a single shot. In this study, we demonstrate the proposed method in a simulated experiment using a circular cone as a target object.

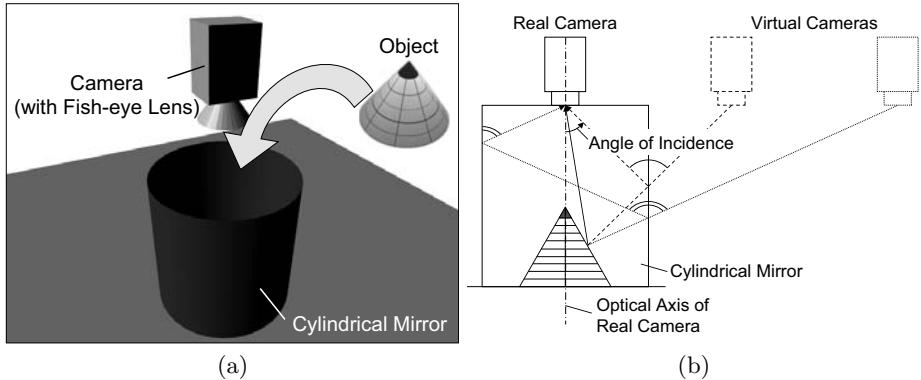
## 2 Proposed System

### 2.1 System Overview

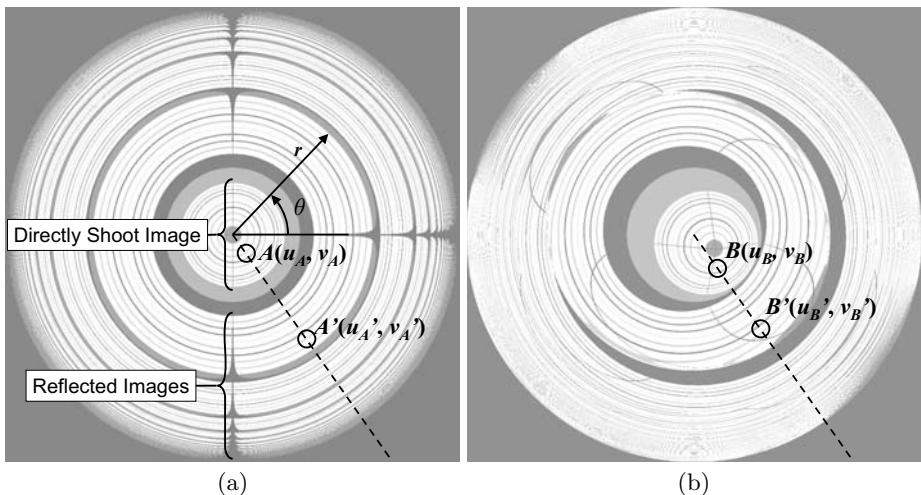
The proposed system consists of a CCD camera to which the fish-eye lens is attached and a cylindrical mirror whose inside is coated by a silver reflective layer. Figures 1 (a) and (b) show a bird-eye view and a side view of the proposed system, respectively. A target object is placed in the cylindrical mirror, and an image is captured by the CCD camera from right above. As shown in Fig. 1 (b), each point of the object can be measured from different directions: one is a direct observation, and the others are the observations via the mirror. This situation implies that a given point is observed by a real camera and virtual cameras at the same time.

### 2.2 Captured Image

Figures 2 (a) and (b) simulate the captured images by the proposed system. The object is set on the center of the cylindrical mirror in Fig. 2 (a) and off the center



**Fig. 1.** A schematic overview of the proposed system. (a) The system consists of a CCD camera with fish-eye lens and a cylindrical mirror. A target object is placed inside the cylindrical mirror, and an image is captured by CCD camera from above. (b) Each point of the object can be observed directly and via the mirror. This situation implies that a point is observed by a real camera and virtual cameras in a single shot



**Fig. 2.** The images captured by the proposed system in case of setting a circular cone on the center of the mirror (a) and off the center of the mirror (b). The area around the center of the image is a directly shoot image, and the peripheral area is a reflected image on the cylindrical mirror. When the optical axis of the camera is identical with the center axis of the mirror, the original point  $A$  ( $B$ ) and the reflected point  $A'$  ( $B'$ ) are always placed on the same line through the center of the cylindrical mirror

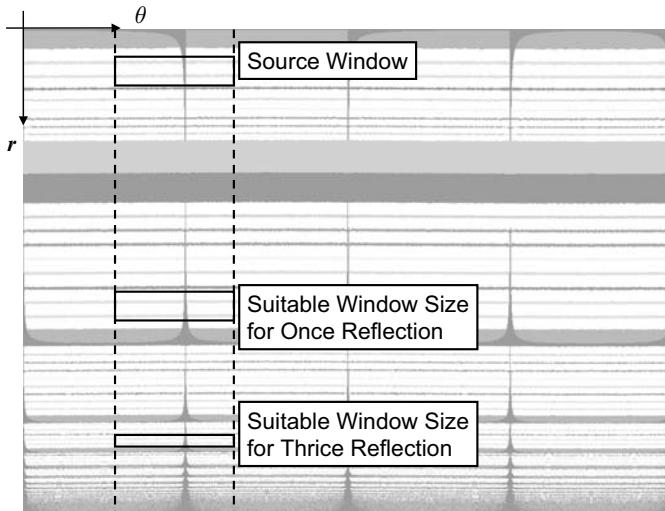
in Fig. 2 (b). The area around the center of the image is a directly shoot image, and the other area is a reflected image on the cylindrical mirror. When the optical

axis of the camera is identical with the center axis of the mirror, the original point  $A(u_A, v_A)$  ( $B(u_B, v_B)$ ) and the reflected point  $A'(u'_A, v'_A)$  ( $B'(u'_B, v'_B)$ ) always lie on the same line through the center of the cylindrical mirror, which does not depend on the position of the object. Consequently, the search area for corresponding points is reduced to a line. The proposed system has a small calculation cost and yields small mismatches between the original points and the reflected points.

### 2.3 Searching Stereo Pair

The set of corresponding points that have same origin should be estimated to measure three-dimensional shape. Sum of Square Difference (SSD) is employed in this system. First, the captured images are represented in a polar coordinate system. Next, the area of the windows for SSD is normalized because the size of the window is changed dynamically according to its position in the image. Then, the set of points are searched using SSD. The detail of each step is described below.

**Coordinate Conversion into Polar Coordinate System.** A coordinate system of the captured image is converted into a polar coordinate system to facilitate calculating SSD values. Figure 3 is the image converted from Fig. 2 (a). The vertical and the horizontal axes in Fig. 3 correspond to a radius  $r$  and an angle  $\theta$  in Fig. 2 (a), respectively.



**Fig. 3.** The captured image Fig. 2 (a) in a polar coordinate system. The coordinate system of the captured image is converted into the polar coordinate system. The vertical and the horizontal axes in the converted image correspond to the radius  $r$  and the angle  $\theta$  in the original image, respectively

**Stretching the Target Region.** The vertical length of the reflected image depends on the angle of incidence and a normal direction of the surface. As shown in Fig. 3, the vertical length changes according to the number of reflection. For example, the vertical length of the thrice reflected image is smaller than that of the once reflected image in Fig. 3. The optimal choice of the window size for SSD cannot be derived analytically, because the normal direction at each point of the object is unknown. Therefore, the combinations of the various window size are prepared and examined. The selected regions should have a same size for the calculation of SSD values. The target region is stretched to the same size as the source region. A converted luminance of R, G, and B,  $l_i(u, v)$  ( $i = r, g, b$ ), is represented as follows:

$$v_{\text{int}} \equiv \text{trunc} \left( \frac{v}{s} \right), \quad (1)$$

$$\Delta \equiv \frac{v}{s} - v_{\text{int}}, \quad (2)$$

$$l_i(u, v) = (1 - \Delta)o_i(u, v_{\text{int}}) + \Delta o_i(u, v_{\text{int}} + 1) \quad (i = r, g, b), \quad (3)$$

where  $s$  is a ratio of the vertical length of the target region to that of the source region, and  $o_i$  ( $i = r, g, b$ ) is the original luminance of each color at  $(u, v)$ , and  $\text{trunc}$  is a function that truncates a number to an integer.

**Calculating SSD.** SSD value  $d_{ST}$  between the source region  $S$  and the target region  $T$  is described as

$$\begin{aligned} lld_{ST}(u, v) = \sum_{u, v} & \left\{ (l_r^S(u, v) - l_r^T(u, v))^2 + (l_g^S(u, v) - l_g^T(u, v))^2 \right. \\ & \left. + (l_b^S(u, v) - l_b^T(u, v))^2 \right\}, \end{aligned} \quad (4)$$

where  $l_i^S(u, v)$  and  $l_i^T(u, v)$  ( $i = r, g, b$ ) are luminance of color at the source region and the target region, respectively. The set of points that provides the smallest SSD value is selected as an optimal set.

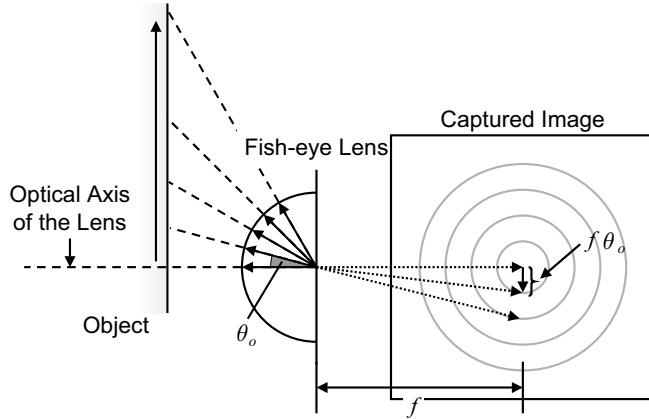
## 2.4 Estimating the Position of the Virtual Cameras

The position of the virtual cameras is estimated from the captured image. Figure 4 shows a relationship between an angle from an optical axis of the lens and a distance from a center of the captured image. In the proposed system, the distance from the center of the image  $d$  is defined as

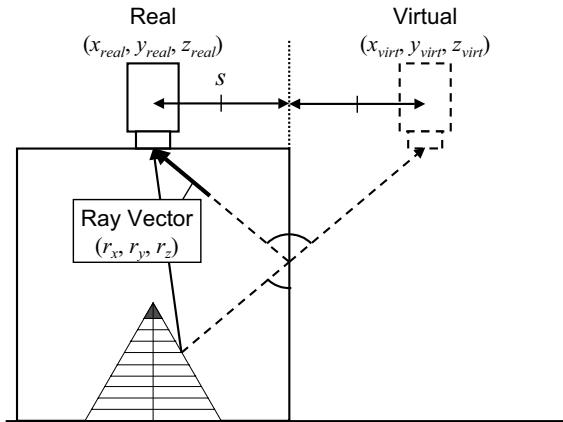
$$d = f\theta_o, \quad (5)$$

where  $f$  is a focal length of the lens and  $\theta_o$  is an angle from the optical axis of the lens.

A position of a virtual camera is determined by the angle  $\theta$ , a radius of the cylindrical mirror and a number of reflection. Figure 5 shows how a camera position is determined. Assuming that the position of a real camera is the point



**Fig. 4.** A relationship between an angle from an optical axis of the lens and a distance from a center of the image. The distance from the center of the image is proportional to the angle from the optical axis. Such a projection is called as an equidistance projection



**Fig. 5.** Estimation of the position of the virtual camera. A position of a virtual camera can be estimated using Eqs. (6), (7) and (8)

$C_{\text{real}}$  at  $(0, y_{\text{real}}, 0)$  and the height of the virtual camera is same as that of the real camera, the position of a virtual camera that corresponds to the  $n$ th reflection is represented as

$$x_{\text{virt}}^n = (-1)^{2n-1} (2nR \cos \theta), \quad (6)$$

$$y_{\text{virt}}^n = y_{\text{real}}, \quad (7)$$

$$z_{\text{virt}}^n = (-1)^{2n-1} (2nR \sin \theta), \quad (8)$$

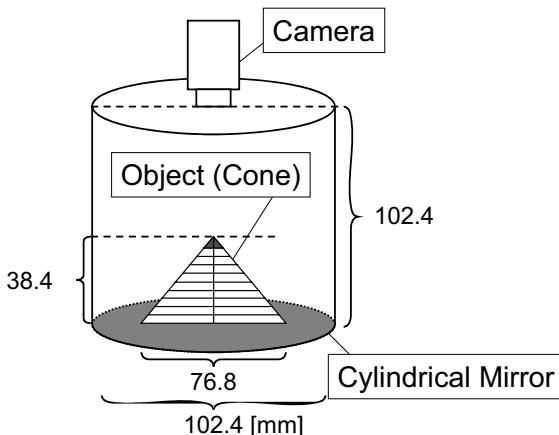
where  $R$  is a radius of a cylindrical mirror.

### 3 Experimental Results

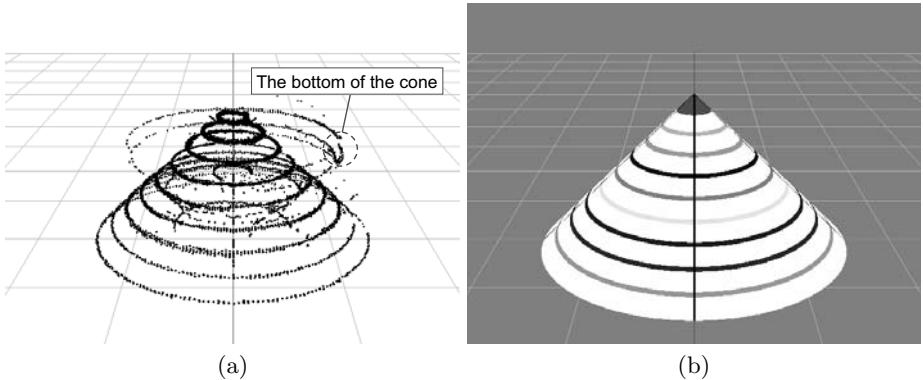
An actual experimental situation was simulated and the three-dimensional shape of the target object was estimated from the simulated image. A simulation environment is shown in Fig. 6. A target object, a circular cone, was placed inside the cylindrical mirror. The camera was set at the same altitude as the top of the cylinder, and the optical axis of the camera was arranged to be coincident with the center axis of the cylindrical mirror. The diameter and the height of the cylinder are 102.4mm, and the cone bottom diameter and the cone height are 76.8mm and 38.4mm, respectively. The cone has differently colored lines at even intervals. A profile of the fish-eye lens is assumed to be an equidistance projection. The size of the captured image is  $1024 \times 1024$  pixels, and the size of the converted image in the polar coordinate system is also  $1024 \times 1024$  pixels.

Figure 7 (a) shows a reconstructed model from Fig. 2 (a) using SSD with a  $20 \times 15$  size window, and Fig. 7 (b) is the actual shape of the target cone. The source region for calculating the value of SSD is limited to the curves that are drawn on the surface of the cone. The value of  $s$  in Eq. (1) varies from 0.6 to 1.0 in steps of 0.1. In addition, only the directly shoot image and the once reflected image are used to reconstruct this model. As shown in Fig. 7, the shape of the obtained model was almost similar to the actual shape of the cone. However, the bottom of the cone failed to be measured appropriately.

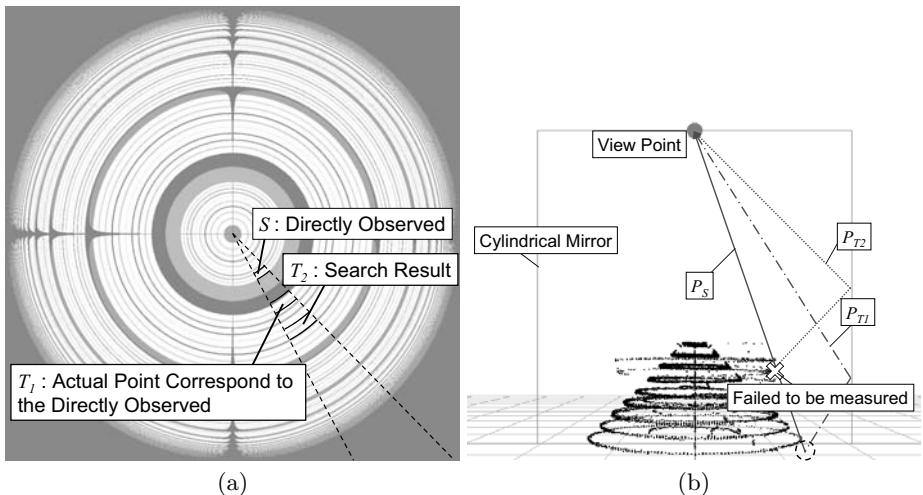
Figure 8 shows the case of identifying the set of points to measure the cone bottom. The true corresponding regions are the regions  $S$  and  $T_1$ . However, the region  $T_2$  was estimated instead of  $T_1$  in the simulation. This is because the curve's color in the region  $T_2$  is similar to that in the region  $S$ . The SSD



**Fig. 6.** An illustration of a simulation environment. A circular cone, which is a target object, is placed inside the cylindrical mirror. The optical axis of the camera is coincident with the center axis of the cylindrical mirror. The altitude of the camera lens is same as the top of the cylinder



**Fig. 7.** Reconstructed three-dimensional model from Fig. 2 (a) using the proposed method (a) and the actual shape of the target cone (b). These figures show that the shape of the obtained model using the proposed method is similar to the actual shape of the cone



**Fig. 8.** An example of searching the set of points at the bottom of the cone. (a) The regions in the captured image.  $S$  is the directly observed region,  $T_1$  is the actual region that corresponds to the region  $S$ , and  $T_2$  is the region that has the smallest SSD value between the region  $S$ . (b) A side view of the reconstructed model. The paths  $P_S$ ,  $P_{T1}$  and  $P_{T2}$  correspond to the region  $S$ ,  $T_1$  and  $T_2$  in the figure (a), respectively

value  $d_{ST2}$  between the regions  $T_2$  and  $S$  was smaller than that  $d_{ST1}$  between the regions  $T_1$  and  $S$ . Therefore, the height of the cone bottom failed to be estimated using the path  $P_S$  and the incorrect path  $P_{T2}$ . To avoid these mismatches, the outlier points are diminished by resizing the optimal window based on the reconstructed model.

In addition, some error factors should be taken into consideration in case of implementing the system. The cylindrical mirror in the actual system have distortions. Furthermore, it is difficult that the optical axis of the camera is set to be precisely identical with the center axis of the mirror. These error factors cause mismatches in searching the stereo pairs due to the geometrical distortion of the image. Distortion correction using a calibration pattern may be beneficial to search the correct stereo pairs.

## 4 Conclusion

In this paper, we have proposed the method for the three-dimensional measurement using a CCD camera and a cylindrical mirror. The three-dimensional shape of the object can be easily measured in a single shot using the proposed method. The result demonstrated that the similar three-dimensional model to the actual shape was reconstructed from the captured image. However, the SSD value between images that have a similar color was smaller than the value between the actual pair, and the shape of the target object failed to be measured in some cases.

Future work will aim at developing a robust method for searching the set of points in case that the several candidate regions with similar color or texture exist in the captured image. In addition, we are planning to build a prototype system and measure the three-dimensional shape of the real object using the prototype system.

## Acknowledgement

This research is partly supported by Core Research for Evolutional Science and Technology (CREST) Program “Advanced Media Technology for Everyday Living” of Japan Science and Technology Agency (JST).

## References

1. Shiaw, H., Jacob, R.J.K., Crane, G.R.: The 3D vase museum: A new approach to context in a digital library. Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries (2004) 125–134
2. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The digital michelangelo project: 3D scanning of large statues. Proceedings of ACM SIGGRAPH 2000 (2000) 131–144
3. Ullrich, A., Studnicka, N., Riegl, J.: Long-range high-performance time-of-flight-based 3D imaging sensors. Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (2002) 852–855
4. Sato, K., Inokuchi, S.: Three-dimensional surface measurement by space encoding range imaging. Journal of Robotic Systems **2** (1985) 27–39

5. Laurentini, A.: The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16** (1994) 150–162
6. Matusik, W., Buehler, C., Raskar, R., Gortler, S.J., McMillan, L.: Image-based visual hulls. *Proceedings of ACM SIGGRAPH 2000* (2000) 369–374
7. Klette, R., Schlüns, K., Koschan, A.: *Computer Vision: Three-Dimensional Data from Images*. Springer-Verlag Singapore (1998)
8. Gluckman, J., Nayar, S.K.: Rectified catadioptric stereo sensors. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2000) 2380–2387
9. Zhang, Z.Y., Tsui, H.T.: 3D reconstruction from a single view of an object and its image in a plane mirror. *Proceedings of International Conference on Pattern Recognition* **2** (1998) 1174–1176
10. Manabe, Y., Uranishi, Y., Yasumuro, Y., Imura, M., Chihara, K.: Three-dimensional measurement for small moving object. *Proceedings of SPIE-IS&T Electronic Imaging* **5665** (2005) 235–242

# Mottling Assessment of Solid Printed Areas and Its Correlation to Perceived Uniformity

Albert Sadovnikov, Petja Salmela, Lasse Lensu,  
Joni-Kristian Kamarainen, and Heikki Kälviäinen

Laboratory of Information Processing,  
Department of Information Technology, Lappeenranta University of Technology,  
P.O.Box 20, 53851 Lappeenranta, Finland  
`{sadovnik, psalmela, ltl, jkamarai, kalviai}@lut.fi`

**Abstract.** Mottling is one of the most important printing defects in modern offset printing using coated papers. Mottling can be defined as undesired unevenness in perceived print density. In our research, we have implemented three methods to evaluate print mottle: the standard method, the cluster-based method, and the bandpass method. Our goal was to study the methods presented in literature, and modify them by taking relevant characteristics of the human visual system into account. For comparisons, we used a test set of 20 grey mottle samples which were assessed by both humans and the modified methods. The results show that when assessing low-contrast unevenness of print, humans have diverse opinions about quality, and none of the methods accurately capture the characteristics of human vision.

## 1 Introduction

Printability of paper and print quality are very important attributes when modern printing applications are considered. Especially in prints containing images, high print quality is a basic requirement. Because of non-ideal interactions of paper and ink in high-speed printing processes, there are several undesired effects in prints. One of these effects is mottling which is related to density and gloss of print. It is the uneven appearance of solid printed areas, and it depends on the printing ink, paper type, and printing process. Depending on the phenomenon causing this unevenness, there exists three types of mottling: back-trap mottle (uneven ink absorption in the paper), water-interface mottle (insufficient and uneven water absorption of the paper causing uneven ink absorption), and ink-trap mottle (wet or dry; incorrect trapping of the ink because of tack) [1].

Mottling can be defined as undesired unevenness in perceived print density. In the ISO/IEC 13660 standard, a more technical definition is given [2]: “aperiodic fluctuations of density at a spatial frequency less than 0.4 cycles per millimeter in all directions”. In most situations, mottling is a stochastic phenomenon, but different types of noise in print related to mottling can include some form of

regularity. For example, possibly regular drift in the printing process causes macro-scale noise in print, whereas structures in the paper formation are random in nature and cause micro-scale noise invisible to a human being as such.

Several methods to evaluate mottling by an automatic machine vision system have been proposed. The ISO 13660 standard includes a method for monochrome images. It is based on calculating the standard deviation of small tiles within sufficiently large area [2]. In the standard, the size of the tiles is set to a fixed value, which is a known limitation [3]. The first improvement to the standard method was to use tiles of variable sizes [4]. Other methods relying on clustering, statistics, and wavelets have also been proposed [5, 6, 7]. Other approaches to evaluate greyscale mottling have their basis in frequency-domain filtering [8], and frequency analysis [9]. All of the before-mentioned methods are designed for binary or greyscale images. If colour prints were assessed, the correlation of the methods to human assessments would be severely limited. Also the grounds for the methods do not arise from any models for the phenomena causing mottling, nor vision science.

Mottling can be physically defined, but it becomes problematic when a print is perceived. If a person looking at a solid print perceives unevenness, mottling is a problem. Thus, the properties and limits of the human visual system must be taken into account when proper methods to assess mottling are designed. This is especially very important in the assessment of colour images. When perception of image noise is of concern, visual sensitivity to contrast and spatial frequencies of the human visual system (HVS) is independent of luminance within common luminance levels [10]. However, contrast sensitivity depends on spatial frequency [11], thus, mottles of different sizes are perceived differently. The peak sensitivity of the HVS is approximately at 3 cycles/degree, and the maximum detected frequency is from 40 cycles/degree (sinusoidal gratings) [12] to over 100 cycles/degree (single cycle) [13].

The purpose of this work was to compare the artificial methods to a human assessment of mottling samples. In our study, we sought proper background for the methodological selections based on vision science. We implemented three methods based on literature, and modified them as needed to accommodate appropriate knowledge concerning the HVS.

## 2 Methods

We implemented three methods to automatically assess print mottle: the standard method to evaluate image quality of printer systems [2], a cluster method [4], and a band-pass method [8]. We slightly modified them as needed to accommodate an appropriate contrast-sensitivity function for the human visual system. To study the correlation of the implemented methods with human perception, we carried out an initial human assessment of 20 mottling samples. We asked experts and laymen to evaluate perceived mottling in the samples. The mean of these subjective assessments was used as a reference into which the results of the methods were compared.

## 2.1 Standard Method

The ISO 13660 standard is designed for assessing print quality of office equipment that produce monochrome output [2]. The density attributes for large print areas (larger than 21.2 mm squared) include graininess and mottling. In the standard, a fixed value has been chosen to separate this two forms of print unevenness. Aperiodic fluctuations of print density at spatial frequencies higher than 0.4 cycles/degree are considered as graininess, whereas frequencies lower than the limit are mottling. The standard method is presented in Algorithm 1.

### Algorithm 1.

```

1:  $\dots \dots \dots \dots \dots \dots \dots$ 
2:  $\dots \dots \dots \dots \dots \dots \dots$ 
3:  $\dots \dots \dots \dots \dots \dots \dots$ 

```

In Step 1, the region of interest is divided into tiles of size 1.27 mm squared. Within each tile, 900 independent measurements of density are made.

## 2.2 Cluster Method

This method is based on the idea by Wolin [4]. In this method, the raster image is filtered with a low-pass filter, and thresholded separately on both the lighter and darker sides of the median grey value. Geometric measures of the thresholded blobs (mottles) are used as features. In our implementation each blob is weighted by its size and contrast, and the weighted number of blobs is used as the mottling index of the sample. The method is shown in Algorithm 2.

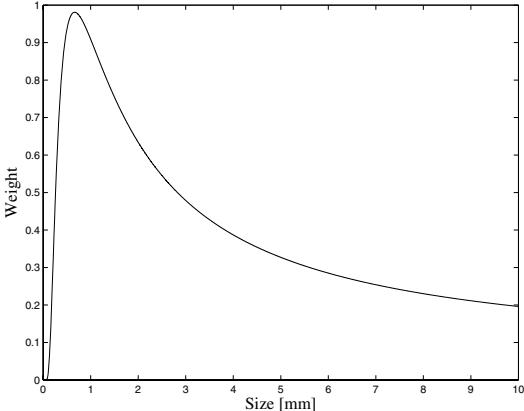
### Algorithm 2.

```

1:  $\dots \dots \dots \dots \dots \dots \dots$ 
2:  $\dots \dots \dots \dots \dots \dots \dots$ 
3:  $\dots \dots \dots \dots \dots \dots \dots$ 
4:  $\dots \dots \dots \dots \dots \dots \dots$ 
5:  $\dots \dots \dots \dots \dots \dots \dots$ 
6:  $\dots \dots \dots \dots \dots \dots \dots$ 
7:  $\dots \dots \dots \dots \dots \dots \dots$ 

```

In Step 1, a suitable Gaussian low-pass filter is designed to meet the specified lower limit of feature size, 0.5 mm. The cutoff frequency for the filter naturally depends on the image resolution, and size. This step practically removes the dot pattern caused by screening, and softens any isolated defects in print (which are not considered as mottling). In Step 2, the image is thresholded on both sides of the median based on the Weber fraction. From psychometric research, it is known that the human threshold for optical contrast, when expressed as the Weber fraction  $dR/R$ , tends to be constant over a wide range of reflectances  $R$  [8, 14]. This suggests that suitable thresholds around the image median can be selected, and the bins for contrast classes can be of equal size. If the mean reflectances of the samples vary considerably, the logarithmic nature (Fechner



**Fig. 1.** Weighting of size classes (Mannos CSF)

or some power law) of the sensitivity of the HVS should be considered. In Step 3, morphological opening is used to remove blobs which are too small to be considered as mottling. Also narrow isthmuses between the blobs are cut by the same operator. In Step 4, all blobs in touch with the image border are removed. In Step 5, blob areas are computed as features representing the blobs. In Step 6, the blobs are divided into 10x10 classes according to their area and contrast. The classes for the area are 0 – 1 mm, 1 – 2 mm, ..., 9 – 10 mm. Blobs larger than 10 mm are discarded. The classes for contrast are 0-1%, 1-2%, ..., 9-10%. Spots with higher contrast are discarded. A monochrome contrast sensitivity function (CSF) shown in Fig. 1 is used to weight the contrast information of blobs of varying sizes [15]. Note that the CSF is derived from perception of sinusoidal patterns, but mottling is a stochastic phenomenon. The number of cycles in a grating visible to a human observer substantially affect the contrast sensitivity [16]. However, mottles do not appear as single cycle gratings [13], and thus, we use the one derived using sinusoidal gratings.

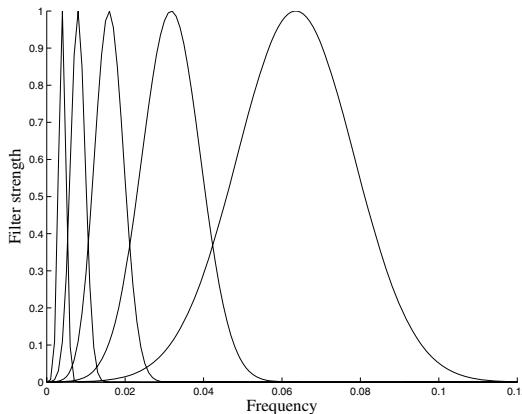
Mottling index is computed as a sum of products of size and contrast weights for each blob, i.e.,

$$M = \sum_i W_a(a_i)W_c(c_i) \quad (1)$$

where  $i$  is the index for a mottle,  $a_i$  is the area of the  $i$ th mottle,  $c_i$  is the contrast of  $i$ th mottle,  $W_a(a_i)$  is the weight of a size class of the  $i$ th mottle, and  $W_c(c_i)$  is the weight of a contrast class of the  $i$ th mottle.

### 2.3 Bandpass Method

This method is based on applying a series of Gaussian band-pass filters to the image in the frequency domain, and computing the coefficient of variation of reflectance ( $CV_R$ ) for each image representing a frequency band. Different coef-



**Fig. 2.** The filters in 2-D representing the 0.5-1, 1-2, 2-4, 4-8, and 8-16 mm spatial bands

ficients represent the difference in reflectance within each band. The coefficients are weighted with the CSF and then summed together as the mottling index [8]. The method is summarized in Algorithm 3.

### Algorithm 3.

- 1:  $\dots \rightarrow \dots$
- 2:  $\dots \rightarrow \dots$
- 3:  $\dots \rightarrow \dots$
- 4:  $\dots \rightarrow \dots$

In Step 1, the image is filtered in the frequency domain with a series of bandpass filters. Five spatial bands are fixed to an octave series: 0.5-1, 1-2, 2-4, 4-8, and 8-16 mm. The band containing the smallest details has been included when compared to [8]. The Gaussian filters are illustrated in Fig. 2. The DC component is set to 1 so that the mean grey value of the image does not change due to filtering.

In Step 2, the coefficients of variation are computed in the spatial domain for each band. The coefficient of variation is the ratio of standard deviation of reflectance and mean reflectance, i.e.,

$$CV_R = \frac{\sigma_R}{R}. \quad (2)$$

In Step 3, the coefficients are weighted with a CSF [15] illustrated in Fig. 1. The weights are taken at points representing 0.75, 1.5, 3, 6, and 10 mm.

## 2.4 Visual Assessment

To compare the results of the implemented methods to human perception, we circulated a set of 20 mottling samples, and asked the human observers to eval-

uate the perceived mottling. The mean values of these subjective assessments were used as initial reference mottling indices against which the results of all the machine vision methods were compared.

The questionnaire for the assessment consisted of two parts. In the first part, two samples were concurrently compared, and the observer was asked to select the sample which has less mottling. The main function of this part was to present all samples to the observer, and to give some idea of different forms and levels of mottling. These pairwise evaluations could also be used to find possible inconsistencies in the second part. In the second part, each sample was evaluated one at a time, and the observer was asked to rate its level of mottling in a five point Likert scale. There were also two control questions for the observer about the number of times the person had previously evaluated mottling, and the time needed for the test. The primary function of the questionnaire was to evaluate perceived level of mottling of the test set. The secondary, and unwanted, function was to evaluate the person's capability to evaluate mottling and thoroughness of test set evaluation.

The results of the assessments were processed as follows. The people taking the test were divided into two distinct groups based on the control question about the number of times the person has evaluated mottling. The first group consisted of common people who evaluated mottling for the first time and were not experts in the field of print assessment. The second group was formed by experts who evaluated prints as a part of their work. The second control question about the time spent for the test was used to estimate carefulness of the samples evaluation. Selection criteria for outliers were difficult to design. Each observer had his or her own way of selecting the mean value and the use of the scale. However, the mean and standard deviation could be used as elementary criteria to select outliers. If either one differs significantly from the average of all assessments, the assessment was marked as an outlier.

### 3 Experiments

We present the results for the 20 K70 (70% black) samples. The original samples are approximately 4.5 cm × 4.5 cm in size. The paper used for printning is 70 g/m<sup>2</sup> LWC (Lightweight Coated) paper and the samples were printed using heatset offset printing process in the KCL layout (KCL heatset layout 01/2003, 60 I/cm, round dot, upper units). The samples were originally scanned with 1200 dpi and 2.2 gamma using flatbed office scanner. The gamma value was not altered before applying the machine vision methods. To reduce computing time, the images were re-sampled to 600 dpi.

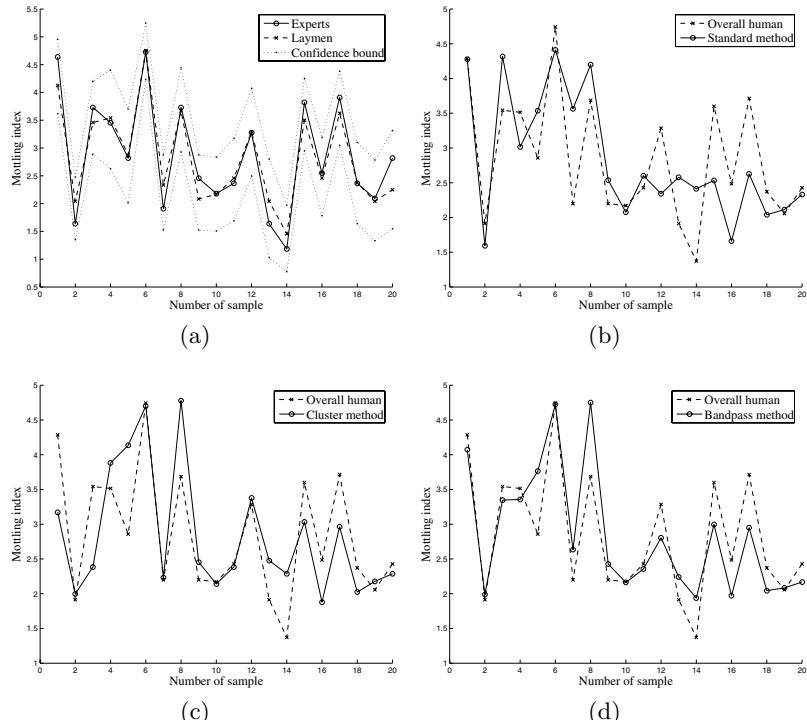
We inspected mottle sizes ranging from 0.5 to 16 mm while viewing the sample from a distance of 30 cm (spatial frequency range 0.03-1 cycles/mm). Spatially higher- and lower-frequency unevennesses were considered as graininess and banding. Also the inspected contrast of print density was limited to ±10% of the median grey-value of an image.

### 3.1 Visual Assessment

These results are based on the 35 human evaluations. The assessments were made in normal office light conditions. However the conditions were not constant, but could vary from one evaluation to another. Evaluators were divided into two groups: experts (12 representatives) and laymen (23 representatives). This division was made based on the professionalism of the person, i.e., the number of mottling evaluations done prior to the suggested one. As it can be seen from Fig. 3(a), there is only a little difference in evaluations between laymen and experts. This is natural since it would be confusing if experts evaluated print quality of samples in which mottling is most visible completely distinctly to end-users. Confidence bounds in Fig. 3(a) present the average results across the whole population  $\pm$  standard deviation, and show how similar the mottling indices were among all evaluators.

### 3.2 Machine Vision Assessment

The standard method was implemented in the way it is described in the ISO 13660 standard [2]. The implementation of this method is easy and does not



**Fig. 3.** Mottling assessments: (a) Human evaluation; (b) Standard method; (c) Cluster method; (d) Bandpass method

require much programming effort. As it was expected, the results produced by the standard method show low correlation with human evaluation (see Fig. 3(b)). In the standard, the size of the tiles is set to a fixed value which is a known limitation [3]. The cluster method can handle only monochrome images. Another shortcoming is its performance: processing images with a large number of blobs is time consuming. The results of this method can be seen in Fig. 3(c). The bandpass method can also handle only monochrome images. A small number of bands limits the number of spatial classes, and the method becomes similar to a series of low-pass filters used in the early mottling methods. Performance of the method is limited by the resolution of the image and the number of bands. The results of this method can be seen in Fig. 3(d).

All the artificial methods produced mottling indexes in their own scale. Thus, appropriate scaling is needed for the method comparison. We used simple normalization which equalizes mean value and standard deviation of the experimental values across the samples.

### 3.3 Results Summary

In Table 1 inter-method similarity is presented. Correlation coefficients were used as the similarity measure.

**Table 1.** Mottling assessment correlations

Methods	Overall	Experts	Laymen	Standard	Cluster	Bandpass
Overall human	1.0000	0.9848	0.9957	0.6956	0.7330	0.8579
Experts	0.9848	1.0000	0.9644	0.6568	0.6717	0.8125
Laymen	0.9957	0.9644	1.0000	0.7078	0.7568	0.8715
Standard	0.6956	0.6568	0.7078	1.0000	0.6742	0.8810
Cluster	0.7330	0.6717	0.7568	0.6742	1.0000	0.9070
Bandpass	0.8579	0.8125	0.8715	0.8810	0.9070	1.0000

The collected correlation data allow to state that the bandpass method outperforms the other two methods. It can be also noticed that the machine vision methods correlate better among each other than with human evaluation based data. This leads to the conclusion that all artificial methods have a similar nature and the model of human visual system they assume is not accurate. Fig. 3 shows performance graphs for different assessment approaches.

## 4 Conclusions

In the presented work, we performed an initial comparison between human and machine vision evaluation of mottling phenomenon. The results of the human evaluation appear to be highly distributed and, thus, a larger number of assessments is needed both in evaluators and in samples. The high deviation in single

sample evaluation results leads to the conclusion that a machine vision system modelling an average end-user is necessary. This could bring more precision in delivering printed products of desired quality.

The presented machine vision methods, though having a relatively good correlation with averaged human observation, still need improvement in the sense of modelling of the human visual system. The standard method presented can be considered only as a starting point because this method does not model the HVS at all and also it does not have significant correlation with the human mottling evaluation. The cluster method is based on spatial image processing. This method has some HVS background, but at the same time the approach of "mottle by mottle" processing shows little perspective for improvement. Among the presented methods, the bandpass method shows the best results and it has HVS-based grounds. This method shows potential for improvement and is definitely a candidate for an industrial level machine vision application.

The goals for the future research can be defined as follows:

- Making methods closer to human perception.
- Incorporating mottling evaluation of colour samples.

The general conclusion of our research, is that for the implementation of a machine vision solution to the human perception problem, one needs a suitable HVS model and good statistical characteristics of how the humans perceive the phenomenon.

## Acknowledgments

This work was done as a part of Papvision project funded by European Union, National Technology Agency of Finland (TEKES Projects No. 70049/03 and 70056/04), and Academy of Finland (Project No. 204708).

## References

1. IGT: IGT information leaflet w57: Back trap mottle. WWW:www.igt.nl (2002) [Accessed 2005-02-25]. Available: <http://www.igt.nl/igt-site-220105/index-us/w-bladen/GST/W57.pdf>.
2. ISO: ISO/IEC 13660:2001(e) standard. information technology - office equipment - measurement of image quality attributes for hardcopy output - binary monochrome text and graphic images. ISO/IEC (2001)
3. Briggs, J., Forrest, D., Klein, A., Tse, M.K.: Living with ISO-13660: Pleasures and perils. In: IS&Ts NIP 15: 1999 International Conference on Digital Printing Technologies, IS&T, Springfield VA (1999) 421–425
4. Wolin, D.: Enhanced mottle measurement. In: PICS 2002: IS&T's PICS conference, IS&T (2002) 148–151
5. Armel, D., Wise, J.: An analytic method for quantifying mottle - part 1. Flexo (1998) 70–79
6. Armel, D., Wise, J.: An analytic method for quantifying mottle - part 2. Flexo (1999) 38–43

7. Streckel, B., Steuernagel, B., Falkenhagen, E., Jung, E.: Objective print quality measurements using a scanner and a digital camera. In: DPP 2003: IS&T International Conference on Digital Production Printing and Industrial Applications. (2003) 145–147
8. Johansson, P.Å.: Optical Homogeniety of Prints. PhD thesis, Kungliga Tekniska Högskolan, Stockholm (1999)
9. Rosenberger, R.R.: Stochastic frequency distribution analysis as applied to ink jet print mottle measurement. In: IS&Ts NIP 17: 2001 International Conference on Digital Printing Technologies, IS&T, Springfield VA (2001) 808–812
10. Barten, P.: Contrast Sensitivity of the Human Eye and its Effects on Image Quality. SPIE (1999)
11. Schade, O.H.: Optical and photoelectric analog of the eye. *Journal of the Optical Society of America* **46** (1956) 721–739
12. Kang, H.R.: Digital Color Halftoning. SPIE & IEEE Press (1999)
13. Campbell, F.W., Carpenter, R.H.S., Levinson, J.Z.: Visibility of aperiodic patterns compared with that of sinusoidal gratings. *Journal of Physiology* **204** 283–298
14. Pratt, W.: Digital Image Processing. A Wiley-Interscience publication (1991)
15. Mannos, J., Sakrison, D.: The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory* **20** (1974) 525–536
16. Coltman, J.W., Anderson, A.E.: Noise limitations to resolving power in electronic imaging. In: Proceedings of the Institute of Radio Engineers. (Volume 48.) 858–865

# In Situ Detection and Identification of Microorganisms at Single Colony Resolution Using Spectral Imaging Technique

Kanae Miyazawa<sup>1</sup>, Ken-ichi Kobayashi<sup>1</sup>, Shigeki Nakauchi<sup>1</sup>, and Akira Hiraishi<sup>2</sup>

<sup>1</sup> Department of Information and Computer Sciences,  
Toyohashi University of Technology,

1-1 Hibarigaoka, Tempaku, Toyohashi, Aichi 441-8580, Japan

{kanae, kobayashi04, naka}@bpel.ics.tut.ac.jp

<http://www.bpel.ics.tut.ac.jp/>

<sup>2</sup> Department of Ecological Engineering,

Toyohashi University of Technology,

1-1 Hibarigaoka, Tempaku, Toyohashi, Aichi 441-8580, Japan

hiraishi@eco.tut.ac.jp

**Abstract.** In situ detection and identification of microorganisms in the environment are important in general microbial ecology. Also the rapid inspection of microbial contamination at food processing plant is urgent task. We propose a method of detecting and identifying microorganisms for rapid inspection using spectral imaging technique. Spectral images of photosynthetic and non-photosynthetic bacterial colonies having different absorption spectra in near infrared wavelength region were measured directly from Petri dish. Bacterial region in the images was first detected and then identified using multiple discriminant analysis. Detection and identification errors for various sized colonies were analyzed. As the result, colonies with diameters of 100 and 300  $\mu\text{m}$  were detected and identified with sufficient accuracy, respectively. This means the time for detection and identification can be shorten less than a half and about several weeks compared with the conventional methods.

## 1 Introduction

In situ detection and identification of microorganisms in the environment are important in general microbial ecology. Also food poisoning incidents caused by microorganisms have frequently occurred recently and the inspection at food processing plant is conducted with scrupulous care. Traditional culturing techniques, in which microorganisms are cultured for a few days and inspected by human eyes, is still widely used for detecting microbial contamination in food products. However, it takes relatively long time and the whole plant is possibly contaminated before microorganisms are detected. In the worst case, the contaminated food products must be recalled. It makes not only huge economic loss but also compromises their reputation. Therefore, rapid inspection of microbial contamination at food processing plant is urgent task.[1]

Fluorescence microscopy, flow cytometry, polymerase chain reaction techniques have recently been the focus of attention for identification of microorganisms.[2],[3] These methods can be used as powerful tools for precise identification in laboratory.

However, they are not appropriate ways for rapid food inspection at plants, because chemical pretreatment including poisonous staining is needed and/or the equipments are extremely expensive. Not precise identification of microorganisms but rapid screening is needed at the plants. Rapid inspection technique using shape information of growing microorganisms has been proposed and the equipment has come onto the market. This can be useful for rapid detection for relatively small size microorganisms called microcolony, however alien substances are possibly mis-detected as microorganisms. In the point of view, shape information is not enough for detection.

On the other hand, it is known that microorganisms like bacteria have typical absorption spectra at ultra violet (UV) and infrared (IR) wavelengths regions caused by pigment, protein and so on. In laboratory, spectrophotometer is usually used for identification of microorganisms. Sugiura et al.[4] proposed a simple method for absorption spectrophotometry to identify photosynthetic pigments of microbial mats using a portable spectrophotometer in the field. This method is simple and easy to use in the field. However, a large quantity of samples like mats are required for measurement, because the sample is placed between two slide glasses and the transmitted light is measured by putting samples between light source and detector. In this case, average spectra of microbial mats can be measured and it is not applied to food inspection which needs to detect and identify a single colony. To detect a single colony of target microorganisms from contaminated foods, imaging technique is necessary.

Spectral imaging technique, in which each pixel in an image has spectral information, has received a great deal of attention recently. Spectral imaging devices are under a rapid development and the applications of spectral imaging including UV and IR wavelengths regions are expected. Spectral imaging technique has been used to detect stained[5],[6] and unstained microbial colonies[7],[8] which are fully developed. In this study, we propose a method to detect and identify a single colony of live unstained microorganisms for rapid inspection using spectral imaging technique. The method was applied to various sized colonies and we investigated the limitation of the method.

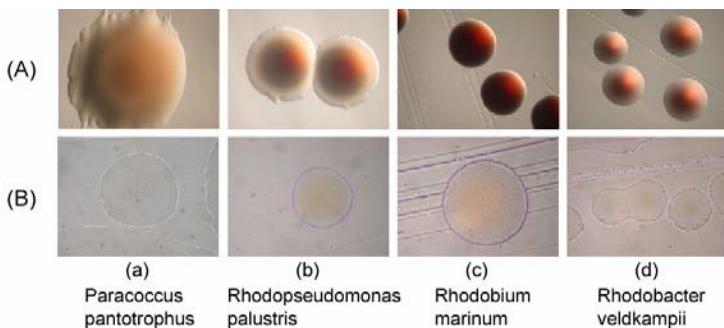
The paper is organized as follows. In Section 2, we describe materials and methods consisted of sample bacteria used in the experiment, an experimental setup for spectral imaging, method for detection and identification of microbial colonies and evaluation method. The experimental results are shown in Section 3 and Section 4 gives the discussion and conclusions.

## 2 Materials and Methods

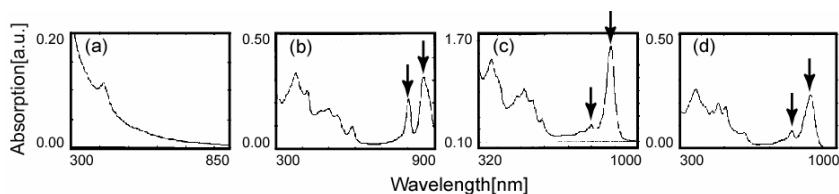
### 2.1 Bacteria

(a)*Paracoccus pantotrophus* JCM6892, (b)*Rhodopseudomonas palustris* DSM123, (c)*Rhodobium marinum* DSM2698 and (d)*Rhodobacter veldkampii* ATCC35703 were used as model organisms. Examples of bacterial colonies used in the experiment are shown in Figure 1(A), (B). The photos were taken by an RGB digital camera attached to an optical microscope with (A) 4- and (B) 10-power object lens, respectively. Both in (A) and (B), (b), (c) and (d) are photosynthetic bacteria having distinctive spectral absorption peaks caused by pigment-protein complexes with bacteriochlorophyll *a* (Bchl *a*). A non-photosynthetic bacterium (a) which has no Bchl *a* was also used as a

control bacterium. These bacteria were cultured aerobically on agar plates under incandescent illumination and were used for experiment without any chemical pre-treatment such as staining. The sizes of the colonies were about (A) 1-2 millimeter and (B) a few hundreds micrometer. Detection and identification of colonies in (A) may not difficult task for experts. However, colonies in (B) are difficult to detect because of its small size and almost impossible to identify even for experts because colonies are too thin and the colors are all reddish and transparent. Shape information cannot help us because all of them are flat and round. Figure 2 shows absorption spectra of bacterial cells measured with a conventional spectrophotometer. Distinctive peaks caused by pigment-protein complexes with Bchl *a* can be seen in (b), (c) and (d), where (a) has no peaks at infrared region.



**Fig. 1.** Four species of bacteria used in the experiment. (A) Relatively large size colonies taken with 4-power object lens, (B) microcolonies taken with 10-power object lens. The sizes of the colonies were about (A) 1-2 millimeter and (B) a few hundreds micrometer. (b)-(d) are photosynthetic bacteria having distinctive spectral absorption peaks caused by pigment-protein complexes with bacteriochlorophyll *a* (Bchl *a*) and (a) is a non-photosynthetic bacterium having no Bchl *a*

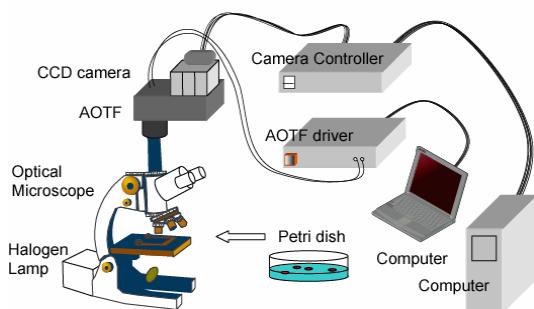


**Fig. 2.** Absorption spectra of bacterial cells measured with a conventional spectrophotometer. Distinctive peaks caused by pigment-protein complexes with Bchl *a* can be seen at infrared region in (b), (c) and (d) as pointed by arrows, where (a) has no peaks at infrared region

## 2.2 Experimental Setup

An experimental setup for a spectral imaging system is shown in Figure 3. A Petri dish, whose cover was opened, was set on an optical microscope with a halogen lamp (Nikon SMZ-10 with 4-power zoom object lens, Nikon LABOPHOT-2 with 10-power object lens). An intensity image of the bacterial colonies trans-illuminated by

the halogen lamp was taken with a monochrome CCD camera (HAMAMATSU, ORCA-ER-1394) through an acousto-optic tunable filter (AOTF) (Brimrose, CVA100-0.4-1.0) whose transmitting wavelength can be electrically changed visible to near infrared region (400 to 1000 nm). Wavelength range for taking images in this experiment was between 750 and 946 nm at 2 nm intervals, because photosynthetic bacteria could be identified with the spectral resolutions of 2 nm in the region. According to Lambert-Beer law, absorption spectral images were calculated using spectral images of colonies and reference. As the reference, a spectral image of a Petri dish with culture media, which was not included colonies, was used for each species. Reference images varied depending on the media whose ingredients were different, but not on the places where taking images. The experiment was done in laboratory environment with the temperature of 24 - 25 degree Celsius.



**Fig. 3.** Spectral imaging system. An intensity image of the bacterial colonies trans-illuminated by the halogen lamp was taken with a monochrome CCD camera through an acousto-optic tunable filter

### 2.3 Detection and Identification

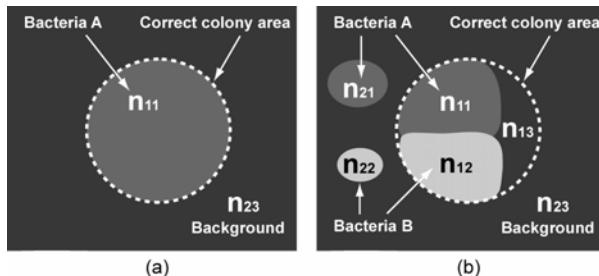
Analysis consisted two phases, detection and identification. First, pixelwise detection was done by separating colony area and background area by multiple discriminant analysis (MDA) using mean absorption and standard deviation of the spectra. The mean absorption of colony area is theoretically high compared with that of background area which was consisted of Petri dish and culture media. Experimentally, however, the mean absorption of colony area is sometimes lower than that of background area due to the experimental artifacts. On the other hand, standard deviation of the spectra of colony area is high in infrared region because of absorption peaks caused by pigment-protein complexes with Bchl *a*. Therefore both mean absorption and standard deviation of the spectra were used for the detection in the first phase.

Next, pixelwise identification was done by MDA only for the colony area detected in the first phase. Spectra extracted from each colony area were used as the training data for MDA. In the results of identification, there were isolated pixels like pepper noises which were not reasonable as real bacteria. To reduce such noises,  $5 \times 5$  majority filter was applied. Final results of identification were pseudo-colored as shown in Section 3. Evaluation of detection and identification for colonies with various sizes was carried out in Subsection 3.3. Evaluation method is shown in Subsection 2.4.

## 2.4 Evaluation

Evaluation of detection and identification was done by comparing with known results. First, mean absorption image was made. Colonies and background area were manually selected from the mean absorption image. We call them as known results and used for evaluation. The known result of largest colony in each species was used for training data in MDA for identification described in Subsection 2.3. Uncertain area was not used.

Figure 4 shows the schematic diagram of (a) ideal and (b) six possible results for identification. In case bacteria A exists in dashed circle and no other bacteria exist in the Petri dish as shown in Figure 4(a), identified result in the circle has three possibilities as shown in Figure 4(b), i.e.,  $n_{11}$ ,  $n_{12}$  and  $n_{13}$ .  $n_{11}$  is correct answer,  $n_{12}$  is bacteria at correct place but wrong species and  $n_{13}$  is wrong material (background) at wrong place. Identified result in the background area has also three possibilities, i.e.,  $n_{21}$ ,  $n_{22}$  and  $n_{23}$ .  $n_{21}$  is correct bacteria at wrong place,  $n_{22}$  is wrong bacteria at wrong place and  $n_{23}$  is correct answer. Using these values, correct detection ratio and correct identification ratio were defined as shown in Eq. 1-6,



**Fig. 4.** Schematic diagram of (a) ideal and (b) six possible results for identification. In case bacteria A exists in dashed circle as shown in (a), identified result in the circle has three possibilities as shown in (b), i.e.,  $n_{11}$ ,  $n_{12}$  and  $n_{13}$ . Identified result in the background area has also three possibilities, i.e.,  $n_{21}$ ,  $n_{22}$  and  $n_{23}$ . Using these values, correct detection and identification ratio were defined

$$D_{mc} = (n_{11} + n_{12}) / (n_{11} + n_{12} + n_{13}) . \quad (1)$$

$$D_{bk} = n_{23} / (n_{21} + n_{22} + n_{23}) . \quad (2)$$

$$D = (D_{mc} + D_{bk}) / 2 . \quad (3)$$

$$C_{mc} = n_{11} / (n_{11} + n_{12} + n_{13}) . \quad (4)$$

$$C_{bk} = n_{23} / (n_{21} + n_{22} + n_{23}) . \quad (5)$$

$$C = (C_{mc} + C_{bk}) / 2 . \quad (6)$$

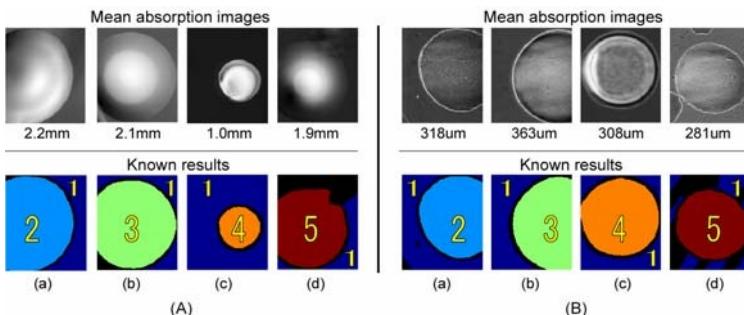
where  $D_{mc}$ ,  $D_{bk}$  and  $D$  are correct detection ratio of bacteria, background and in an image which is the average of  $D_{mc}$  and  $D_{bk}$ , respectively.  $C_{mc}$ ,  $C_{bk}$  and  $C$  are correct identification ratio of bacteria, background and in an image which is the average of  $C_{mc}$  and  $C_{bk}$ , respectively. In detection, species of bacteria were not considered.

### 3 Results

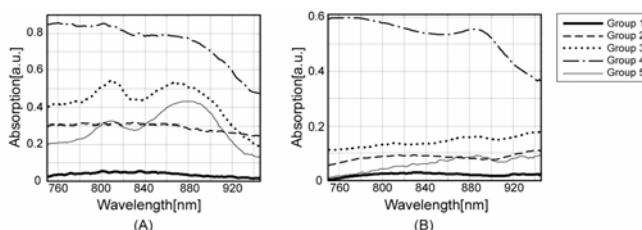
Detection and identification results for relatively large colonies shown in Figure 1(A) and microcolonies shown in Figure 1(B) were described in Subsection 3.1 and 3.2, respectively.

#### 3.1 Large Colonies

Figure 5(A) shows mean absorption images, diameters of colonies and known results with numbers of groups. The size of the image is  $240 \times 201$  pixels which correspond to about  $2232 \times 1869 \mu\text{m}$  ( $9.3 \mu\text{m} / \text{pixel}$ ). All colonies in Figure 5(A) are the largest ones in each species taken with 4-power object lens. Numbers of groups were assigned as 1: Background, 2: *Paracoccus pantotrophus*, 3: *Rhodopseudomonas palustris*, 4: *Rhodobium marinum* and 5: *Rhodobacter veldkampii*. Figure 6(A) shows

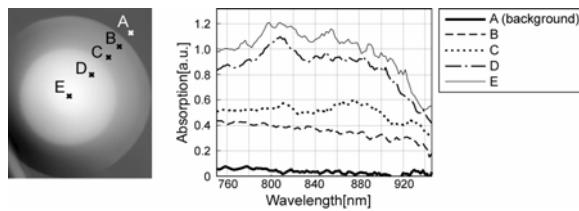


**Fig. 5.** Mean absorption images, diameters of colonies and known results with numbers of groups. (A) Large colonies, (B) microcolonies



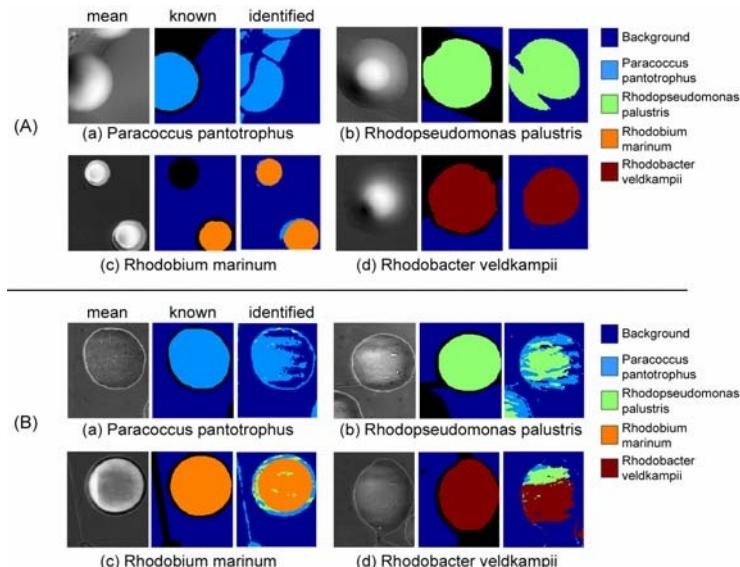
**Fig. 6.** Examples of absorption spectra averaged in each group. (A) Large colonies, (B) microcolonies

examples of absorption spectra averaged in each group. Absorption peaks caused by pigment-protein complexes with Bchl *a* were seen even though the spectral shape was smooth compared with Figure 2. The reason of the smoothness was that colonies were not chemically preprocessed at all and pigment-protein complexes with Bchl *a* was covered with other materials such as cell membrane. Absorption spectra at different pixels are also shown in Figure 7. Absorption at the center of a colony is large, but still dull shaped because of the cell membrane.



**Fig. 7.** Absorption spectra at different pixels. Absorption at the center of a colony is large, but still dull shaped because of the cell membrane

Figure 8(A) shows, from left to right, mean absorption images, known results and identification results for each species. Correct detection ratio D and correct identification ratio C are, (a) 94.8%, 92.0%, (b) 96.0%, 96.0%, (c) 99.8%, 96.8% and (d) 85.5%, 85.5%, respectively. Most species were correctly identified.



**Fig. 8.** Mean absorption image, known results and identification results. (A) Large colonies, (B) microcolonies

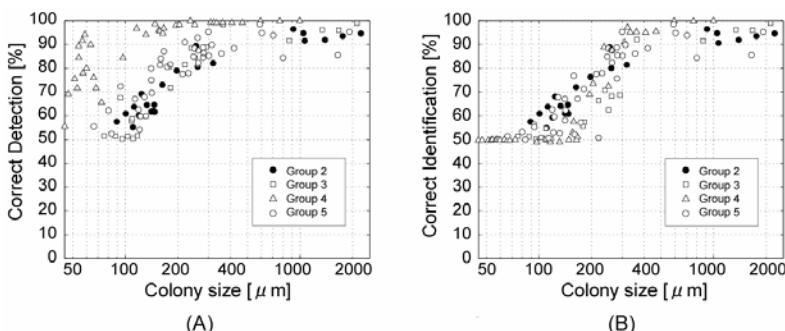
### 3.2 Microcolonies

Figure 5(B) shows mean absorption images, diameters of colonies and known results with numbers of groups. The size of the image is  $240 \times 201$  pixels which correspond to about  $360 \times 101 \mu\text{m}$  ( $1.5 \mu\text{m} / \text{pixel}$ ). All colonies in Figure 5(B) are the largest ones in each species taken with 10-power object lens. Numbers of groups were assigned as 1: Background, 2: *Paracoccus pantotrophus*, 3: *Rhodopseudomonas palustris*, 4: *Rhodobium marinum* and 5: *Rhodobacter veldkampii*. Figure 6(B) shows examples of absorption spectra averaged in each group. Compared with Figure 6(A), absorption peaks caused by pigment-protein complexes with Bchl *a* could not be seen clearly, because thicknesses of colonies were quite thin and absorption became low.

Figure 8(B) shows, from left to right, mean absorption images, known results and identification results for each species. Correct detection ratio D and correct identification ratio C are, (a) 80.6%, 80.1%, (b) 88.5%, 70.8%, (c) 99.9%, 88.9% and (d) 84.6%, 75.7%, respectively. Compared with Figure 8(A), correct ratio was low. The reason of the error was mis-identification of bacteria (b), (c), (d) to background or bacteria (a). This is understandable, because background and bacteria (a) do not have any absorption peaks in the wavelength region and it is easy to confuse with those.

### 3.3 Limit of Colony Size

Evaluation of detection and identification of colonies with various sizes was carried out. Figure 9 shows the relation between colony size and (A) correct detection D, (B) correct identification C, respectively. The number of colonies were 5(Group2), 3(Group3), 4(Group4), 5(Group5) and total 17 in Figure 9(A) and 9(Group2), 10(Group3), 10(Group4) and 10(Group5) and total 39 in Figure 9(B). In Figure 9(A) and (B), the colonies of these bacteria used in the experiment were correctly detected / identified with the ratio of over 60 / 80% in case the colony size was over  $100 / 300 \mu\text{m}$ , respectively. In this experiment, background was almost 100% correctly detected and identified. It means that the error came from colony area. For example, correct detection D = 60% means not 60 colonies in 100 colonies can be detected, but background area of 100% and colony area of 20% can be detected. In the viewpoint of microbiology, 60% for detection is accurate enough because colony area of 20% can



**Fig. 9.** The relation between (A) colony size and correct detection, (B) correct identification

be easily counted as a single colony. 100% detection ratio is not necessary. Therefore, the limit of colony size for detection and identification with sufficient accuracy is about 100  $\mu\text{m}$  and 300  $\mu\text{m}$  from the microbiological viewpoint, respectively.

## 4 Discussion and Conclusions

We presented an application of the spectral imaging technique to detect and identify microorganisms for rapid food inspection. Colonies having a diameter of about 100  $\mu\text{m}$  and 300  $\mu\text{m}$  were detected and identified with sufficient accuracy for every species used in this study, respectively. This means that the detection time can be shorten less than a half and the identification time can be shorten about several weeks compared with the conventional methods such as spectrophotometer which needs weeks for sample preparation.

The main advantages of this technique are the following: living microcolonies can be detected and identified without any chemical pretreatment such as staining, a single colony on the surface of Petri dish can be directly used as a sample, no extended culturing is necessary to prepare samples, alien substances cannot be mis-detected since spectral information is used.

In this study, we used photosynthetic bacteria as model bacteria to evaluate the validity of this method. Next steps in this research are as follows: to detect and identify contaminated samples with several species in a Petri dish, to fix target bacteria and to customize culturing and measuring condition for rapid food inspection at food processing plant. In practice, selection of appropriate culture media and/or selection of wavelengths can be considered. At this stage, spectral resolution was 2 nm and 99 images were measured for one spectral image. Precise spectral information, however, is not needed if a few spectral channels were well chosen. Dedicated spectral imaging system with a few filters could be designed in the future.

## Acknowledgements

We appreciate microbial samples provided by The Department of Ecological Engineering, Toyohashi University of Technology, Dr. Hiroyuki Futamata, Ms. Yoko Okubo, and others that have assisted in this study. This work was partially supported by Cooperation of Innovative Technology and Advanced Research in Evolutional Area.

## References

1. M. W. Griffiths, Rapid microbiological methods with hazard analysis critical control point, J AOAC Int. 1997 Nov-Dec;80(6):1143-50.
2. T. C. George, D. A. Basiji, B. E. Hall, D. H. Lynch, W. E. Ortyn, D. J. Perry, M. J. Seo, C. A. Zimmerman and P. J. Morrissey, Distinguishing modes of cell death using the ImageStream multispectral imaging flow cytometer, Cytometry Part A, 59A, 237-245, 2004.
3. Z. Fu, S. Rogelj and T. L. Kieft, Rapid detection of Escherichia coli O157:H7 by immunomagnetic separation and real-time PCR, Int J Food Microbiol. 2005 Mar 1;99(1):47-57.

4. M. Sugiura, M. Takano, S. Kawakami, K. Toda and S. Hanada, Application of a Portable Spectrophotometer to Microbial Mat Studies, *Microbes and Environments*, vol.16, No.4, 255-261, 2001.
5. T. Zimmerman, J. Pietodorf, R. Pepperkok, Spectral imaging and its applications in live cell microscopy, *FEBS Letters* 546, 87-92, 2003.
6. M. Sunamura, A. Maruyama, T. Tsuji and R. Kurane, Spectral imaging detection and counting of microbial cells in marine sediment, *J. of Microbiological methods*, 53 57-65, 2003.
7. A. P. Arkin, E. R. Goldman, S. J. Robes, C. A. Goddard, W. J. Coleman, M. M. Yang and D. C. Youvan, Applications of imaging spectroscopy in molecular biology II. Colony screening based on absorption spectra, *Bio/Technology*, vol.8, 746-749, August, 1990.
8. D. C. Youvan, Imaging sequence space, *Nature*, vol.369, 79-80, 5 May 1994.

# Dolphins Who's Who: A Statistical Perspective

Teresa Barata and Steve P. Brooks

Statistical Laboratory, Centre for Mathematical Sciences,  
Wilberforce Road, Cambridge, CB3 0WB, UK  
[{T.Barata, S.P.Brooks}@statslab.cam.ac.uk](mailto:{T.Barata,S.P.Brooks}@statslab.cam.ac.uk)  
<http://www.statslab.cam.ac.uk/~steve/>

**Abstract.** When studying animal behaviour and ecology the recognition of individuals is very important and in the case of bottlenose dolphins this can be done via photo-identification of their dorsal fins. Here we develop a mathematical model that describes this fin shape which is then fitted to the data by a Bayesian approach implemented using MCMC methods. This project is still at a testing stage and we are currently working with simulated data. Future work includes: extracting the outline of the fin shape from the pictures; fitting the model to real data; and devising a way of using the model to discriminate between individuals.

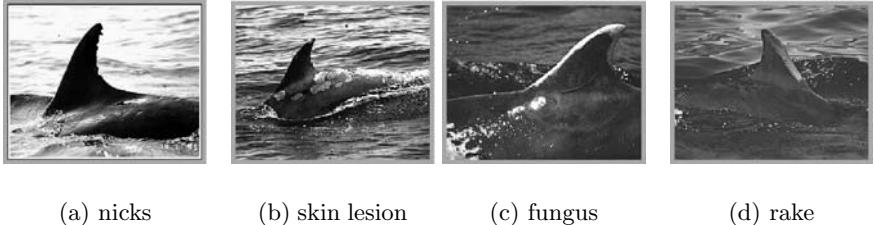
## 1 Introduction

For researchers of animal behaviour and ecology being able to identify individuals plays an important role in their studies and it has been shown that for most large long-living animals this can be done from natural marks. When species live underwater and are only visible for brief moments, individuals are photographed to provide a record of the encounter, which can be compared to a catalogue of pictures for identification purposes, see [9].

For bottlenose dolphins, the dorsal fin (especially its trailing edge) is the most identifying feature and individuals are identified by: the shape of the fin; shading of the fin, scrapes, scratches, nicks and wound marks; and pigment patterns such as fungus. A well-marked dolphin is one that is recognised by more than one single feature.

However there are many factors that complicate the photo-identification procedure: the difficulties in obtaining pictures, the ambiguous nature of the markings on the dolphins and the huge number of matching decisions are some examples. But, probably the biggest problem is the gradual loss of old non-permanent marks and the gain of new ones. As it would be impossible to link the two sets of pictures, the dolphin would then be classified as new a individual. Thus the same animal may appear in the database more than once.

In spite of this, photo-identification has been used on a wide variety of cetaceans, and a few studies now extend for periods of over twenty years. Moreover, the validity of photo-identification by natural markings has been confirmed by studies that combine this technique with tagging.



**Fig. 1.** Four examples of dolphin fins, courtesy of Paul Thompson, Lighthouse Field Station, Cromarty

In this paper we propose a statistical approach to identify individual dolphins via photographs of their dorsal fins. Section 2 gives a brief overview of the previous approaches to this problem. In section 3 we discuss the mathematical model developed to characterise the dolphin's fin shape. Section 4 is a brief review of Bayesian statistics and MCMC and in section 5 we apply these methodologies to the model in section 3. Some preliminary results using simulated data are presented and analysed in section 6. Finally, we give our conclusions and discuss future work in section 7.

## 2 Previous Work on Automatic Photo-Identification of Bottlenose Dolphins

The first attempt at developing an automated method for photo-identification of bottlenose dolphins was the Dorsal Ratio method explained in [3]. In this very simple method the top points of the two largest notches on each photograph are labelled A (top) and B (bottom). The Dorsal Ratio (DR) is then defined as being the ratio of the distance between A and B divided by the distance from B to the top of the fin. The catalogue is then examined for fins with similar dorsal ratios, to the one being identified. DR does not depend on the size of the fin and it can also handle moderate cases of parallax. However, it can only be used for dolphins with two or more notches and it may lack consistency as the locations of the tip and notches are defined by the user.

A computer-assisted approach to determine the Dorsal Ratio was implemented in [8]. In this case a digitised image constitutes the input to the system, followed by an edge detection module to generate the edge of the dolphin's dorsal fin of which only the trailing section is considered. The user selects the start and end points of the fin's trailing edge and is also required to prune false notches. This edge can be uniquely represented by its curvature function and is used to automatically extract the Dorsal Ratio.

A more sophisticated approach is the string matching method described in [1]. Edge detection and curvature description are done in exactly the same way as in the previous method, but this time the curvature function, or string, is used

to identify each dolphin. As in the previous method only the trailing section is considered. To identify an individual, its string representation is aligned pairwise with the ones in the database and a distance measure between the two is used to assess the degree of matching. This has the disadvantage of only taking into account the trailing edge of the fin shape, and relying on notches to identify dolphins. This method has been coded in a user-friendly graphical interface software called CEDR (Computer Extracted Dorsal Ratio) that can be downloaded from: <http://ee.tamu.edu/~mckinney/Dolf.html>.

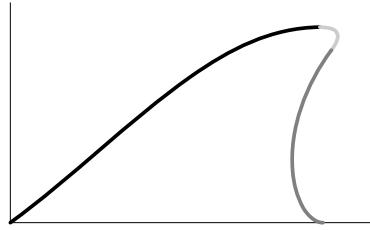
Another option is DARWIN, a computer vision application developed to assist researchers with photo-identification of dolphins. It starts by using an edge detection algorithm followed by a preprocessing step which allows for the manipulation of the curve in three dimensions (to account for the angle at which the dolphin might have been photographed). First, the centroid of the outline (i.e., the centre of the fin shape, see [4]) is calculated. Next, the distance from the centroid to the points along the outline is found and plotted, in what is known as a signature. The new fin is matched against each database fin at relative rotations to allow for dolphins that were surfacing or diving. Also the position of the centroid is affected by the user designation of the beginning and ending of the outline, so a process that corrects the positions of the centroids was also implemented. When matching is complete, the database fins are presented in rank order of their mean squared error. DARWIN deals with the fact that the dolphins might have been photographed at an angle, but it does so by letting the user estimate this angle in a preprocessing step. This has the disadvantage of being user dependent and the estimated angle can not be changed further during the analysis. DARWIN can be downloaded from: <http://darwin.eckerd.edu/>

Recently the mismatch area method was introduced, see [7], which is an affine invariant curve matching technique for photo-identification of marine mammals based on silhouettes. The process starts by carrying out edge detection and the output curve is smoothed using cubic B-splines. Subsequently, this curve is matched to the ones in the database by using an affine transformation to force both curves to overlap as closely as possible. Having done so, the mismatch area is computed and the database curves are ranked accordingly.

### 3 A Model for the Fin Shape

Although automated photo-identification of dolphins usually relies on the position of nicks and notches on their dorsal fins, we propose the use of the overall fin shape instead. The advantages are the following: the overall fin shape does not change even when new marks are acquired, but most importantly, it is a very good way of identifying poorly marked dolphins (which is crucial in areas where dolphins lack predators). As it is quite difficult to distinguish different fin shapes “by eye”, a parametric curve can be used to characterise them.

In this section we introduce a parametric model for the dorsal fin shape of bottlenose dolphins. As there is not a single parametric line that describes this



**Fig. 2.** Example of a fin shape curve

shape accurately, we use segments from three curves, matching their start and end points, as well as, their first derivatives to get a smooth curve.

The three curves chosen were: a Gaussian shaped exponential, which models the back of the fin; a parabola, to model the tip of the fin and a logarithmic spiral, that models the trailing edge of the fin. The model has a total of eight parameters that must be estimated and seven fixed parameters, to do the matching between the curves, as follows:

### Exponential:

Parameters to be estimated:  $E_1$  and  $E_2$ .

$$\begin{cases} x_e(t) = t \\ y_e(t) = \exp(-(t - E_1)^2 / E_2) - \exp(-E_1^2 / E_2) \end{cases} \quad t \in [0, E_1] \quad (1)$$

### Parabola:

Parameters to be estimated:  $P_1$  and  $P_2$ .

$$\begin{cases} x_p(t) = -\frac{\cos(\theta)}{P_1} \frac{(E_1 + 1 - t)^2}{(E_1 + 1 - t)^2} - \frac{\sin(\theta)}{P_1 P_2} \frac{(E_1 + 1 - t)}{(E_1 + 1 - t)} + a \\ y_p(t) = -\frac{\sin(\theta)}{P_1} \frac{(E_1 + 1 - t)^2}{(E_1 + 1 - t)^2} + \frac{\cos(\theta)}{P_1 P_2} \frac{(E_1 + 1 - t)}{(E_1 + 1 - t)} + b \end{cases} \quad t \in [E_1, E_1 + 2] \quad (2)$$

### Logarithmic spiral:

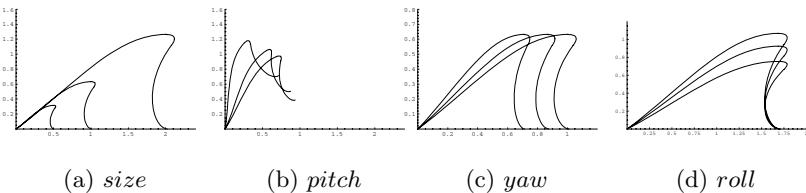
Parameters to be estimated:  $S_1$ ,  $S_2$ ,  $S_3$ , and  $S_4$ .

$$\begin{cases} x_s(t) = g \sin(S_1 (E_1 + 2 + 2\pi - t)) \exp(S_2 (E_1 + 2 + 2\pi - t)) + d \\ y_s(t) = g c \cos(S_3 (E_1 + 2 + 2\pi - t)) \exp(S_4 (E_1 + 2 + 2\pi - t)) + f \end{cases} \quad t \in \left[ E_1 + 2, E_1 + 2 + \frac{3\pi}{4} \right] \quad (3)$$

This gives rise to the shape in Fig. 2, where the exponential curve is given in black, the parabola in light grey and finally the logarithmic spiral in dark grey.

### 3.1 Angles and Size

The above model must then be fitted to the outline of the fin extracted from the photographs. Hence four more parameters need to be included. These are *size*



**Fig. 3.** The model curve for different values of the parameters *size*, *pitch*, *yaw* and *roll*

and the angles: *pitch*, if the dolphin is diving or emerging; *yaw*, that takes into account the angle between the camera and the dolphin; and *roll* if the dolphin is rolling on its side. The updated model is given in equation 4,

$$\begin{cases} x(t) = size [x_{\cdot}(t) \cos(pitch) - y_{\cdot}(t) \sin(pitch)] \cos(yaw) \\ y(t) = size [x_{\cdot}(t) \sin(pitch) + y_{\cdot}(t) \cos(pitch)] \cos(roll) \end{cases} \quad (4)$$

where the subscript is  $e$ ,  $p$  or  $s$  according to the value of  $t$  and equations 1, 2 and 3. Fig. 3 shows how these parameters affect the curve given in Fig 2.

The inclusion of these additional parameters has the advantage of dealing with the angles in the photograph, while the shape parameters are still being estimated. However, it is now possible for two different fin shapes (i.e. with different shape parameters) to look very similar when one of them has been rotated, making it impossible for the model to distinguish between these two cases. In any case, our goal is not to find a perfect match for the dolphin in the photograph, but to give a ranking of the most likely individuals. Hence, the two fin shapes being discussed would both be considered.

## 4 Bayesian Statistics and MCMC

As we use Bayesian statistics and MCMC to fit the model discussed in the last section, next we give a brief overview of these methods.

Suppose we are fitting a model with parameters  $\theta$  to a dataset  $x$ . The Bayesian statistics approach allows us to combine information from this dataset with expert prior information on the parameters, see for example [5] for a full account of the Bayesian approach. Both sources of information have to be summarised as probability distributions, these are the likelihood function  $L(\theta; x)$  for the data and the prior distribution  $p(\theta)$  for the expert information. Bayes' theorem then combines these to obtain the posterior distribution,

$$p(\theta|x) \propto L(\theta;x)p(\theta) \quad (5)$$

Inference on  $\theta$  usually involves integrating  $p(\theta|x)$  to get estimates of its expected value and variance, for example. Often these integrals are either too

complicated to be done explicitly or  $p(\theta|x)$  is only known up to proportionality. In this case, Markov Chain Monte Carlo (MCMC) methods can be used to sample from  $p(\theta|x)$  and obtain sample estimates of the quantities of interest. An overview of the MCMC methods is given in [2].

MCMC is a generalisation of the Monte Carlo simulation methods and is used whenever it is impossible to sample from the posterior distribution directly. In this case a Markov chain with stationary distribution  $p(\theta|x)$  is used as an indirect sampling method. Thus after a large enough number of iterations has been simulated, and under certain regularity conditions, these values can be treated as a sample from  $p(\theta|x)$ . In practice, long simulations are run and iterations within an initial transient phase or burn-in period are discarded.

There are many important implementational issues associated with Bayesian statistics and MCMC. Some of these are technical (choice of priors and convergence diagnosis, for example) but most of them are problem dependent. A good reference is [6] which has a chapter on Markov Chain Monte Carlo and Image Analysis that gives an overview of a variety of image models and the use of MCMC methods to deal with them. It also reviews some of the methodological innovations in MCMC stimulated by the needs of image analysis.

## 5 Fitting the Model to the Data

In order to use the statistical framework discussed in the last section a likelihood function and priors for the parameters are needed. We assume that the data, i.e. the coordinates for each pixel on the edge of the fin, follows a bivariate normal distribution centred on the model values for these coordinates which were defined in equation 4. That is,

$$(x_i, y_i) \sim N\left((x(t_i), y(t_i)), \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}\right), \quad i = 1, \dots, k \quad (6)$$

where  $k$  is the number of pixels on the edge of the fin and  $\sigma^2$  models the edge detection and pixelisation errors and is a parameter to be estimated.

As for the prior densities, for the shape parameters we chose the following uniform priors to make sure the resulting curve would look like a fin shape:

$$\begin{aligned} p(E_1) &= \text{Unif}\left([0, 3/\sqrt{2 E_2}]\right), \quad p(E_2) = \text{Unif}([0, 10]) \\ p(P_1) &= \text{Unif}([10, 70]), \quad p(P_2) = \text{Unif}([0.5, 2]), \quad p(S_1) = \text{Unif}([0.7, 1.3]) \\ p(S_2) &= \text{Unif}([0, 2.5]), \quad p(S_3) = \text{Unif}([0.7, 1.3]), \quad p(S_4) = \text{Unif}([0, 3]) \end{aligned} \quad (7)$$

With respect to the nuisance parameters, the prior for *size* was chosen to depend upon the width and height of the image (respectively  $n$  and  $m$ ) as the outline of the fin should not be too small and the entire outline has to be within the image. As for the angles, *pitch* does not change the shape of the dolphins' fin and its prior takes that into account by allowing quite big variations of this

parameter. The same is not true for *yaw* and *roll*, and only images that, “by eye”, have both angles equal to zero, are entered in the database. If the dolphin’s right side has been photographed, and not the left as in Fig. 2, *yaw* will be close to  $\pi$ , and not zero. In our case, this problem is dealt with in a preprocessing step where the user selects the side of the dolphin being considered. This is common practice in marine biology, and usually two separate databases are kept for each side. To summarise, we have chosen the following priors for these parameters,

$$\begin{aligned} p(\text{size}) &= \text{Unif} [1/64\sqrt{n \times m}, \sqrt{n \times m}], \quad p(\text{pitch}) = \text{Unif} [-\pi/3, \pi/3] \\ p(\text{roll}) &= \text{Unif}[0, \pi/6] \\ p(\text{yaw}) &= \begin{cases} \text{Unif}[0, \pi/6], & \text{if side = left} \\ \text{Unif}[5\pi/6, \pi], & \text{if side = right} \end{cases} \end{aligned} \quad (8)$$

As for  $\sigma^2$ , as there is no prior information, we will assume a non-informative positive prior  $\text{Gamma}(\epsilon, \epsilon)$ , with  $\epsilon$  small. Putting this together with equations 6, 7 and 8 we get the following posterior density function ( $\theta$  represents the shape parameters;  $\mu$ , *size* and the angles; and  $(x, y)$  the data),

$$p(\sigma^2, \theta, \mu | (x, y)) \propto \frac{1}{\sigma^{2k}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^k ((x_i - x(t_i))^2 + (y_i - y(t_i))^2) \right\} \quad (9)$$

$$p(\sigma^2) \ p(\theta) \ p(\mu)$$

This is an extremely complicated posterior distribution as the model points  $(x(t_i), y(t_i))$ , defined in equation 4, depend on the parameters in quite a non-linear way. We used MCMC, namely the Gibb’s sampler algorithm, in order to do the estimation. The technical details for this algorithm can be found in [2], but the idea is as follows: as the posterior distribution is too hard to work with, a random sample is simulated, assuming at each step that only one of the parameters is a random variable and the others are fixed, that is, we use their posterior conditional distributions. Unfortunately, even in this case, for most parameters (*size* is the exception) we need to sample from a non-standard distribution which is only known up to proportionality. Thus we must use the Metropolis-Hastings algorithm as an indirect sampling method, see [2] for a full account on this methodology. All the parameters were simulated in a similar way, hence details are given for  $E_1$ , as an example.

The conditional posterior distribution for  $E_1$  is

$$\begin{aligned} p(E_1 | \sigma^2, \theta \setminus \{E_1\}, \mu, (x, y)) &\propto \\ \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^k ((x_i - x(t_i))^2 + (y_i - y(t_i))^2) \right\}, \quad E_1 &\in \left[0, 3/\sqrt{2 E_2}\right] \end{aligned} \quad (10)$$

And the Metropolis-Hastings algorithm works as follows. Suppose that at a given step the current value of the Markov chain is  $E_1$ . A proposed new value  $E'_1$  is simulated from the auxiliary distribution defined below.

$$E'_1 \sim q(E_1, E'_1) = N(E_1, \sigma_{E_1}^2) \quad (11)$$

If  $E'_1 \notin [0, 3/\sqrt{2} E_2]$  it is promptly rejected and the old value  $E_1$  is kept. Otherwise, the probability of  $E'_1$  being accepted is given by,

$$\begin{aligned} \alpha(E_1, E'_1) = \min \left\{ 1, \frac{p(E'_1 | \cdot)q(E'_1, E_1)}{p(E_1 | \cdot)q(E_1, E'_1)} \right\}, \text{ where } \frac{p(E'_1 | \cdot)q(E'_1, E_1)}{p(E_1 | \cdot)q(E_1, E'_1)} = \\ \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^k ((x_i - x'(t_i))^2 - (x_i - x(t_i))^2 + (y_i - y'(t_i))^2 - (y_i - y(t_i))^2) \right\} \end{aligned} \quad (12)$$

Hence  $E'_1$  is more likely to be accepted if its conditional posterior is higher than that of  $E_1$  under the current values of the other parameters. The above steps are repeated until a large enough number of values has been simulated.

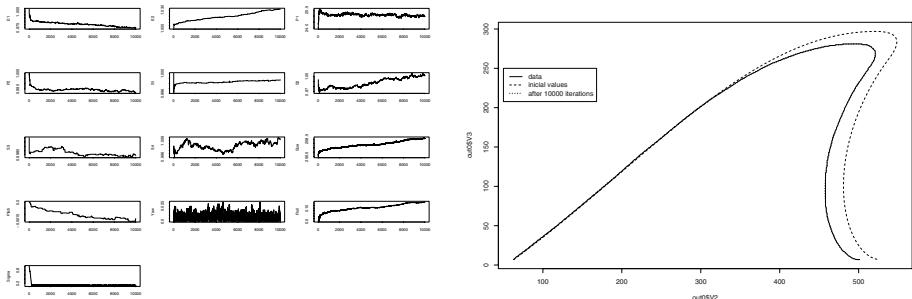
The convergence rate of the chain depends heavily on  $\sigma_{E_1}^2$  and this must be small enough so that there is a fair chance  $E'_1$  will be accepted but big enough so that different values of  $E_1$  are explored. In this very preliminary stage of our work, while still working with simulated data, we have chosen  $\sigma_{E_1}^2$  to be 0.005 on the basis of pilot tuning.

## 6 Preliminary Results Using Simulated Data

The use of simulated data has the advantage of the true parameter values being known and thus it is a good way to verify the estimation methods described above.

We used the black and white version of Fig. 2 as a simulated curve and transformed it into a JPEG file. This file is scanned into the program that extracts the coordinates of the black pixels and fits the model to them. Fig. 4 provides plots of some preliminary results.

In Fig. 4 (b), it seems the model is fitting the data extremely well, as after 10,000 iterations the model curve is indistinguishable from the data. However, these simulations were based upon starting points chosen to be the true values of the shape parameters. Future work will include running simulations with different starting values. On the other hand, the traceplots given in Fig. 4 (a) show that some of the parameters are highly correlated. For example, the exponential parameters  $E_1$  (the left most graph on the first line) and  $E_2$  (the graph to its right) have inverse behaviours, when  $E_1$  decreases,  $E_2$  increases. These high correlations imply that if the estimated value for one of the parameters is wrong the other parameters are able to compensate so that the overall shape will not be affected. This is an extremely undesirable behaviour, and we are currently looking at ways of making the parameters more independent.



(a) traceplots for the simulated values of the parameters

(b) model curves

**Fig. 4.** Preliminary results

## 7 Discussion and Future Work

In this paper we have presented a mathematical model for the fin shape of bottlenose dolphins which also takes into account the angles at which this shape is viewed from. We have also shown preliminary results of fitting the model to simulated data, however this is still very much work on progress and although our results are promising much is still to be done. Future developments include the following. (a) Working with real data. (b) An edge detection algorithm, which ideally would take into account the uncertainty in extracting the outline of the fin. Hence instead of the output being a single curve, for each point we would have the probability of it being on the edge. An alternative would be to deal with this uncertainty while fitting the model, which is the approach we followed in this paper. (c) Devising a way of using the model parameters to compare fin shapes and hence identify individuals. This can be achieved by using Reversible Jump MCMC. The idea is very similar to the Metropolis-Hastings method explained earlier. Suppose at a given iteration it is assumed that the new dolphin is dolphin A in the database. We then propose this new dolphin to be dolphin B, say. This move is accepted with a given probability calculated in a similar way to equation 12. The database dolphins can then be ranked with respect to the proportion of iterations where it was assumed they were the new dolphin. (d) Finally we also wish to compare our method with other available alternatives.

## Acknowledgements

The first author is sponsored by Fundação para a Ciência e Tecnologia with the scholarship SFRH/BD/8184/2002.

## References

1. B. Araabi, N. Kehtarnavaz, T. McKinney, G. Hillman and B. Würsig, A String Matching Computer-Assisted System for Dolphin Photoidentification , *Annals of Biomedical Engineering*, 2000, vol. 28, pp. 1269-1279.
2. S. P. Brooks, Markov Chain Monte Carlo method and its applications, In: *The Statistician*, 1998, vol.47, pp. 69-100.
3. R. H. Defran, G. M. Shultz, and D. W. Weller, A Technique for the Photographic Identification and cataloguing of dorsal Fins of the Bottlenose Dolphin *Tursiops truncatus*, In: *Individual Recognition of Cetaceans: Use of Photo Identification and other Techniques to Estimate Population Parameters*, Report of the International Whaling Commission, edited by P.S. Hammond, S.A.Mizroch and G.P. Donovan. Cambridge: Cambridge University Press, 1990, vol. 12, pp. 53-55.
4. I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*, John Wiley, 1998.
5. A. Gelman, J. B. Carlin, H. S. Stern and D. R. Rubin *Bayesian Data Analysis*, Chapman & Hall, 1995. .
6. P. J. Green, MCMC in Image Analysis, In: *Markov Chain Monte Carlo in practice*, W. Gilks, S. Richardson and D. J. Spiegelhalter (eds.), Chapman & Hall,1996, pp. 381-399.
7. C. Gope, N. Kehtarnavaz, G. Hillman and B. Würsig, An affine invariant curve matching method for photo-identification of marine mammals, In: *Pattern Recognition*, 2005, vol.38, pp. 125-132.
8. A. Kreko, N. Kehtarnavaz, B. Araabi, G. Hillman, B. Würsig and D. Weller, Assisting Manual Dolphin Identification by Computer Extraction of Dorsal Ratio”, In: *Annals of Biomedical Engineering*, 1999, vol. 27, pp. 830-838.
9. B. Würsig and T. Jefferson, Methods of Photo-Identification for Small Cetaceans, In: *Individual Recognition of Cetaceans: Use of Photo Identification and other Techniques to Estimate Population Parameters*, Report of the International Whaling Commission, edited by P.S. Hammond, S.A.Mizroch and G.P. Donovan. Cambridge: Cambridge University Press, 1990, vol. 12, pp. 43-51.

# Local Shape Modelling Using Warplets

Abhir Bhalerao and Roland Wilson

Department of Computer Science, University of Warwick, UK  
`{abhir, rgw}@dcs.warwick.ac.uk`

**Abstract.** We develop a statistical shape model for the analysis of *local* shape variation. In particular, we consider models of shapes that exhibit self-similarity along their contours such as fractal and space filling curves. Overlapping contour segments are parametrically modelled using an orthogonal basis set, Legendre Polynomials, and used to estimate similarity transformations to a reference segment, which may or may not be from the contour being analysed. The alignment is affine and regresses the model to the data by least squares fitting and is followed by a PCA of the coregistered set of contour segments. The local shape space is defined jointly by the segment-to-segment ‘warps’ and the mean plus eigen vectors of the shape space, hence Warplets. The parametric modelling makes the alignment correspondence-free so that arbitrary sized segments can be aligned and the local warps can be inverted to reconstruct model approximations of the data. The approach shows potential in capturing fine details of shape variation and is applicable to complex shapes and those with repetitive structure, when only a few training examples are available.

## 1 Introduction

Statistical shape models are built from a set of training examples and, by Principal Component Analysis (PCA), the eigen modes of the model represent the most likely variations of the shape [1]. Such models have been used in object segmentation to constrain the variation of a deformable template [2]. Two important issues normally arise with such models: the size of the training set and the need for point-to-point correspondences in the training samples (homology). The former problem exacerbates the latter as it is tedious to hand label homologous points for larger training samples and complex shapes need greater numbers of points. Much work has focussed on these problems in recent years, e.g. [3, 4]. However, for certain classes of shapes where there are periodic variations, e.g folding patterns of cortical geometry, these global or parameter models ([5]) are inadequate or become impractical to use. Many ‘active’ shape model (ASM) based segmentation methods therefore defer to the physical constraints for the final refinement steps when registering a deformable template to such a class of shapes. Hybrid shape models have been proposed that attempt to incorporate models of shape variation into the global scheme. For example, multiresolution analysis by wavelet

transforms combined with fractal priors [5] or with a PCA [4], or curve evolution based on scale-spaces [6]. These models strive to separate the fine detail of the shape variation from the large scale changes to leverage either stability for the physical deformations or greater power in the lower modes of variation.

Here we describe a locally parameterised statistical shape model, the contour Warplet. By explicitly modelling each segment with Legendre polynomials, the affine pose estimation can be performed by regression. The warplets are subsequently decomposed by PCA in the normal way and by retaining the pose parameters, we can reconstruct the entire contour to measure the segment-to-segment variation.

## 2 Method

### 2.1 Localising the Contour

A contour,  $C$ , can be described as a set of  $N$  points in the plane,  $\mathbf{c}_i = (x_i, y_i)^T$ ,  $0 \leq i < N - 1$ . Without loss of generality, we consider point sets with sizes in powers of two;  $N = 2^e$ , e.g.,  $N = 512$ . The contour can be divided into equal sections of size,  $B = 2^f$ , such that  $2 \leq f < e$ , made to overlap by half their lengths  $B/2$ . For example, a contour of size 512 can be divided into 32 overlapping segments of size  $B = 32$ . Let segment  $j$  be denoted by  $S_j = \{\mathbf{c}_i\}$ ,  $jB/2 \leq i \leq (j + 1)B/2$ . If the final segment is made to straddle the  $B/2$  points at the end and the first  $B/2$  points in the start of  $C$ , the contour is regarded as being closed.

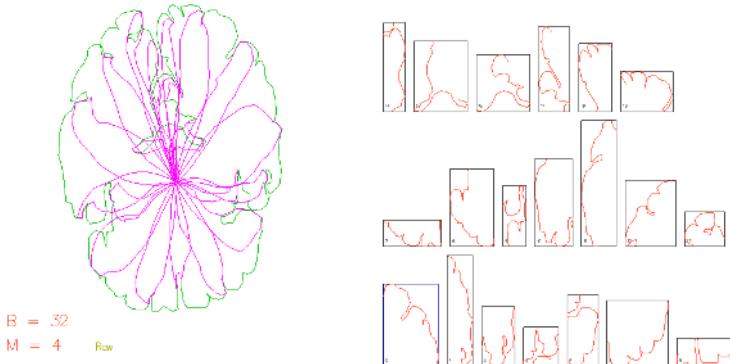
To reconstruct the contour, the segment coordinates are windowed by a  $\cos^2()$  function centred on the middle point and the windowed coordinates of the overlapping sections,  $SW_j$ , are simply summed in to the output. For  $n_s$  segments, the reconstructed contour is calculated by

$$C = \sum_{j=0}^{n_s-1} \{\cos^2((i - B/2)\frac{\pi}{2})\mathbf{c}_i\}_j. \quad (1)$$

The reason this works is that the window function shifts by  $\frac{\pi}{2}$  for each half segment, hence  $\cos^2()$  becomes  $\sin^2()$  and in the overlap region points proportionally sum to 1, Figure 1.

### 2.2 Contour Modelling

Each contour segment,  $S_j$ , is parameterised in polar form transforming points  $\mathbf{c}_i$  to  $\mathbf{p}_i = (\rho_i, \theta_i)^T$ , where  $\rho_i = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}$  and  $\theta_i = \tan^{-1}((y_i - y_o)/(x_i - x_o))$ . If the contour is centred on the origin, i.e.  $\sum_i \mathbf{c}_i = \mathbf{0}$ , then  $(x_o, y_o)^T = \mathbf{0}$  and the polar parameters represent the radial and angular variations of the curve around the origin. Polar coordinates are convenient for the subsequent modelling as the contours of interest will be more or less circular.



**Fig. 1. Localisation of contour into segments of size  $B = 32$ .** The original contour is shown in green on the left-hand images together with the windowed segments which form loops centred on the origin in magenta. The segmented sections are shown in the right-hand window

The Legendre Polynomials (LP) derive from solving Legendre's differential equation and are defined for the range,  $x \in [-1, 1]$  as follows:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n \quad (2)$$

where  $n$  is the polynomial order.

$$P_0(x) = 1, \quad P_1(x) = 1, \quad P_2(x) = \frac{1}{2}(3x^2 - 1), \quad P_3(x) = \frac{1}{2}(5x^3 - 3x). \quad (3)$$

These functions form an orthogonal basis set in the interval  $-1 \leq x \leq 1$  since

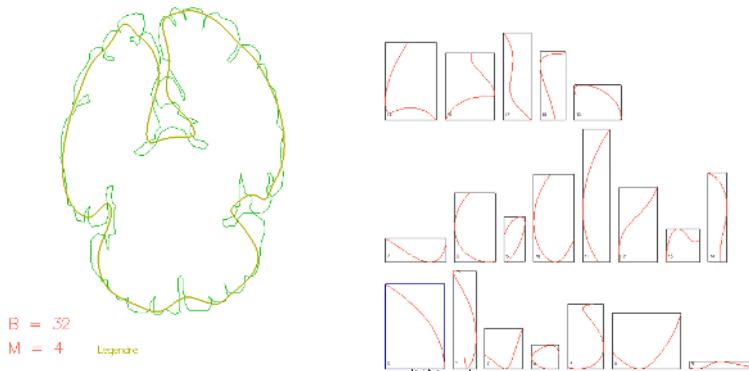
$$\int_{-1}^1 P_m(x) P_n(x) dx = 0, \quad m \neq n. \quad (4)$$

Note that these polynomials can be efficiently generated by the recurrence formula:

$$P_{n+1}(x) = \frac{(2n+1)xP_n(x) - nP_{n-1}(x)}{(n+1)}, \quad (5)$$

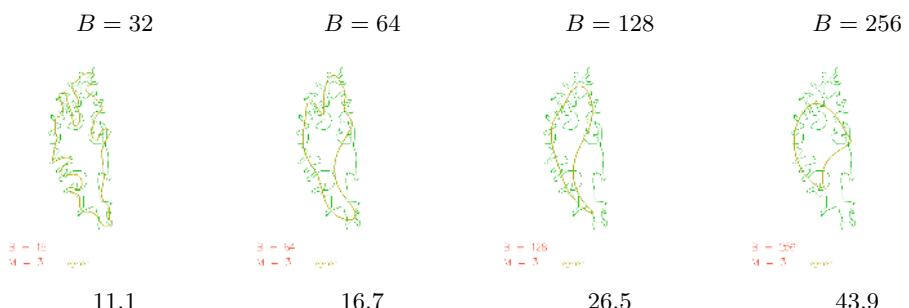
and fitting a function, say  $f(x)$ , by  $(n+1)$  coefficients can be accomplished by projection onto the basis set. We model all contour segments,  $S_j$ , by fitting polynomials of a given order  $M$ , to the radial and angle functions of the polar parameterisation functions of each segment. For example, the radial function is approximated by the weighted sum as follows:

$$\hat{\rho}(x) = \sum_{m=0}^{M-1} l_m P_m(x), \quad \rho(x) = \rho_i, \quad x = (i - \frac{B}{2}) / (\frac{B}{2}) \quad (6)$$



**Fig. 2. Examples of Legendre Polynomial models of contour segments for  $M = 4$ . The yellow contour is the model reconstruction. The right-hand images show the individual model segments for  $B = 32$**

where  $l_m$  result from the linear regression. The angle function  $\hat{\theta}(x)$  is similarly modelled. Figure 2 illustrates the LP contour modelling for the two previous example contours for  $M = 4$ . The right-hand images show the estimated model for segments of size  $B = 32$  and the yellow contour on the left-hand image is the model reconstruction using overlapping weighted segments. In these examples, each contour segment is coded by  $2(M + 1)$  coefficients plus the centre of arc coordinates ( $\mathbf{0}$ ). In figure 3 model reconstruction for a range of segment sizes for  $M = 3$  are shown with errors given in RMS point distances under each figure. Lengthening the segments for a given order smoothens the shape representation in quite a natural way and is reminiscent of curve-evolution approaches using scale-spaces. The LP representation is fundamentally free of the number of points in each segment. For all the examples presented here, we set  $B$  to be the same size for each segment.



**Fig. 3. Effects of segment length  $B$  for a given LP model order ( $M = 3$ ).** Examples show results on a convex white matter contour of size approximately  $256 \times 512$ . RMS contour position errors are given below each figure

### 2.3 Procrustes Registration

The analysis of shape variation across the contour segments requires them to be registered to a common coordinate frame: the Kendall Shape Space [7]. By registration, the segments are corrected for position, rotation and scale variation. Here, an affine warp is estimated by transforming the modelled segments  $\hat{S}_j$  to some prototype segment  $S_k$ . Dryden [7] notes that affine matching using regression reduces to linear regression for any number of dimensions. Thus we can satisfactorily separate the 2D pose alignment problem into two, 1D regressions for the polar coordinate functions  $\theta_j(x), \rho_j(x)$  of each segment,  $S_j$ . Let the warped segment be  $W_j$ , then define warps of the form:

$$W_j = T_A(\hat{S}_j) = \begin{pmatrix} a_{2\rho} + a_{1\rho}x & 0 \\ 0 & a_{2\theta} + a_{1\theta}x \end{pmatrix} \hat{\mathbf{p}}_i + \begin{pmatrix} a_{0\rho} \\ a_{0\theta} \end{pmatrix} \quad (7)$$

The warps,  $S_j \rightarrow S_k$ , from all segments to a chosen prototype segment,  $S_k$  are sought. These are each represented by 6 transformation parameters,  $A_{jk} = (\mathbf{a}_\rho, \mathbf{a}_\theta)$ . For example, the radial model function for segment  $j$  is transformed by

$$W_{j\rho} = (a_{2\rho} + a_{1\rho}x)\hat{\rho}(x) + a_{0\rho} \quad (8)$$

As with the LP model fitting, this yields a system of equations for each model point in  $\hat{S}_j$  against a target data point in the prototype segment,  $S_k$  and the system can be solved by SVD. It is perhaps useful to briefly comment on the effects of some simple similarity transformations on a typical segment. For a rotation of a segment, we require an estimate only of an angular displacement,  $a_{0\theta} \neq 0$ . A constant radial scaling will be represented by  $a_{1\rho}$  with  $a_{2\rho} = 0$ , while a linear scaling across the segment will be accounted for by a non-zero term:  $(a_{2\rho} + a_{1\rho}x)$ .

To transform a segment back to the contour space, we trivially apply the inverse mapping,

$$T_\rho^{-1}(W_{j\rho}) = \frac{(W_{j\rho} - a_{0\rho})}{(a_{2\rho} + a_{1\rho}x)}, \quad (9)$$

and similarly for the warped segment angle function using  $T_\theta^{-1}$ .

### 2.4 Warplets and Affine Shape Spaces

Having registered all contour segments to a reference segment,  $S_k$ , a PCA is used in the standard way to encode the shape variation. Denoting the decomposition as

$$\hat{W}_j = M_k + \sum_{d=1}^D \Gamma_{jd} \cdot \Phi_{kd}, \quad (10)$$

where  $M_k$  is the mean warplet and  $\Gamma_j$  are the weighting for each of the  $D$  eigen vectors,  $\Phi_k$ , associated with non-zero eigen values. The contour warplet is defined jointly by the regression parameters of the Procrustes alignment,  $A_{jk}$ , and the coordinates of the segment in the tangent shape space:

$$\mathcal{W}_{jk} = \{A_{jk}, \Gamma_{jk}\}. \quad (11)$$

If the segments are evenly distributed around the mean, then the tangent space distribution is even also and should be independent of the registration. However, a full Procrustes analysis requires a pair-wise registration between shapes,  $S_j, S_k$  for all  $j, k$ . Thus, the choice of the prototype shape,  $S_k$  for  $\mathcal{W}$  is debatable, but iterative optimisation is possible (e.g. as in [8]).

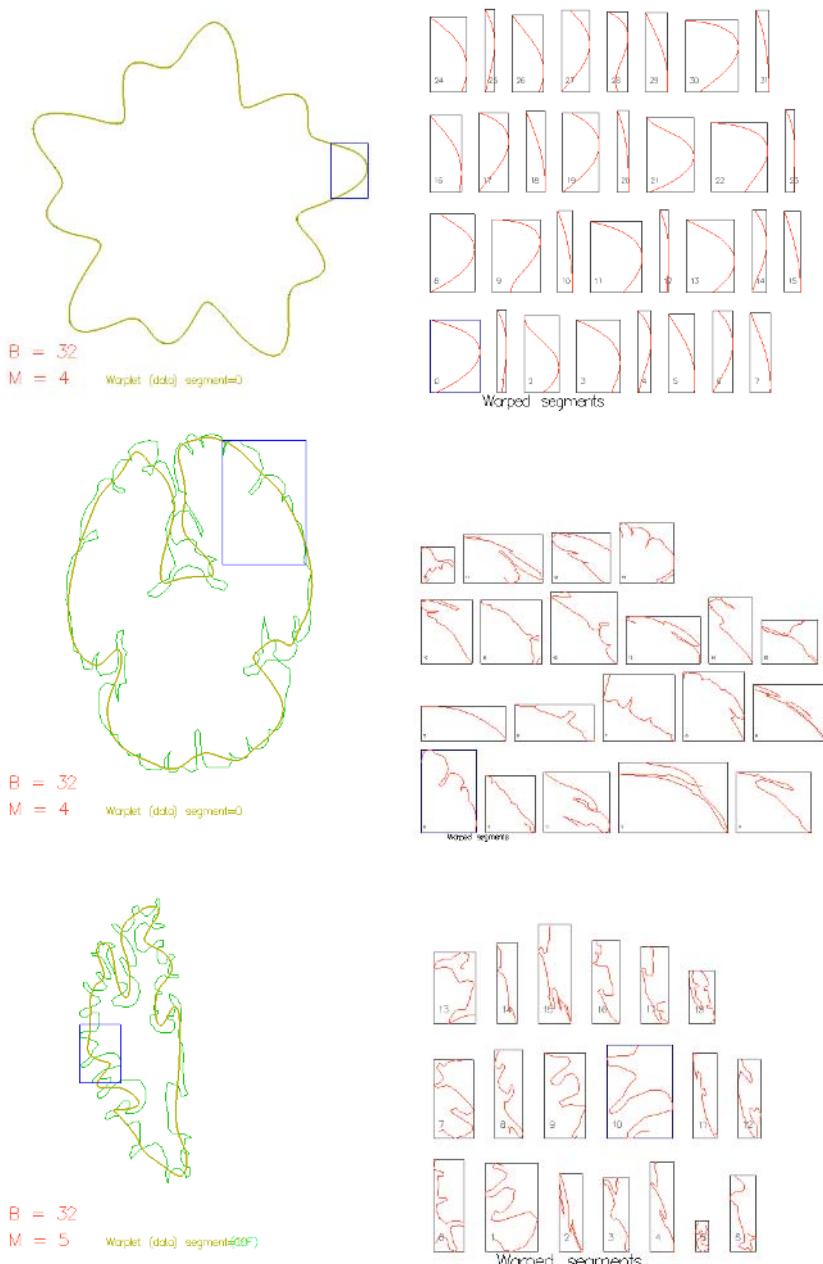
### 3 Experiments

Figure 4 shows three examples of warplet shape spaces for the synthetic ‘wavy’ curve, a grey-matter brain contour hand-drawn from a slice of an MR brain data set, and a hand-drawn white-matter contour. All contours were analysed using 32 point segments starting at an arbitrary position and modelled with 4 Legendre Polynomials per segment  $M = 4$ , i.e. a cubic fit. The blue-box on the left-hand images shows the bounds of the prototype segment onto which all others were warped; warped segments are shown together with their extents on the right-hand images (these images have been rescaled). Each of these warped segments become input shapes to a PCA analysis. In the top two examples, the scaling and rotation of the segments can be seen to roughly match the prototype. On the white-matter contour, the alignment results are harder to interpret. After PCA, we can plot the sizes of the principal non-zero eigen values and, in figure 5, we have plotted these for PCA without alignment and with alignment to the segment that produced the most ‘compact’ shape space for the white-matter example, i.e. the plot which gave the least variation from the mean.

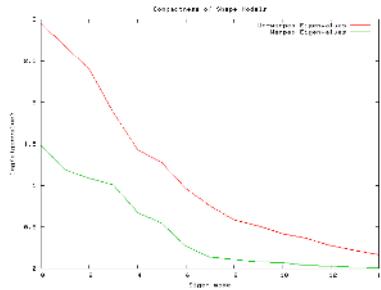
In figure 6, the first few principal modes,  $d < D$ , have been plotted for the grey-matter contour (middle row figure 4) and shown in proportions of  $\sqrt{\lambda_d}$  around each mean,  $M_k$ . The mean is in the first box on each row and then the modes in order across the page. As expected, most of the variability is in the first 4 or 5 modes around 2 standard-deviations around the mean. Some of the finer contour variation is apparent in the higher modes, whereas the lower modes show largely elongation and shrinking not accounted for by the warping. Finally, in figure 7 we reconstruct the contour using equation (9) after taking increasing numbers of modes up to  $D$ . Next to each result is a RMS point error between the shape model approximation (shown in black) and the original contour (shown in green). Where the segments deviate in shape from the prototype (shown in the blue box on each image), we see that the contour reconstruction curve lies far from the true contour. It is only beyond mode  $d = 4$  that the reconstruction errors fall below about 10 pixels.

### 4 Conclusions

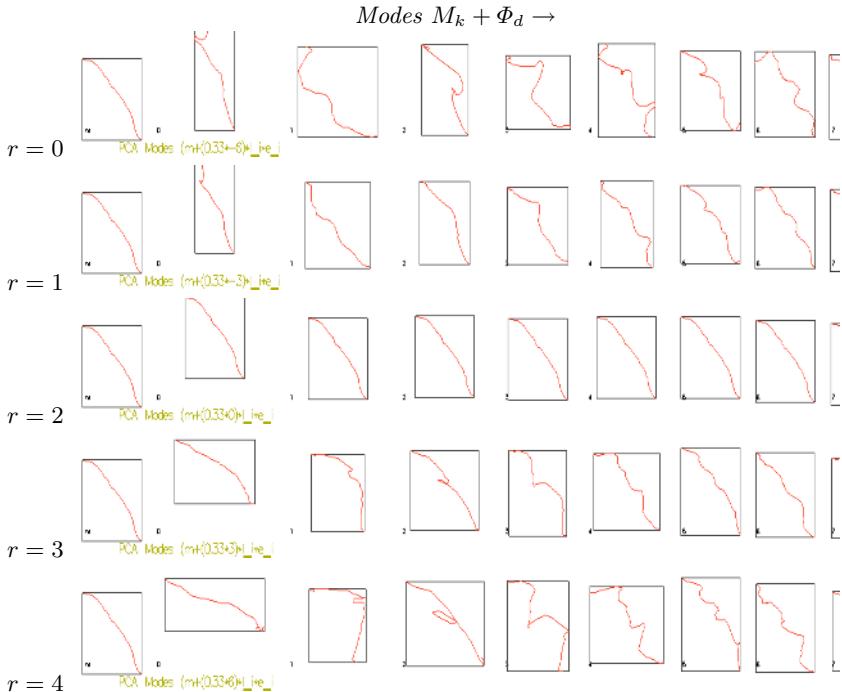
We have described a new statistical shape model for local contour modelling and analysis, contour warplets, which has some unique properties. Our experiments thus far have been limited to a small set of arbitrarily chosen curves, some



**Fig. 4. Contour segments after alignment.** The rectangles marked in blue on the left-hand image shows the bounding-box of the prototype segment against which all are registered. Registered segments shown in right-hand images (the order of presentation is left-to-right, bottom-to-top). White-matter contours (bottom two rows) have been parameterised from estimated centres-of-arc (see 2.1)

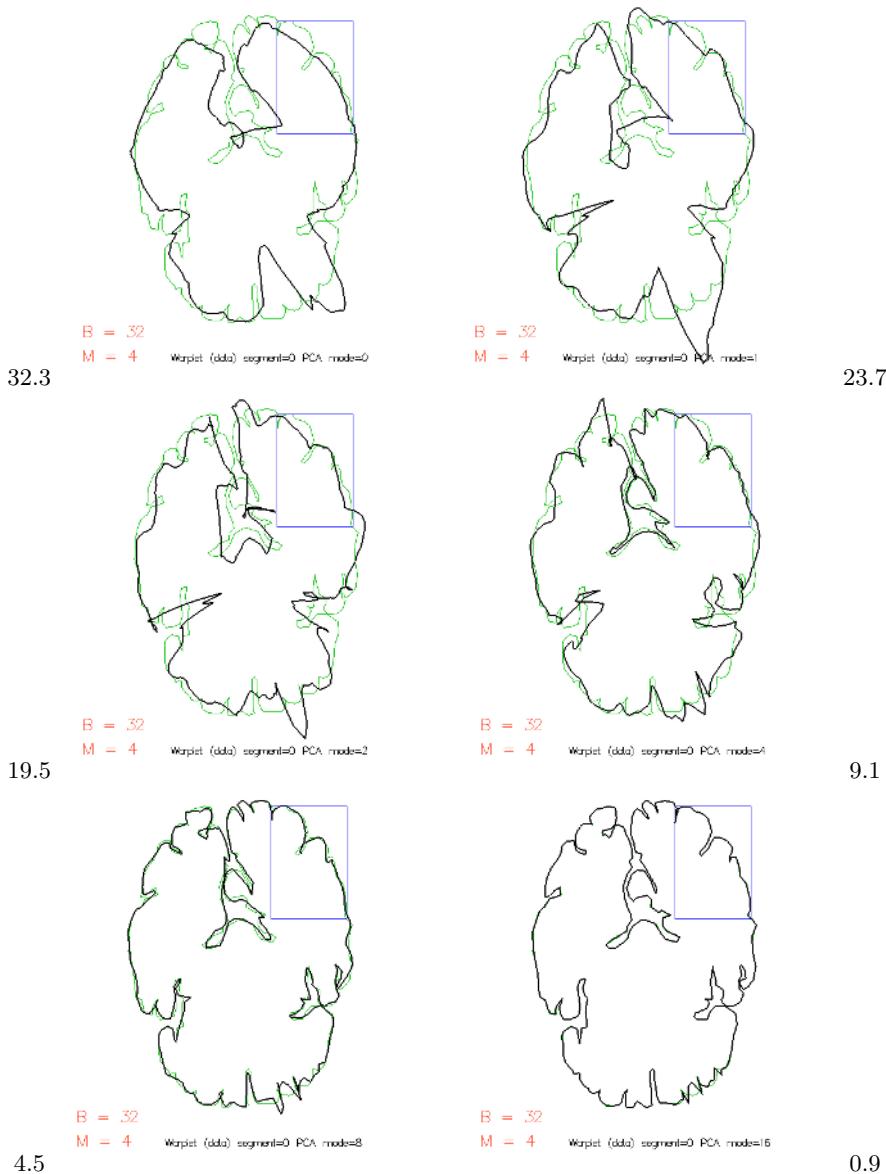


**Fig. 5. Comparison of warplet PCA model against unregistered model.** Log plot of eigenvalues against number of principal mode. The PCA on unregistered segments (red) gives an upper-bound for this curve (warplet PCA in green)



**Fig. 6. Mean and eigen modes of aligned segments.** Shown for entire brain contour (shown in middle row, figure 4). Each row,  $0 \leq r \leq 4$ , shows the principal non-zero modes,  $d$ , as  $M_k + (r - 2)\sqrt{\lambda_{kd}}\Phi_{kd}$

of which exhibit the characteristics we claim to be able to succinctly capture. We suggest that the model could be usefully applied to study brain morphology [9, 10]. For example, Mangin et al. [11] note the fundamental problem of establishing sulcal-based homologies between subject brains so that the gyri can be compared. Perhaps, a warplet description of the folding could provide a



**Fig. 7. Warplet reconstructions.** Shown for increasing numbers of PCA modes on brain contour  $D = \{0, 1, 2, 4, 8, 16\}$ . Point RMS errors given beside each figure

way to generate these alphabets or codebooks for sulcal-based homologies between subjects [11]. Further investigations however is needed into: extension of warplets to model surfaces by the use of spherical polar coordinates and overlapping, windowed patches of mesh points; the choice an appropriate prototypical patch, perhaps using an iterative learning approach based on residual discripan-

cies in shape space [7]; the variation of the contour lengths/patch sizes, possibly using a pyramid decomposition; and the constraints on the warps, e.g. to be diffeomorphic [12].

## References

1. T. Cootes and C. Taylor. Active Shape Models. In *Proceedings of British Machine Vision Conference*, pages 265–275, 1992.
2. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.
3. R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. 3D Statistical Shape Models Using Direct Optimisation of Description Length. In *Proceedings of ECCV 2002*, volume 3, pages 3–20, 2002.
4. C. Davatzikos, X. Tao, and D. Shen. Hierarchical Active Shape Models, Using the Wavelet Transform. *IEEE Transactions on Medical Imaging*, 22(3):414–423, 2003.
5. B. C. Vermuri and A. Radisavljevic. Multiresolution Stochastic Hybrid Shape Models with Fractal Priors. *ACM Transaction on Graphics*, 13(2):177–207, 1994.
6. J. A. Schanabel and S. R. Arridge. Active Shape Focusing. *Image and Vision Computing*, 17(5-6):419–429, 1999.
7. I. Dryden. General shape and registration analysis. In O. E. Barndorff-Nielsen, W. S. Kendall, and M. N. M. van Lieshout, editors, *Stochastic Geometry*. Chapman and Hall, 1999.
8. I. Corouge, S. Gouttard, and G. Gerig. A Statistical Shape Model of Individual Fiber Tracts Extracted from Diffusion Tensor MRI. In *Proceedings of MICCAI 2004*, volume 3, pages 671–679, 2004.
9. P. M. Thompson, R. P. Woods, M. S. Mega, and A. W. Toga. Mathematical/Computational Challenges in Creating Deformable and Probabilistic Atlases of the Human Brain. *Human Brain Mapping*, 9:81–92, 2000.
10. L. R. Monteiro. Multivariate Regression Models and Geometric Morphometrics: The Search for Causal Factors in the Analysis of Shape. *Systems Biology*, 4(1):192–199, 1999.
11. J.-F Mangin, D. Riviere, A. Cachia, E. Duchesnay, Y. Cointepas, D. Papadopoulos-Orfanos, P. Scifo, and T. Ochiai. A framework to study the cortical folding-patterns. *Neuroimage*, 23:129–138, 2004.
12. T. Cootes, C. J. Twining, and C. J. Taylor. Diffeomorphic Statistical Shape Models. In *Proceedings of BMVC 2004*, volume 1, pages 447–456, 2004.

# Learning Based System for Detection and Tracking of Vehicles

Hakan Ardo

Center for Mathematical Sciences, Lund University, Sweden,  
ardo@maths.lth.se

**Abstract.** In this paper we study how learning can be used in several aspects of car detection and tracking. The overall goal is to develop a system that learns its surrounding and subsequently does a good job in detecting and tracking all cars (and later pedestrians and bicycles) in an intersection. Such data can then be analyzed in order to determine how safe an intersection is. The system is designed to, with minimal supervision, learn the location of the roads, the geometry needed for rectification, the size of the vehicles and the tracks used to pass the intersection. Several steps in the tracking process are described. The system is verified with experimental data, with promising results.

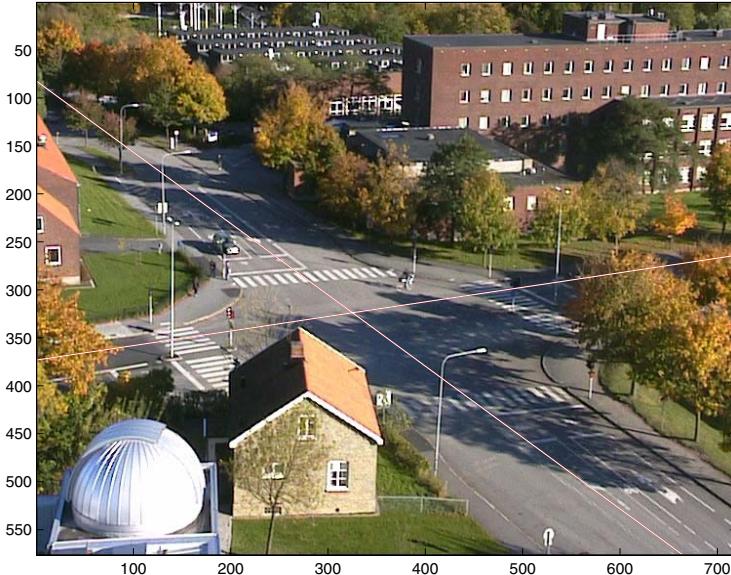
## 1 Introduction

There is an increased need for automatic analysis of car and pedestrian traffic. Methods for detecting and tracking cars in images and image sequences has been developed for quite some time now. Most of the existing methods, do however concentrate on car detection using appearance models [11], [4].

This project is initiated by the needs of governmental organizations such as 'Väg och trafik Institutet' but also department of public works in cities. One of the prime motives here is the possibilities to assess traffic safety. Often safety is neglected when constructing new roads and intersections. If safety is considered there are very few tools to actually measure and monitor traffic safety. One possibility is to study the number of accidents reported to the police from a certain intersection. Recent research [6] has, however, shown that it is possible to predict how many such accidents there is by manually observing certain events during a shorter time interval, e.g. 2-3 days. These traffic studies are however very time-consuming and requires that trained personnel study the traffic.

The goal of this study is to use learning to increase the robustness of a tracking system. Learning is used, to estimate models of scene background, to automatically rectify the image and to locate the lanes in the images. The problem of counting the number of cars driving through a intersection, such as the one shown in Figure 1 will be addressed as well as for each car deciding which entry and exit lane is used.

Several methods for estimating and updating the background images exist. One such method by Stauffer and Grimson[10] is based on modeling the probability distribution of each background pixel as a mixture of Gaussians using the EM algorithm. The dominant component is considered to be the background and is used for estimating foreground pixels. A more recent paper [5] extends this to include pan-tilt rotations of the camera.



**Fig. 1.** Video sequences used for the experiments, with the detected roads plotted

Methods for detecting the common object trajectories in a general scene under surveillance have been developed by for example Makris and Ellis in [7] and, Johnson and Hogg in [8]. In both articles it's mentioned that this kind of models can be used to predict the future position of the tracked objects, which could be very useful in the prediction step of a tracking algorithm. But no tracking algorithm exploiting this learnt information are presented or tested. In this paper a tracker developed that uses learnt models for both prediction and detection of cars is implemented and it's results is compared to a tracking operating on the same input data but using no prior information. Also, it is shown that the models can be used to reduce a 2D tracking problem into several 1D tracking problems. The road model presented in this paper explicitly models the right-hand-side driving of cars in the road and are thus able to detect the roads in the scene even if there are a lot of pedestrians walking across the road in strange directions. This would probably not be the case of the more general purpose algorithms [7] and [8].

Ground plane calibration have been done by Stauffer and Grimson in [3] and by Renno, Orwell and Jones in [9]. In both cases the a ground plane were estimated from the height of tracked objects, which assumes that the objects used for the calibration is high and thin, e.g. people. In this paper the area is used instead which allows lower and wider objects to be used for the calibration such as cars.

## 2 Methods

The idea is to start out with a simple tracker based on the background/foreground-segmentation described by Stauffer-Grimsson [10]. From this binary segmentation image, connected segments are extracted and considered objects and then segments over-

lapping between adjacent frames are connected into tracks. If there are several overlapping segments only the largest segments are considered. The resulting tracks are not very good. Typically a relatively high proportion of them are broken into several tracks and some are erroneously merged. However, the results are good enough to allow the system to automatically locate the roads (Section 2.1), rectify the image (Section 2.2) and estimate the mean size of the objects.

In Section 2.2 a novel system for automatic rectification is presented. The idea here is that all cars are approximately the same size, which can be estimated. Then the probability of finding an object of this size at a certain position in the rectified images is estimated in Section 2.3. This improves the tracking as a lot of noise can be removed by assuming a minimum size of the tracked objects. In Section 2.4 the tracks are organized into classes depending on how they drive through the intersection. For each of those classes a bspline-curve is fitted to the tracks describing the mean motion. The final tracker then detects objects of the estimated size following one of those splines.

The calibration steps of finding the roads, the rectification and the splines are typically performed off-line. Once that is done the main processing time of the 1D tracker is done by the background/foreground segmentation followed by the viterbi-optimization described in Section 2.4. For both those algorithms there exists fast implementations that should allow real time performance.

## 2.1 Automatic Road Detection

The first step in the analyze is to determine where in the image the roads are located. The idea here is to model a lane as a straight line of a certain width containing objects traveling in one direction along the line. Then a road can be modelled as two parallel lanes with opposite traveling directions. The problem is to find a probability distribution function,  $P_{road}(o|\theta)$ , that, given a set of parameters describing the road,  $\theta$ , generates the probability that an object,  $o$ , is located on the road. Here an object is described by its position  $(x, y)$  and it's traveling direction  $(\delta_x, \delta_y)$ , a unit length vector. That is  $o = (x, y, \delta_x, \delta_y)$ .

*Lane Modelling.* The model originates from a straight line  $ax+by+c=0$ , representing the center of the lane. Assume that the distance between a car in the lane and this line is Gaussian distributed with a mean value zero and a variance depending on the width of the lane. The distance from an object at  $(x, y)$  to the line is

$$t = \frac{ax + by + c}{\sqrt{a^2 + b^2}}, \quad (1)$$

which is inserted into the one-dimensional Gaussian probability distribution function. The result is divided by  $L$ , the length of the line, to make sure the the pdf still integrates to one. The resulting distribution is

$$P(x, y|a, b, c, \sigma, L) = \frac{1}{L\sigma\sqrt{2\pi}} e^{-\frac{(ax+by+c)^2}{(a^2+b^2)2\sigma^2}}. \quad (2)$$

The parameters  $a, b, c$  can be rescaled without changing the line they represent. This degree of freedom is used to represent the variance,  $\sigma$ . A particularly simple form is achieved by setting  $\sigma = 1/\sqrt{2(a^2 + b^2)}$ , then

$$P(x, y|a, b, c, L) = \frac{\sqrt{a^2 + b^2}}{L\sqrt{\pi}} e^{-(ax+by+c)^2}. \quad (3)$$

The traveling direction of each object is defined as a unit length vector  $(\delta_x, \delta_y)$  indicating in which direction it is moving. For a given line there are two possible traveling directions, which are found by rotating the normal  $\pm \frac{\pi}{2}$ . By defining the traveling direction as the normal rotated  $\frac{\pi}{2}$  the sign of the normal will decide which the model represents. Assuming that the traveling direction of the cars also is Gaussian distributed with this mean direction, some variance  $\sigma_x, \sigma_y$  and that it is independent of  $(x, y)$  gives

$$P_{lane}(o|\theta) = \frac{\sqrt{a^2 + b^2}}{2L\pi^{3/2}\sigma_x\sigma_y} e^{-(ax+by+c)^2} \cdot e^{-\frac{\left(\delta_x + \frac{b}{\sqrt{a^2+b^2}}\right)^2}{2\sigma_x^2} - \frac{\left(\delta_y - \frac{a}{\sqrt{a^2+b^2}}\right)^2}{2\sigma_y^2}}, \quad (4)$$

where

$$o = (x, y, \delta_x, \delta_y, L) \quad (5)$$

$$\theta = (a, b, c, \sigma_x, \sigma_y) \quad (6)$$

*Road Modelling.* A road consists of two lanes of opposite traveling directions. Let the line defined by  $ax + by + c = 0$  represent the center of the road, and use the sign function to indicate which side of the line a point is located. If right hand side driving is assumed the normal rotated  $\frac{\pi}{2}$  is the traveling direction on the positive side and  $-\frac{\pi}{2}$  on the negative side, which gives

$$P_{road}(o|\theta) = \frac{\sqrt{a^2 + b^2}}{2L\pi^{3/2}\sigma_x\sigma_y} e^{-(ax+by+c)^2} \cdot e^{-\frac{\left(\delta_x - \frac{\text{sign}(ax+by+c)b}{\sqrt{a^2+b^2}}\right)^2}{2\sigma_x^2} - \frac{\left(\delta_y + \frac{\text{sign}(ax+by+c)a}{\sqrt{a^2+b^2}}\right)^2}{2\sigma_y^2}}. \quad (7)$$

*Parameter Estimation.* The task of finding the two roads of the image is now reduced to that of estimating the parameters  $\theta_1$  and  $\theta_2$  as defined by equation 6 and 5. For this the EM-algorithm [1] is used to maximize (7) for the data produced by the initial tracker. It requires an estimate of  $\theta$  from a set of measured data-points. For that an algorithm similar to RANSAC [2] is used, where a set of candidate parameters is generated by iterating the steps below. Then the candidate with the highest probability is chosen.

- (i) Choose two points  $p_1$  and  $p_2$  in the dataset at random.
- (ii) Let  $(a, b, c)$  be the line through  $p_1$  and  $p_2$ .
- (iii) Rotate all data-points to make the above line parallel to the x-axis.
- (iv) Estimate the standard deviation,  $\sigma$  along the x-axis.
- (v) Encode  $\sigma$  into  $a, b$  and  $c$  by scaling them by  $\frac{1}{\sigma\sqrt{2(a^2+b^2)}}$ .
- (vi) Estimate  $\sigma_x$  and  $\sigma_y$

Setting  $L$  to the length of the line visible in the image will give short roads a higher probability. To prevent this  $L$  is chosen to be a constant equal to the diagonal of the image.

The EM-algorithm is initialized with an initial guess of two roads one parallel to the x-axis and one parallel to the y-axis. The algorithm is then executed for a few iterations to get a better starting point before an uniform background distribution is introduced. It absorbs much of noise outside the lanes.

## 2.2 Automatic Rectification

Once the roads have been located all tracks not located within the roads can be removed the rest can be used to rectify the image. The assumption here is that most objects traveling on the roads are of the same size and that the roads can be approximated with a plane. This will be used to find a rectification of the image that will give all objects the same area in the image. The estimated road models from the previous sections are used to filter out tracks not originating from cars.

To rectify the image a  $3 \times 3$  matrix,  $H$ , has to be found, that transforms the coordinates of the original image  $(x, y)$  into the coordinates of a rectified image  $(\hat{x}, \hat{y})$  according to

$$\lambda \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (8)$$

The area of objects in the original image,  $A$ , will in the rectified image be scaled by the functional determinant into

$$\hat{A} = \left| \begin{array}{cc} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{array} \right| A = \frac{\det H}{(h_{31}x + h_{32}y + h_{33})^3} A. \quad (9)$$

For each position of each tracked object there is one data-point consisting of a position  $x_k, y_k$  and a corresponding area  $A_k$ . All areas in the rectified image,  $\hat{A}$ , are assumed to be 1 (equal), and  $\det H$  can be ignored as it is only a constant, rescaling all areas equally. This turns (9) into the set of equations

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & & \\ x_n & y_n & 1 \end{pmatrix} \begin{pmatrix} h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} A_1^{1/3} \\ A_2^{1/3} \\ \dots \\ A_n^{1/3} \end{pmatrix} \quad (10)$$

with three unknowns,  $h_{31}, h_{32}$  and  $h_{33}$ .

Solving this set of equations gives the last row in  $H$ . If the camera is calibrated the rectification is a pure rotation and the last two rows can be chosen such that  $H$  becomes a rotation matrix. This still leaves one degree of freedom which corresponds to an in-plane rotation of the rectified image and can thus be chosen arbitrarily.

## 2.3 Car Indicator

From the rectification algorithm it is also straight forward to extract the area of the cars. Using this area estimate it is then possible to construct a car indicator. The probability

of finding a car at position  $L_c = (x_c, y_c)$  in frame  $f$ ,  $P_c(L_c, f)$ , can be calculated from the binary background/foreground segmentation,  $S_f(x, y)$ . This image is one for every pixel considered foreground and zero for every pixel considered background. Assuming that the pixel has the correct classification with a probability of  $p_{fg}$  and that all cars are square with a radius of  $r$ ,  $P_c$  can be found as

$$P_c(L_c, f) = \prod_{\substack{|x - x_c| < r \\ |y - y_c| < r \\ S_f(x, y) = 1}} p_{fg} \prod_{\substack{|x - x_c| < r \\ |y - y_c| < r \\ S_f(x, y) = 0}} (1 - p_{fg}) \quad (11)$$

## 2.4 1D Tracking

The final step is to learn the typical tracks a car can use to legally cross the intersection and then use these tracks to extract full car tracks from the  $P_c(L_c, f)$  images. This is done by fitting a bspline-curve,

$$(x, y) = B_k(t), \quad 0 \leq t \leq 1, \quad k = 1..N. \quad (12)$$

to each of the  $N$  typical tracks. In a medium sized intersection  $N = 16$  as there is 4 roads entering the intersection and 4 leaving it (12 if u-turns are not considered).

For each  $B_k$   $P_c$  is restricted to  $M$  points along the curve. This generates  $k$  new images,  $I_k$ , with one column per frame  $f$ ,

$$I_k(f, t) = P_c(B_k(\frac{t}{M}), f), \quad (13)$$

where  $k = 1..N$  and  $t = 0..M$ , cf. Figure 5.

Every car driving, legally, through the intersection will drive along one of the bspline-curves and always in the same direction. The parameterization,  $t$ , can thus be chosen so that a car always enters at  $t = 0$  and exits at  $t = 1$ . Considering each of the  $I_k$  images separately and assuming that cars are driving continuously, along one of the bspline curves, a car consists of a sequence of points

$$C = \{(f_i, t_i)\}_{i=1}^L, \quad (14)$$

where  $f_i \leq f_{i+1} \leq f_i + 1$  and  $t_i \leq t_{i+1} \leq t_i + 1$ .

If the motion is also assumed to be continuous this means that one of

$$\begin{cases} f_i = f_{i-1} + 1 \\ t_i = t_{i-1} \end{cases}, \begin{cases} f_i = f_{i-1} \\ t_i = t_{i-1} + 1 \end{cases} \text{ or } \begin{cases} f_i = f_{i-1} + 1 \\ t_i = t_{i-1} + 1 \end{cases} \quad (15)$$

will always be true.

The different curves,  $B_k$ , overlap. Thus a car belonging to class  $k$  has to drive along the entire curve  $B_k$ . That is  $t_1 = 0$  and  $t_L = M$ . For such a sequence, the probability of it being a car is

$$p(C) = \prod_{i=1}^L I_k(f_i, t_i). \quad (16)$$

Finally, assume that all cars driving through the intersection generate local maxima to (16) above some small threshold,  $P_{nc}$ . Then for each potential entry point (local maxima to  $(f, 0)$ ) and exit point (local maxima to  $(f, M)$ ) a local maxima to (16) can be found by viterbi-optimization. All those sequences form a finite set of candidate cars  $S_{cc} = \{C_j\}$ ,  $j = 1..N_{cc}$ , and the set of cars driving through the intersection,  $S_c$ , can be found as the non-overlapping subset of  $S_{cc}$  maximizing  $\sum_{C_j \in S_c} p(C_j)$ , where  $S_c$  non-overlapping means that

$$\begin{cases} C_i, C_j \in S_c \\ (f, t) \in C_i \end{cases} \Rightarrow (f, t) \notin C_j. \quad (17)$$

As  $I_k$  is a sampled, discrete image two close local maxima might be merged to one. So accepting overlapping points in  $S_c$  for a few  $t$ -values, makes the system more robust.

### 3 Experimental Results

The initial tracker simply combining overlapping connected segments were tested on a 5 min sequence (see Figure 1) containing 62 vehicles (1 tractor, 1 bus and 60 cars) and 761 tracks were found. This includes vehicles, bicycles, pedestrians and pure noise. Many tracks are slitted into several. No objects are entirely missed though.

The road detection algorithm were then successfully applied, as shown in Figure 1, followed by the rectification algorithm resulting in the left image in Figure 2. The road is clearly tilted to the right. This is due to the simple linear optimization method used and the fact that there is quiet a lot of outliers and noise. By manually selecting a set of good tracks the right image is achieved, which is the wanted result. This means that by using non-linear optimization or better measurements the algorithm will probably perform just fine.



**Fig. 2.** Output from the automatic rectification. To the left the algorithm worked on all tracks available. To the right some good tracks were manually chosen



**Fig. 3.** The two roads in the image located approximately. Each road is represented by two lanes of opposite driving direction. The solid line shows the center of the road separating the two lanes and the two dashed lines shows the border of the road



**Fig. 4.** All tracks used to pass the intersection clustered into 12 clusters based on their entry and exit lane. Each cluster is represented with a spline (shown in the figure) fitted to all the tracks in the cluster. The red splines were found by the proposed algorithm and the two yellow ones had to be added by hand

The road detection algorithm have been tested on tracks originated from both the original image and from the rectified, with similar results shown in Figure 1. The hori-



**Fig. 5.** The probability that there is a car located at some position (y-axis) along one of the splines shown in Figure 4 over time (x-axis). The straight lines are cars passing through the intersection without stopping. The lines with a long horizontal part in the middle are cars that stop and wait for the red light. This image is denoted  $I_k(f, t)$  where  $t$  is the position along spline  $k$  in frame  $f$

zontal road is detected above the road centrum line in the image, which is correct as the height of the cars will place the centrum point of the tracked segments slightly above their position on the road. The result of the later is shown in Figure 3.

The result of detecting the typical tracks used to traverse the crossing is shown in Figure 4. 10 of all 12 tracks were found. The two missed are marked yellow in the Figure. The problem with the last two is that the initial tracker produced no tracks belonging to these classes, which is because there is very few cars using them.

Finally the  $I_k$  images were produced, see Figure 5, and the 1D tracker were executed. In total 66 (of totally 62) tracks were found including 3 bikes, with significantly lower  $p(C)$ , and 1 ghost car originating from a combination of cars, which could be removed by removing overlapping cars from the final set of cars from all classes. Also the bus and the tractor were detected as 2 cars each, as they are significantly larger than a car. Also, two cars were missed, one occluded by the building. All other cars were detected correctly.

## 4 Conclusions

In this paper we have introduced methods for automatic rectification, road, lane and car detection based on learning. The results on real data are promising but additional work is needed.

Experiments have indicated that the rectification algorithm seems robust to errors in the estimation of the focal-length. This means that rectifying an uncalibrated camera should also be possible by for example assuming that the aspect ratio of a car is 1x2 and let the traveling direction indicate the longer side. Also, an optimization method less sensitive for outliers are needed, or cleaner input data which could be achieved by using a better initial tracker, or by implementing some feedback loop passing the data produced by the final tracker back to the rectification to allow it to tune the rectification parameters. It should also be quite possible to remove the assumption that all cars have the same size, and instead assume that any single object traveling through the intersection does not change its size. Then all the tracks could be clustered not only on their entry and exit lanes, but also on their area and different 1D-trackers could be used for vehicles of different sizes.

One single car will produce data in several of the  $I_k$  images and thus optimizing over all of them at once instead of one by one should yield better results. In that case a more advanced model of how cars are allowed to move among the sample points is required, such as for example a Markov model. In that case a uniformly distributed set of sample points might be enough and there will be no need to estimate the bspline curves as the Markov model will represent the relations between the sample points.

## References

1. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (Series B):1–38, 1977.
2. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications-of-the-ACM*, 24(6):381–95, 1981.
3. W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 22. IEEE Computer Society, 1998.
4. Michael Haag and Hans-Hellmut Nagel. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision*, 35(3):295–319, 1999.
5. E. Hayman and J-O. Eklundh. Background subtraction for a mobile observer. In *Proc. 9th Int. Conf. on Computer Vision*, Nice, France, 2003.
6. C. Hydén. *The development of a method for traffic safety evaluation: The Swedish traffic conflicts technique*. PhD thesis, Institutionen för trafikteknik, LTH, Lund, 1987.
7. Neil Johnson and David Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC '95: Proceedings of the 6th British conference on Machine vision (Vol. 2)*, pages 583–592. BMVA Press, 1995.
8. Dimitrios Makris and Tim Ellis. Path detection in video surveillance. *Image Vision Comput.*, 20(12):895–903, 2002.
9. J. R. Renno, James Orwell, and Graeme A. Jones. Learning surveillance tracking models for the self-calibrated ground plane. In *BMVC*, 2002.
10. Chris Stauffer. Adaptive background mixture models for real-time tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 246–252, 1999.
11. Tao Zhao and Ram Nevatia. Car detection in low resolution aerial image. In *Proc. ICCV 2001*, pages 710–717, 2001.

# Texture Analysis for Stroke Classification in Infrared Reflectogramms

Martin Lettner and Robert Sablatnig

Vienna University of Technology,  
Institute of Computer Aided Automation,  
Pattern Recognition and Image Processing Group,  
Favoritenstrasse 9/183/2, 1040 Vienna, Austria  
`{lettner, sab}@rip.tuwien.ac.at`  
<http://www.rip.tuwien.ac.at>

**Abstract.** The recognition of painted strokes is an important step in analyzing underdrawings in infrared reflectogramms. But even for art experts, it is difficult to recognize all drawing tools and materials used for the creation of the strokes. Thus the use of computer-aided imaging technologies brings a new and objective analysis and assists the art experts. This work proposes a method to recognize strokes drawn by different drawing tools and materials. The method uses texture analysis algorithms performing along the drawing trace to distinguish between different types of strokes. The benefit of this method is the increased content of textural information within the stroke and simultaneously in the border region. We tested our algorithms on a set of six different types of strokes: 3 classes of fluid and 3 classes of dry drawing materials.

## 1 Introduction

Infrared reflectogramms are a popular tool for the investigation of underdrawings from medieval painted works of art. Underdrawings constitute the basic concept of an artist when he starts his work of art. Normally they are hidden by paint layers of the finished work and thus unseen in the visible range of the electromagnetic spectrum. Infrared reflectography (IRR) allows a look through the paint layers and thus a visualization of the underdrawing. The wavelength of IRR lies in the range from approx. 1000 to 2500nm where the longer wavelength facilitates the penetration of the paint layers. The generated image is called infrared (IR) reflectogram [3]. Conservators and art historians are interested in the development of underdrawings, their relation to other drawings and differences between underdrawings and the covering painting. Further more painting instructions and the identification of the drawing tool and material used for the creation of the underdrawing are of particular interest. But the recognition of the drawing tool and material of painted strokes in IR images is not always clear and unambiguous. The limited resolution of the acquisition system, the use of different tools in a painting and disturbing paint layers make a recognition with the naked eye difficult. Thus the use of computer-aided systems can assist art

experts in doing their work comparable to the usage of computers in medical applications which are nowadays inconceivable without computers.

Painted Strokes can be drawn either in dry or fluid drawing material. Chalk and graphite are examples for dry materials and paint or ink applied by pen or brush are examples for fluid painting materials. The appearance of the boundary characteristics, the texture, the stroke endings or the color variety can be used for the visual recognition.

In this work we are going to develop an algorithm which allows the identification of the drawing material used for the creation of painted strokes. Several work in this direction has been done before. A segmentation and classification of underdrawing strokes by the help of snakes is reported in [6]. The analysis of the texture of strokes is shown in [4] and [8]. Wirotius et al. [14] showed a differentiation in gray level distributions for writer identification.

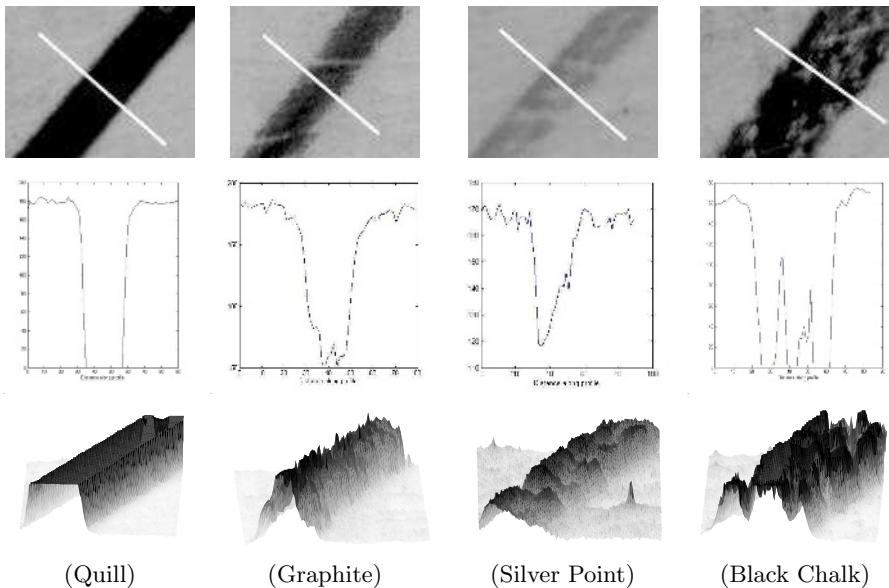
Our approach differs with respect to [4] and [8] in that the calculation of the textural features is aligned i.e. the window in which the features are extracted moves along the stroke trace, parallel to the stroke boundary. Through this innovation we have several advantages: we can use bigger analysis windows in order to have more texture information, we have more texture information of the border region of the strokes and the method takes the directional nature of the texture formation process into account.

The organization of this paper is as follows. The next section shows the data material used in our work. Section 3 covers the algorithm. In Section 4 experiments and results are given followed by conclusions and an outlook in Section 5.

## 2 Data

The texture of painted strokes depends primarily on the painting tool and material used. Also the underground affects the appearance of the strokes. But for medieval panel paintings the underground is prepared to be as plain as possible. Thus the effect of the painting underground will be not investigated in this work as well as we have no information about this fact in IR reflectogramms.

The strokes considered in the present study are applied on test panels prepared by a restorer. Bomford specified typical drawing media used for the creation of underdrawings in medieval painted work of art [1]. These drawing media will be examined and considered in this work: graphite, black chalk and silver point are the representatives for the dry strokes and ink applied by brush, quill or reed pen are the considered fluid strokes. Figure 1 gives examples for dry and fluid drawing materials. The first row shows a sample window from a scanned image in the size of  $100 \times 80$  pixels. Pixel-value cross-sections along the white line segments from this image can be seen in the second row. The image profile varies clearly between the strokes and the main differences lie within the border regions of the profiles and thus in the border region of the strokes. Hence important information to distinguish between strokes lies within the border region of the strokes and thus even these regions have to be considered



**Fig. 1.** Strokes considered: normal view, profile and 3D view

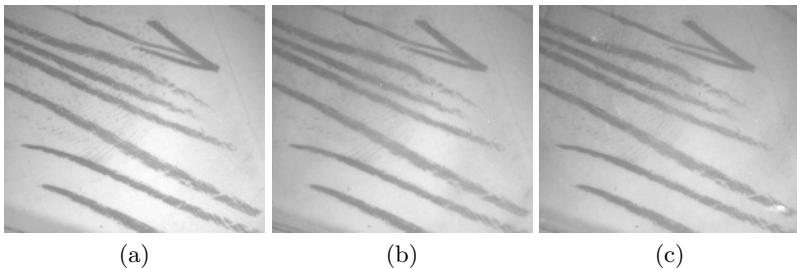


**Fig. 2.** The reed pen stroke shows discontinuities along the stroke surface (texture)

in the texture analysis. The third row shows a 3D view of the stroke surface with the pixel value on the  $z$ -axis. The surface between dry materials varies clearly but the surface from fluid drawing materials is nearby constant. Differences can be seen in the distribution of the texture over the whole stroke. Quill strokes have a very homogeneous black surface. The texture from brush strokes is very similar except to some brighter areas. The surface from the reed pen shows some discontinuities with some brighter areas in the medial part which has less drawing material than the border region. This incident can be seen in Figure 2.

## 2.1 Scanned Images

For our first tests we digitized the panels using a flat-bed scanner with a relative resolution of 1200dpi. Examples of the scanned images can be seen in Figure 1.



**Fig. 3.** IR images without (a) and with the covering color layers Alizarin(b) and Ultramarine blue(c)

## 2.2 IR Images

To test our method on real underdrawings we covered the test panels with paint layers in order to simulate real underdrawing strokes. Our acquisition system consists of a Vidicon tube camera (Hamamatsu C1000) with a spectral sensitivity between 900nm and 2000nm which was attached to a Matrox frame grabber. The size of the digitized images is  $629 \times 548$  pixel with an relative resolution of approx. 700 dpi.

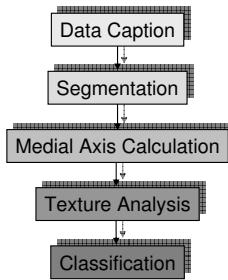
The visibility of underdrawing strokes in IR reflectograms depends on the transmission rate  $\tau$  and the thickness of the color pigments and the contrast between underdrawing strokes and the background [9]. For our work we used two color pigments with a high transmission rate. The Schmincke Mussini® colors Alizarin (red) and Ultramarine blue were applied on thin glass plates which covered our test panels in order to simulate underdrawing strokes. Figure 3 shows IR images where the test panels were covered with the paint layers. It can be seen that the strokes and even their texture can be realized in the covered IR images (b) and (c). Only some reflection points and blurred parts can be seen. Remember that the strokes in (b) and (c) cannot be realized in the visible range through the covering paint layer.

## 3 Algorithm

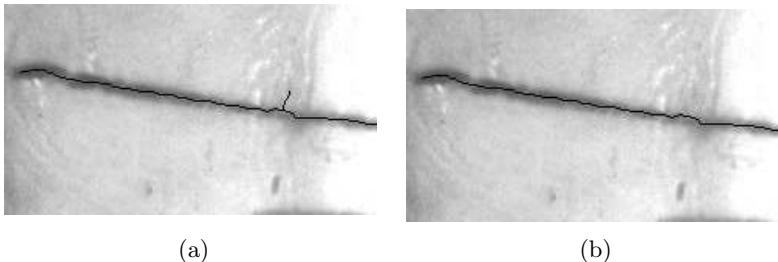
Figure 4 gives an overview of our algorithm. After the acquisition of the test panels and the segmentation of the individual strokes we calculate the medial axis in order to afford the calculation of the textural features along the stroke trace in painting direction. The textural features calculated are the input for the classifier which determines the painting tool and material used for the creation of the stroke.

### 3.1 Segmentation

For our purpose of testing the algorithm on test panels it is sufficient to segment the strokes from the plain background. Through the similarity to document im-



**Fig. 4.** The overall algorithm for the recognition of painted strokes

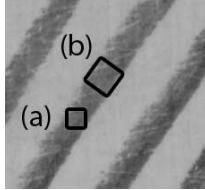


**Fig. 5.** (a) Input image, (b) shows a detail from (a) after the segmentation and calculation of the medial axis

age analysis we use the top hat transformation to plain the background followed by a global threshold produced by Otsu's method to segment the strokes from the background. The method showed best result in [7]. Morphological operations remove artifacts from the background and inside the strokes. After segmenting the strokes from the background the intrinsic segmentation of the strokes is done. We calculate the medial axis of the strokes with the thinning algorithm from [15]. To enable the calculation of the textural features a stroke can have only two endpoints of his medial axis. Thus small artifacts from the skeleton are removed by pruning and crossing strokes are separated into individual segments. Figure 5 shows an example of the medial axis with removed artifacts from an IR image.

### 3.2 Texture Analysis

The primary task in identifying the painted strokes is the extraction of the textural features. The benefit of our method is the direction controlled analysis. Standard texture analysis algorithms perform parallel to the image boundaries and calculate features for every pixel. For the stroke application it is optimal to scan the textural features along the medial axis of the stroke trace. This condition can be seen in Figure 6 where the rotated sampling window (b) contains more



**Fig. 6.** Sample windows to calculate textural features. (a) standard method, (b) proposed method

texture information than the normal window (a). Through this innovation we have several advantages:

- Bigger sample windows can be adopted to gain the textural features for providing more texture information
- The windows include more border information of the strokes which is fundamental to distinguish between them (see Section 2)
- The performance is better because textural features are calculated only for the segmented part

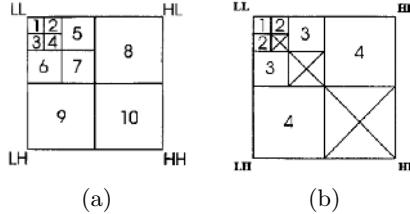
To gain the textural features we applied two different texture analysis methods. A similar application area is the texture analysis of clouds [2]. So the first method used in our work is the Gray Level Co-occurrence Matrix (GLCM) [5] which showed good results in this work and the second method is the discrete wavelet transformation DWT [10] which also outperformed other methods in several comparative studies [13, 12].

**Gray Level Co-occurrence Matrix.** The GLCM is a very popular tool for texture analysis. It was presented in 1973 by Haralick et al. [5]. The  $N \times N$  GLCM describes the spatial alignment and the spatial dependency of the different gray levels, whereas  $N$  is the number of gray levels in the original image. The co-occurrence matrix  $P_{\phi,d}(i,j)$  is defined as follows. The entry  $(i,j)$  of  $P_{\phi,d}$  is the number of occurrences of the pair of gray levels  $i$  and  $j$  at inter-pixel distance  $d$  and the direction angle  $\phi$ . The considered direction angles are  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ . For a chosen distance  $d = 1$  we computed the energy, inertia, entropy and the homogeneity of the mean of these four directions to get a four dimensional feature vector.

**Discrete Wavelet Transformation.** The discrete wavelet transformation (DWT) [10] decomposes an original signal  $f(x)$  with a family of basis functions  $\psi_{m,n}(x)$ , which are dilations and translations of a single prototype wavelet function known as the mother wavelet  $\psi(x)$ :

$$f(x) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} c_{m,n} \psi_{m,n}(x) . \quad (1)$$

$c_{m,n}$  constitutes the DWT coefficients where  $m$  and  $n$  are integers and referred to as the dilation and translation parameters. An efficient way to implement this



**Fig. 7.** (a) 10 channels of a three level wavelet decomposition of an image. (b) Grouping of wavelet channels to form 4 bands to calculate the features [11]

scheme using filters was developed by Mallat [10]. The 2D DWT is computed by a pyramid transform scheme using filter banks. The filter banks are composed of a low pass and a high pass filter and each filter bank is then sampled down at a half rate of the previous frequency. The input image is convolved by a high pass filter and a low pass filter in horizontal direction (rows) followed by another convolution with a high and a low pass filter in vertical direction (columns). Thus the original image is transformed into four sub images after each decomposition step. See [11] for details. A three level decomposition results in 10 sub images, see Figure 7(a) whereas the approximation image is the input image for the next level.

We use a Daubechies *db10* motherwavelet for our analysis. The energy of the coefficient magnitudes  $c_{m,n}$  is calculated to build a four dimensional feature vector: the HL and LH sub images from each channel are combined and the HH sub images are not considered because they tend to contain the majority of noise [11], see Figure 7(b).

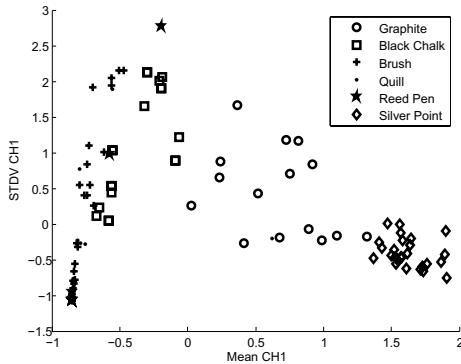
## 4 Experiments and Results

After segmenting the images and the calculation of the medial axis we have 122 scanned strokes and 258 strokes acquired with the Vidicon tube camera. We have 6 classes of strokes from the scanned images and 5 classes for the IR images. Because of the nearly invisibility of the silver point stroke in the IR image they are not considered in this test set. We perform contrast-limited adaptive histogram equalization to enhance the contrast of the IR images.

The number of textural features per stroke depends on its length and the distance between placing the windows. We used a distance of 10 pixels to place the windows along the medial axis of the stroke. To limit the feature space we combined the features calculated within each window by building the mean and standard deviation. Thus we have 8 features for the GLCM method: the mean and standard deviation for the energy, inertia, entropy and homogeneity and 8 DWT features: the mean and standard deviation of the energy in the four channels. The features calculated (GLCM and DWT) were normalized and we used the *kNN* classifier to evaluate the drawing tool and material used for the creation of the strokes.

**Table 1.** Classification results: method, number of features and percentage of correct classified strokes for scanned and IR images

Method	Nof	(%) SCAN	(%) IR
DWT	8	74.6	68.6
GLCM	8	75.4	59.3
Combination	16	75.4	70.5



**Fig. 8.** The feature space from the DWT features in Channel 1

Classification results are tabulated in Table 1. Best results were obtained with  $k = 3$ . The percentage of correct classified scanned strokes constitutes 74.6% for the DWT features and 75.4% for the GLCM features. A combination of the features from both methods could not outperform this results. Expectedly the results for the IR images are a little bit worse due to the limited resolution and contrast. Hence we have 68.6% for the features from the DWT method and only 59.3% for the GLCM features. A combination of both brings 70.5%.

To illustrate the visual description in Section 2 we show the feature space of two DWT features in Figure 8. The  $x$ -axis shows the mean value of the DWT energy in the first channel and the standard deviation from the energy in the first DWT channel is shown on the  $y$ -axis. It can be seen that all strokes show compact clusters for their features except the features for the reed pen which are distributed over the whole feature space. Brush, quill and black chalk show low energy values. The quill stroke features cluster in the lower left part of the feature space. They have a very homogeneous black surface and thus low energy and standard deviation. Brush strokes are similar but they have higher standard deviations due to some artifacts in the surface. The black chalk stroke has higher energies and standard deviation due to its coarser texture. Graphite and silver point show increasing values for their energy. Reed pen strokes with its manifold texture are distributed over the whole feature space. As in agreement with the visual description of the strokes in Section 2 fluid materials have low energy values and dry materials like black chalk, graphite and silver point have higher

**Table 2.** Confusion Matrix for the Classification results, Scanned images

stroke class	classified as (%)					
	Graphite	Chalk	Brush	Quill	Reed Pen	Silver
Graphite	80.0	0.0	0.00	0.00	0.00	20.0
Chalk	10.5	89.5	0.0	0.00	0.0	0.00
Brush	0.00	16.7	50.0	33.3	0.00	0.00
Quill	4.3	0.00	34.8	60.9	0.00	0.00
Reed Pen	0.0	12.5	75.0	12.5	0.0	0.0
Silver	0.0	0.00	0.00	0.00	0.0	100.0

**Table 3.** Confusion Matrix for the Classification results, IR images

stroke class	classified as (%)				
	Graphite	Chalk	Brush	Quill	Reed Pen
Graphite	89.1	8.7	0.00	0.00	2.2
Chalk	16.3	69.7	7.0	4.7	2.3
Brush	0.00	7.7	55.8	23.1	13.5
Quill	0.00	1.9	22.7	66.0	9.5
Reed Pen	17.2	12.5	23.4	21.9	25.0

energy values. The silver point has the highest energy values which is analog to Figure 1 where the 3D view from the silver point shows a fine texture with high frequencies.

To show the classification results for the several classes Table 2 and 3 tabulates the confusion matrix for the classification results from the DWT method for the scanned and IR images. The scanned reed pen strokes are distributed over the whole feature space so that no reed pen stroke is classified correct. Best results are within the silver point, the graphite and the black chalk stroke which show compact clusters in Figure 8. Expectedly the dry drawing materials show better results than the fluid materials due to the fact that the texture from the fluid materials is very similar. The confusion matrix for the IR image in Table 3 shows similar results.

## 5 Conclusion and Outlook

In this work a rotational alignment of texture analysis of painted strokes has been discussed. The method developed was able to recognize up to 75% of painted strokes into a set of 6 predefined classes. We applied two different texture analysis methods, the discrete wavelet transformation and the Gray Level Co-occurrence Matrix. The problems in analyzing the texture of painted strokes, the narrow width and the winding painting trace of the strokes, is avoided by an algorithm which performs along the drawing trace of the strokes to calculate the textural features. So we have a maximum content of stroke texture and we take the

directional nature of the texture formation process into account. The algorithm can be adopted to the identification of different stroke types in painted work of art as well as the recognition of writing tools in handwritten documents. To improve the classification results we will add profile features to the textural features. Profile classification has been used in the work from [14]. Further more the use of a stronger classifier will be evaluated in our future work.

## Acknowledgments

This work was supported by the Austrian Science Foundation (FWF) under grant P15471-MAT.

## References

1. David Bomford, editor. *Art in the Making, Underdrawings in Renaissance Paintings*. National Gallery, London, 2002.
2. C. I. Christodoulou, S. C. Michaelides, and C. S. Pattichis. Multifeature texture analysis for the classification of clouds in satellite imagery. *IEEE Trans. Geoscience and Remote Sensing*, 41(11):2662 – 2668, Nov. 2003.
3. J.V.A. de Boer. *Infrared Reflectography.-A contribution to the examination of earlier european paintings*. PhD thesis, Univ. Amsterdam, 1970.
4. K. Franke, O. Bünnemeyer, and T. Sy. Ink Texture Analysis for Writer Identification. In *8th International Workshop on Frontiers in Handwriting Recognition*, pages 268–273, 2002.
5. R. M. Haralick, K. Shanmugan, and I. Dinstein. Textural Features for Image Classification. *IEEE Trans. System Man Cybernetics*, 3(6):610–621, November 1973.
6. P. Kammerer, G. Langs, R. Sablatnig, and E. Zolda. Stroke Segmentation in Infrared Reflectograms. In *13'th Scandinavian Conference, SCIA 2003*, pages 1138–1145, Halmstad, Sweden, June 2003. Springer Verlag.
7. G. Leedham, C. Yan, K. Takru, J.H.N.Tan, and L. Mian. Comparison of some thresholding algorithms for text background segmentation in difficult document images. In *7th International Conference Document Analysis and Recognition*, pages 859–864, Edinburgh, August 2003.
8. M. Lettner, P. Kammerer, and R. Sablatnig. Texture Analysis of Painted Strokes. In *28th Workshop of the Austrian Association for Pattern Recognition*, pages 269–276, Hagenberg, Austria, June 2004.
9. Franz Mairinger. *Strahlenuntersuchung an Kunstwerken*. Buecherei des Restaurators Band 7. E. A. Seemann, 2003.
10. S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
11. R. Porter and N. Canagarajah. Robust rotation invariant texture classification. In *International Conference Acoustics, Speech, and Signal Procesing ICASSP-97*, volume 4, pages 3157 – 3160, Munich, April 1997.

12. R. Porter and N. N. Canagarajah. Robust relation-invariant texture classification: Wavelet, gabor filter and gmrf based schemes. *IEE Proceedings Vision, Image and Signal Processing*, 144(3):180–188, June 1997.
13. T. Randen and J.H. Husoy. Filtering for Texture Classification: A Comparative Study. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.
14. M. Wirotius, A. Seropian, and N. Vincent. Writer identification from gray level distribution. In *7th International Conference Document Analysis and Recognition*, Edinburgh, August 2003.
15. T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–240, March 1984.

# Problems Related to Automatic Nipple Extraction

Christina Olsén and Fredrik Georgsson

Intelligent Computing Group,  
Department of Computing Science,  
Umeå University, SE-901 87 Umeå, Sweden  
`{colsen, fredrikg}@cs.umu.se`  
[http://www.cs.umu.se/~fredrikg/ic\\_iman.htm](http://www.cs.umu.se/~fredrikg/ic_iman.htm)

**Abstract.** Computerized analysis of mammograms can serve as a secondary opinion, improving consistency by providing a standardized approach to mammogram interpretation, and increasing detection sensitivity. However, before any computer aided mammography algorithm can be applied to the nipple, one of several important anatomical features, need to be extracted. This is challenging since the contrast near the border of the breast, and thus the nipple in mammograms, is very low. Therefore, in order to develop more robust and accurate methods, it is important to restrict the search area for automatic nipple location. We propose an automatic initialization of the search area of the nipple by combining a geometrical assumptions verified against the MIAS database regarding the location of the nipple along the breast border and a geometrical model for deciding how far into the breast region the nipple can occur. In addition, the modelling reduces the need for parameters determining the search area and thus making the method more general. We also investigate the variance between the medical experts often use as ground truth when determining performance measures for developed medical methods.

## 1 Introduction

Mammography, x-ray imaging of the breast, is currently the best method for early detection of breast cancer. During screening mammography, the radiologist may fail to detect cancer and the lack of detections may be due to the subtle nature of the radiographic findings (i.e., low conspicuousness of the lesion), poor image quality, eye fatigue, or oversight by the radiologists. Therefore it is the practise that two radiologists should analyze the mammograms to increase the sensitivity [7]. To minimize the necessity of attention by two radiologists, computerized analysis of mammograms can serve as a secondary opinion, improving consistency by providing a standardized approach to mammogram interpretation, and increasing detection sensitivity. In computerized mammography, the need to automatically detect anatomical features, such as the background (the non-breast area), pectoral muscle, fibroglandular region, adipose region and the

nipple is very high. These entire anatomical landmarks are either direct or indirect necessary during judging mammogram in the search for abnormalities in tissue, which might be breast cancer. The first segmentation procedure involves extracting the principal feature on a mammogram; the breast border (also known as the skin-air interface). This is performed by segmenting the breast and the nonbreast into distinct regions and during this segmentation procedure it is important to preserve the nipple if it is positioned in profile. As stated, the nipple is another important anatomical feature to extract and due to positioning it does not always occur on the breast border and can be depicted inside breast tissue. The extraction of such cases is a particularly challenging image analysis task.

Our aim is to develop robust automated extraction methods for extracting the anatomical features required for an automatic patient positioning assessments in mammography. It is our intention to develop a fully automatic and generic algorithm to extract the position of the nipple. For this reason, we have in this paper developed a robust extraction of the region of interest (ROI) as an initial search area for the nipple, both along the breast border and into the breast tissue. Thereafter, we have implemented a known automatic nipple localization method and applied to our search area. Since many known techniques in medical imaging often are compared to medical expert (i.e. in our case a radiologist) we found it interesting to investigate the variance between the positions of the nipple marked by different experts.

## 2 Existing Approaches

There are several approaches to segment the breast region from mammograms reported in the literature. However, only a few of them also cover the locations of the position of the nipple. One of the earliest approaches to segmentation of the breast contour was presented by Semmlow et al. [5].

Méndez et al., [2] report a fully automatic technique to detect the breast border and the nipple. The proposed algorithm finds the breast contour using a gradient based method. Furthermore, Chandrasekhar and Attikiouzel, [1] outline a simple, fast and promising method (based on 24 images) for automatically locating the nipple in mammograms. Their method search for the nipple along the entire breast border except for 30% of pixels at the top and 5% of pixels at the bottom, this for avoiding artefacts, in these area, to interfere with the automatic method and produce inaccurate result.

We have noticed two interesting issues regarding existing methods. 1) In the methods where a region of interest is chosen for locating the position of the nipple the motivation for the selected area, is to our understanding not clearly stated. 2) The evaluation of the performance of the algorithms are compared to one or several experts in mammography, however, the effect the variance between them might have on the performance measure are not explicitly considered.

### 3 The Method of Finding the Position of the Nipple

The search area for the position of the nipple along the breast border need to be restricted for two reasons; (1) due to edge effects and artefacts that may occur at the inferior portion of the breast, near the infra-mammary fold and the chest wall and (2) due to the development of a more robust and accurate method. For this last reason, it is also necessary to investigate how far into the glandular tissue the nipple may be imaged. By modelling the geometry of the breast, suitable initialisation of the search area can be done automatically, reducing the need for fixed numbers in the algorithm.

The dataset in this study consisted of 322 digitized mammograms from the MiniMammographic Image Analysis Society's (MIAS) Digital Mammography Database<sup>1</sup>. According to Suckling et al. [6], all the mammograms have been carefully selected from the United Kingdom National Breast Screening Programme to be the highest quality of exposure and patient positioning. In this method all images were low-pass filtered and reduced in resolution to  $400^2 \mu\text{m}$  per pixel.

#### 3.1 Search Region Along the Breast Border

In addition, the investigation of the geometrical approach presented in [4] based on automatically finding the pectoralis muscle and the maximum distance perpendicular to the muscle is integrated into this algorithm. The main idea is to perform the Hough transform on a gradient image. Once the line  $P$ , representing the pectoralis muscle is found a normal line  $N$ , passing through the estimated nipple  $MAM = (x_{mam}, y_{mam})^T$  is approximated. To find the point,  $MAX = (x_{max}, y_{max})^T$ , the orthogonal maximum distance between the calculated line  $P$  and a point  $MAX$  on the breast border is estimated.

Based on 305 mammograms, the mean of euclidean distance,  $e$ , between the points  $MAX$  and  $MAM$  is 17.5 pixels with a standard deviation of 23.8 pixels. Since the nipple has an extension in area, estimates of the position of the nipple could be valid even if they are not right on the given position. In the MIAS database the mean extension of the nipple is estimated to  $\ell = 25.3$  pixels [4]. Based on this investigation we can expect to find the nipple with 99% certainty in the interval defined as  $(MAX \pm 4\ell)$  along the breast border.

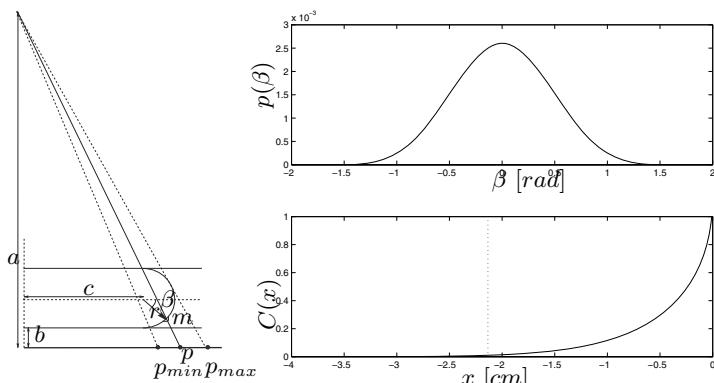
However, the approximation of the position of the nipple, based on the geometrical assumption, is dependent on two features: the angle of the pectoralis muscle and the shape of the breast border. The investigation of how perturbations in these two features effects the final approximation of the position of the nipple shows that the extracted angle of the pectoralis is not critical to estimate the position of the nipple by this geometrical approach. The critical aspect of the algorithm is the extraction of the breast border [4].

---

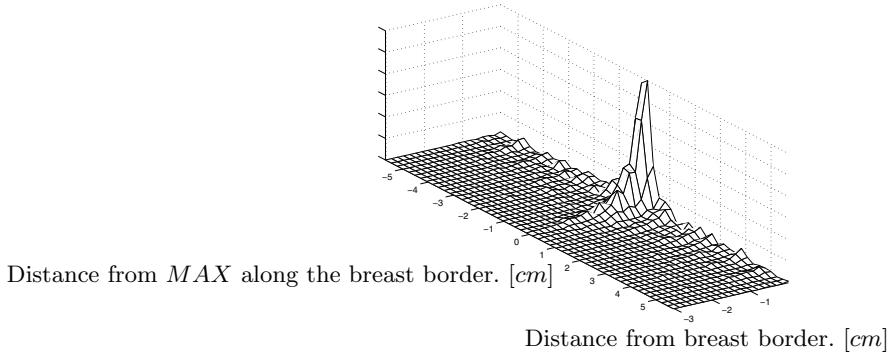
<sup>1</sup> <http://www.wiau.man.ac.uk/services/MIAS/MIASweb.html>

### 3.2 Search Region into the Breast Tissue

In order to set boundaries to the problem of how far inside the breast region the nipple can be expected to appear, some assumptions regarding the geometry of the breast were done. The main assumption is that the bulge of the breast between the compression paddles is a half-circle with radius  $r$ . Other assumptions are that the nipple is considered to be point and that the compression paddles are parallel with the image plane. With reference to Fig. 1 the geometry is approximated and under these assumption it is trivial to calculate the projected position of the center of the nipple. The values for  $a$  and  $b$  are the same for all examinations performed with a specific x-ray machine and can easily be estimated ( $a \approx 60 \text{ cm}$ ,  $b \approx 1 \text{ cm}$ ). The values of  $r$  and  $c$  is dependent on the anatomy of the particular woman under examination and we have chosen to use typical values for these parameters ( $r \approx 3 \text{ cm}$ ,  $c \approx 10 \text{ cm}$ ). The angle  $\beta$  is dependent on the skill of the radiographer and we have assumed that the values of  $\beta$  are described by a probability density function  $p(\beta)$ , see Fig. 1 Top Right. Based on the model and the values of the parameters the distribution of the distances of the projected nipple to the breast border was estimated, see Fig. 1 Bottom Right. Based on this distribution the distance between the projected nipple  $p$  in Fig. 1 and the breast border  $p_{max}$  in Fig. 1 was estimated at a confidence level of 99%. Depending on the choice of distribution for  $p(\beta)$  the following distances were obtained:  $p(\beta) = \Pi(-\frac{\pi}{2}, \frac{\pi}{2}) \rightarrow 3.73 \text{ cm}$ ,  $p(\beta) = \Lambda(-\frac{\pi}{2}, \frac{\pi}{2}) \rightarrow 3.65 \text{ cm}$  and  $p(\beta) = N(0, 0.5) \rightarrow 2.13 \text{ cm}$ . Since it is reasonable to assume that a trained radiographer manages to position the nipple close to halfway between the compression paddles most of the times the uniform distribution ( $\Pi(-\frac{\pi}{2}, \frac{\pi}{2})$ ) is a rather pessimistic assumption. Thus it is concluded that with a high degree of certainty, the nipple is not projected further into the breast than about  $3 \text{ cm}$ .



**Fig. 1.** **Left:** The geometry of nipple projection. **Right Top:** An example of a probability density function for  $\beta$  used in simulations. **Right Bottom:** The cumulative distribution of the probability of  $x = p_{max} - p$ . The dotted line marks the 99% confidence ( $x = 2.13 \text{ cm}$ )



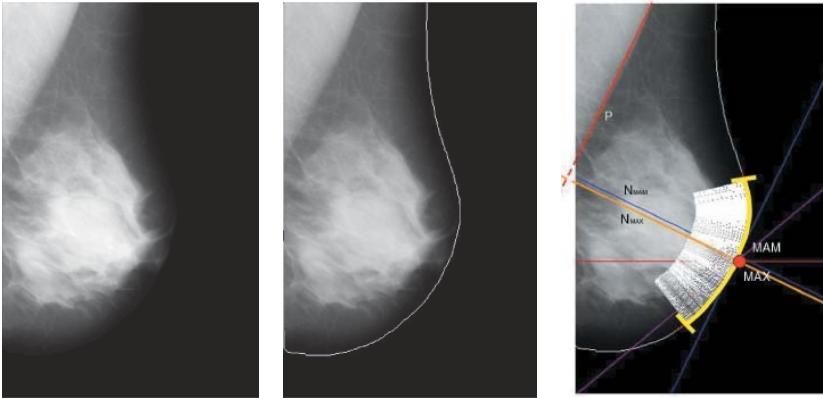
**Fig. 2.** Combined PDF of the position of the nipple. The mode of the distribution is located at  $MAX$ , i.e. the point on the breast border with the furthest distance to the pectoralis muscle

A perturbation analysis gives that the estimated solution is stable in the sense that errors are not magnified. This is important since this show that the error of the estimated distance is not larger than the estimation error of any of the parameters. The most critical parameter is the radius  $r$ . The empirical information presented in [4] were combined with the simulated information to generate an approximation of the 2D PDF of the position of the nipple. In order to do this it was assumed that the position of the nipple along the breast border is independent of how far into the breast the nipple was projected. The result is presented in Fig. 2. Based on the PDF it is possible to derive a narrower ROI for the nipple. For instance, we can say that with 86% certainty that the nipple is within 3 cm from the point denoted  $MAX$ .

### 3.3 Locating the Nipple

The breast is segmented from the background with special care taken to preserve as much as possible of the breast portion of the skin-air interface, including the nipple, if it is in profile [3]. A original mammogram (is shown to the left in Fig. 3) and to the right (in Fig. 3) the same mammogram after the breast border extraction method was applied is shown. For location of the nipple, a method developed by Chandrasekhar and Attikiouzel [1] was implemented, however, modified to search only in our defined ROI.

The breast border is denoted  $B(y)$ . Due to the investigation of the maximum distance  $MAX$  presented in Sect. 3.1 the search areas along the border, is restricted to all integer values of  $y$  running from  $(MAX \pm 4\ell)$ . The  $y$  values lying within these limits is denoted  $y_i$ ,  $\{i = 1, \dots, n\}$ .  $B(y)$  gives the  $x$  value of the skin-air boundary for a given  $y$  in the image. To each of the  $n$  points,  $(B(y_i), y_i)$  the tangent to  $B(y)$  is estimated by the straight line that best fits (in the sense of least squared error) a neighbourhood of  $p$  points on the border, centred on  $y_i$ . The



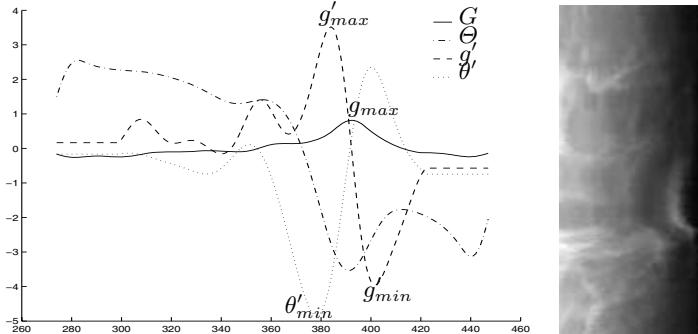
**Fig. 3.** Mammogram (No. 1). To the left is the original mammogram and the middle with the border outlined. To the right, the intensity gradient along the normal is plotted, the pectoralis line  $P$  and the normal to  $P$ ,  $N$  passing through the point  $MAX$  and the point  $MAM$  on the breast border are also plotted. All mammograms are taken from the MIAS database

gradient of this line is denoted by  $m_i$ . The gradient of the normal at  $(B(y_i), y_i)$  is estimated as  $-1/m_i$ , defined with concerns to the search area described above. Associated with this normal is the angle  $\Theta_i$ , which it forms with the positive  $x$ -direction. Pixels intersecting the normal at various distances  $j$ , ( $j = 1, \dots, k$ ), from the test point  $(B(y_i), y_i)$  are identified. The depth in the normal direction, described in Sect. 3.2,  $k$  needs to be 30 mm to be certain to include the nipple. As put in pixels  $30/R$ , where resolution  $R$  is  $400 \mu m$  gives us 75 pixels in the normal direction. The intensity gradient along the normal direction, for each of these,  $I(x_{ji}, y_{ji})$ , is computed as (Fig. 3):  $G_{ji} = \frac{I(x_{ji}, y_{ji}) - I(x_i, y_i)}{j}$ ,  $\forall j \in (1, \dots, k)$  and  $\forall i \in (1, \dots, n)$ . The average of the  $k$  intensity gradients is defined to be the average intensity gradient along the normal,  $G_i = \frac{1}{k} \sum_{j=1}^k G_{ji}$ ,  $\forall i \in (1, \dots, n)$ .

The sequences  $G_i$  and  $\Theta_i$  are smoothed and normalized to yield zero mean, and unit variance, and denoted  $g_i$  and  $\theta_i$ , respectively. These two sequences are passed through a differentiator to yield  $g'_i$  and  $\theta'_i$ . The maximum value of  $g_i$  is found,  $g_{max}$  and its index  $i_{g_{max}}$ . The minimum value of  $\theta'_i$  is also found,  $\theta'_{min}$  and its position,  $i_{\theta'_{min}}$ , see Fig. 4. If  $\theta'_{min}$  is less than a predefined threshold  $t_\theta$ , the nipple is inferred to be in profile otherwise it is not. However, there might exist  $\theta'_{min}$  which is less than  $t_\theta$ , but not in profile. This is handle with a distance measure between  $i_{g_{max}}$  and  $i_{\theta'_{min}}$  [1].

### 3.4 Problems Related to Using Experts for Validation of the Method

Since our main objective is to develop a system for assessing accuracy of the breast positioning we need to investigate how to quantitatively describe quality criteria expressed in imprecise terms such as ..



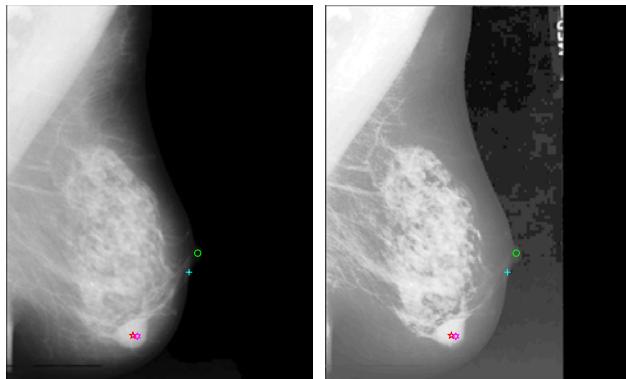
**Fig. 4.** Mammogram No. 101 from MIAS database. Graphs of normal intensity gradient  $g$ , its derivative  $g'$ , the angle  $\theta$  and its derivative  $\theta'$  plotted against  $y$  coordinate of the skin-air interface. To the right, the intensity profile along the normal direction to the breast border is visualized in a grey-level intensity image

. . . In reality, it is difficult (if not impossible) to define rules that determine the exact meaning of such a term in each possible case. Humans tend to base their decision on which is being known to them due to earlier experience in similar situations and can handle the flexibility in the definition of the notions that constitute the rules, in connection with the flexibility of human logic.

To gather data about the way in which human experts assess the quality of mammograms, we performed a questionnaire. It forms the basis for a study in which we are asking radiologists and radiographers, from several countries in Europe, to evaluate 200 randomly selected mammograms from two different standard databases common used during development of computer aided mammography methods. They are asked to mark anatomical features as well as answer questions concerning their decision making. Based on these markings we have seen that there is a large variance between the experts asked and since many medical methods use expert panels as ground truth we wanted to evaluated the accuracy of such comparison.

### 3.5 Definition of the Ground Truth Used for Evaluation of the Method

The accuracy of the implemented automatic nipple location algorithm is evaluated based on the 121 out of the images, described in Sect. 3. The performance measure of the automatic algorithm was evaluated by calculating the vertical distance between the extracted locations of the nipple and to locations defined as the ground truth. The ground truth was defined as follows: For each of the images in MIAS database, the images were enhanced by histogram-equalizing the images so that the nipple could be easily identified and marked by one of the authors. This was carefully performed in a dark room, on a screen with high resolution three times in order to be absolutely sure that the nipple marked is

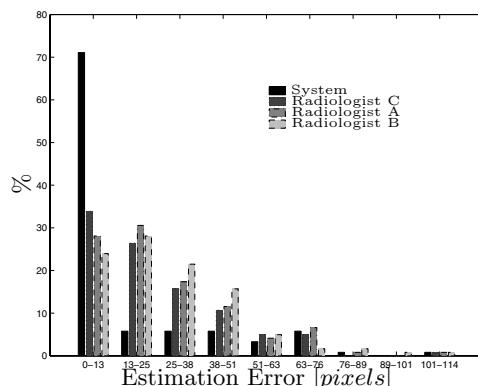


**Fig. 5.** A mammogram is selected for visualizing that the nipple actually can be seen on the histogram-equalize version of the mammograms. Mammogram no. 21 belongs to the subset of the most difficult mammograms to mark by a layperson. Even though this belong to the most difficult subset, we can see by carefully inspecting the image to the right that we can detect and mark the location of the nipple in this image (the circle). The marks of the three experts are the; plus sign, hexagram and pentagram

the true position of the nipple. Even in the most difficult images we can see the nipple in the enhanced images, as shown in Fig. 5.

## 4 Result

A fully automatic method is presented for automatically finding the location and determining whether or not the nipple is in profile. To reduce the complexity



**Fig. 6.** The distance between the ground truth and the proposed location of the nipple by the System (leftmost column in each group), Radiologist C, Radiologist A, and finally Radiologist B (rightmost column). The errors are grouped in intervals of half the diameter of the average nipple (diameter: 23.5 pixels). Thus the first group corresponds to markings of the nipple that are placed within the nipple

of finding the locations and if the nipple was in profile, a successful method for extracting the ROI was developed. Based on this region, the proposed method is tested on 121 randomly selected mammograms from the MIAS database and produces correct result in 71% of the mammograms compared to the ground truth. Our expert panel (consisting of three mammographic experts) produced results like: 34%, 28% and 24% see Fig. 6, compared to the defined ground truth. The results given by our expert panel is only evaluated considering the location of the nipple, not the determination of the quality criteria.

## 5 Discussion

The method for extracting the location of the nipple is based on the fact that at the nipple the glandular-like tissue extends all the way to the breast border. The evaluation of the nipple detection is more thorough (based on more images) than the ones presented in relation to comparable methods. The method has similar performance to the established technique for locating the nipple but with a higher degree of automatization and robustness. The work on detecting the nipple differs in the following points [1]: We make use of geometrical assumptions verified against the MIAS database regarding the location of the nipple along the breast border and we make use of a geometrical model for deciding how far into the breast region that the nipple can occur. These two models allow us to restrict the search for the nipple to an area smaller than the ones proposed by others, but still knowing that we can be very certain that the nipple will occur within the area. Furthermore, the modelling, reduces the need for fixed numbers to determining the search area and thus making the method more general.

The 71% correctly classified nipple features is a lower performance measure than the performance measure reported by Chandrasekhar and Attikiouzel [1]. However, one reason that might reduce the performance measure of our implementation compared to Chandrasekhar and Attikiouzel's [1] implementation is the difficulty of based on an algorithm described in a paper implement and test the same method. Most likely something is either missed to explain by the authors or misinterpreted during our implementation. However, the most interesting is the performance measure difference between our method and the experts compared to the ground truth.

One critical point in our evaluation of the performance measure might be that we actually compare our method to ground truth that we have defined. However, if we observe the image on which we adjusted the contrast during marking, we can see that it is quite obvious on these enhanced images even for laypersons where the locations of the nipple are (Fig. 5).

According to the radiologists' marking compared to the ground truth it appears, knowing that the ground truth is quite true, that many of their markings do not correspond to the location of the nipple. This give rise to the question "what feature do the experts consider the location of the nipple?" One answer might be that our expert panel is asked to mark the anatomical features on

mammogram with lower resolution than they are used to in their clinical practice. Another might be that they are not necessary in their clinical environment while marking these anatomical features. On the other hand, one reference radiologist was, prior to the quality questionnaire study, sent both paper copies and copies printed on copier film of the mammographic database, which could be put on the light box commonly used in their clinical environment considered the mammogram printed on paper to give the highest contrast and was considered suitable for the specific task.

One important thing to remember is that the performance measure of many algorithms in medical imaging is compared to an expert panel, consisting of one to several experts, where the experts often are physician (like our expert panel) working daily with the particular task the algorithm is developed to solve.

Finally, automatic extraction of features in medical images is a well known difficult task. Several pre-processing step might quite often ease the task significantly. Not only considering pre-processing image processing methods but also considering methods to reduce the problem by selecting region of interests (ROI) are useful. However, if this should be useful for the automatic method, these ROI methods needs to not only be dynamic related to some features of the object in focus for extraction (i.e. in our case, related to features of the breast object) but also fully automatic.

## 6 Conclusion

The automatic development of the region of interest and then the evaluation by implementing a known nipple location method show good results based on comparison to the ground truth. This indicates that for future work a more robust and comprehensive location detection algorithm of the nipple might have performance measures with even higher accuracy. Unfortunately since the question "how are we going to evaluate our method since there is, for many medical applications, no objective ground truth available" is still unanswered, we also need to develop objective methods for assessing ground truth, which most be used for evaluating the developed method.

## References

1. R. Chandrasekhar and Y. Attikiouzel. A simple method for automatically locating the nipple on mammograms. *IEEE Transactions on Medical Imaging*, 16:483–494, Oct. 1997.
2. A. J. Mendez, P. G. Tahoces, M. J. Lado, M. Souto, J. L. Correa, and J. J. Vidal. Automatic detection of breast border and nipple in digital mammograms. *Computer Methods and Programs in Biomedicine*, 49:253–262, 1996.
3. C. Olsén. Automatic breast border extraction. In J. M. Fitzpatrick and J. M. Reinhardt, editors, *To appear in proceedings of (SPIE) Medical Imaging*. 2005.

4. C. Olsén and F. Georgsson. The accuracy of geometric approximation of the mamilla in mammograms. In H.U. Lemke, M. V Vannier, A. G. Farman, K. Doi, and J. H. C Reiber, editors, *Computer Assisted Radiology: Proceedings of the international symposium*, Excerpta Medica International Congress Series 1256, pages 956–961. Elsevier, London, 2003.
5. J. L. Semmlow, A. Shadagopappan, L. V. Ackerman, W. Hand, and F. S. Alcorn. A fully automated system for screening xeromammograms. *Computers and Biomedical Research*, 13:350–362, 1980.
6. J. Suckling, J. Parker, D. R. Dance, S. Astley, I. Hutt, C. R. M. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, S-L. Kok, P. Taylor, D. Betal, and J. Savage. The mammographic image analysis society digital mammogram database. In A. G. Gale, S. M. Astley, D. R. Dance, and A. Y. Cairns, editors, *Digital Mammography*, pages 375–378. Elsevier Science, The Netherlands, 1994.
7. E. L. Thurfjell, K. A. Lernevall, and A. A. S. Taube. Benefit of independent double reading in a population-based mammography screening program. *Radiology*, 191:241–244, 1994.

# A Novel Robust Tube Detection Filter for 3D Centerline Extraction\*

Thomas Pock, Reinhard Beichel, and Horst Bischof

Institute for Computer Graphics and Vision, Graz University of Technology,  
Infeldgasse 16/2, A-8010 Graz, Austria  
[{pock, beichel, bischof}@icg.tu-graz.ac.at](mailto:{pock, beichel, bischof}@icg.tu-graz.ac.at)  
<http://www.icg.tu-graz.ac.at>

**Abstract.** Centerline extraction of tubular structures such as blood vessels and airways in 3D volume data is of vital interest for applications involving registration, segmentation and surgical planing. In this paper, we propose a robust method for 3D centerline extraction of tubular structures. The method is based on a novel multiscale medialness function and additionally provides an accurate estimate of tubular radius. In contrast to other approaches, the method does not need any user selected thresholds and provides a high degree of robustness. For comparison and performance evaluation, we are using both synthetic images from a public database and a liver CT data set. Results show the advantages of the proposed method compared with the methods of Frangi et al. and Krissian et al.

## 1 Introduction

As a result of the development of modern volumetric imaging techniques like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), an increasing amount of anatomical and functional details of the human body can be acquired. However, with the benefit of higher spatial and radiometric resolution also the amount of generated data increases. For several applications like diagnosis or surgical planning, . . . structures like blood vessels are of interest. For example, the liver portal vein tree can be used for liver segment approximation, required for tumor resection planning [1]. For 3D applications, such as visualization, segmentation or registration, the detection of vessel centerlines together with radius estimation is an useful preprocessing step. Manual centerline extraction is very time consuming, hence automatic and robust methods for tubular structures would greatly ease this process. In combination with 3D visualization methods, the analysis of the vessels can be substantially improved and simplified, as demonstrated by an Augmented Reality based liver surgery planning system, that has recently been developed [2].

---

\* This work was supported by the Austrian Science Fund (FWF) under the grant P17066-N04.

## 1.1 State of the Art

Medialness functions are used to extract the medial axis of a structure by measuring the degree of belonging to its medial axis. A medialness function can be defined by the convolution product of a kernel  $K(\mathbf{x}, \sigma)$  with a given image  $I(\mathbf{x})$ , where  $\mathbf{x} = (x_1, x_2, x_3)^T$  is a point in 3D space and  $\sigma$  denotes the scale of the measurement. Medialness functions can be classified according to two criteria:

- (a) by the geometric location, where the measurements are made.
  - Central medialness
  - Offset medialness
- (b) by the type of the filter kernel:
  - Linear medialness
  - Adaptive medialness

Frangi et al. [4] developed a vessel enhancement filter based on eigenvalue analysis of the scale space of the Hessian matrix. In terms of classification of medialness functions the vesselness enhancement filter can be classified as linear central medialness, because the eigenvalues are evaluated using a data independent filter kernel exclusively based on the central information. The scale space of the Hessian matrix is given by:

$$\nabla^2 I^{(\sigma)}(\mathbf{x}) = \mathcal{H}^{(\sigma)}(\mathbf{x}) = \sigma^{2\gamma} \left[ \frac{\partial^2 I^{(\sigma)}}{\partial_{x_i} \partial_{x_j}} \right]. \quad (1)$$

Let  $\lambda_1, \lambda_2, \lambda_3$  and  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$  be the eigenvalues and corresponding eigenvectors of  $\mathcal{H}^{(\sigma)}(\mathbf{x})$  such that  $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$  and  $|\mathbf{v}_i| = 1$ . The eigenvalues and eigenvectors correspond to the principal curvatures of the intensity function  $I^{(\sigma)}$ , which is the initial image convolved with a three-dimensional Gaussian kernel  $G(\mathbf{x}, \sigma) = 1/(2\pi\sigma^2)^{3/2} \exp(-(\mathbf{x}^T \mathbf{x})/(2\sigma^2))$ . The term  $\sigma^{2\gamma}$  in Eq. (1) is used for normalization of the second order derivatives, which ensures invariance under image rescaling [9].

The dissimilarity measurement of Frangi et al. takes two geometric ratios into account. The first ratio addresses the deviation from a blob-like structure:  $\mathcal{R}_B = |\lambda_1|/\sqrt{|\lambda_2\lambda_3|}$ . The second ratio is for distinguishing between plate-like and line-like structures and takes into account the two largest second order derivatives:  $\mathcal{R}_A = |\lambda_2|/|\lambda_3|$ . In order to diminish the response of the background pixels, the Frobenius norm of the Hessian matrix is used to define the measure of vesselness:  $\mathcal{S} = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$ . Using these three measures a medialness function is defined:

$$\mathcal{V}_0(\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ \left(1 - e^{-\frac{\mathcal{R}_A^2}{2\alpha^2}}\right) \left(e^{-\frac{\mathcal{R}_B^2}{2\beta^2}}\right) \left(1 - e^{-\frac{\mathcal{S}^2}{2c^2}}\right) & \text{otherwise} \end{cases}, \quad (2)$$

where  $\alpha, \beta$  and  $c$  are thresholds to control the sensitivity of the filter to the measures  $\mathcal{R}_A$ ,  $\mathcal{R}_B$  and  $\mathcal{S}$ . The filter is applied at multiple scales and the maximum response across scales is selected.

$$\mathcal{V} = \max_{\sigma_{min} \leq \sigma \leq \sigma_{max}} \{\mathcal{V}_0(\sigma)\} . \quad (3)$$

Krissian et al. [8] developed an adaptive offset medialness function for 3D brain vessel segmentation. The function is said to be adaptive, because the orientation of the filter kernel is locally adapted by the eigenvectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , which correspond to the two major principal curvatures of the Hessian matrix. Furthermore, the function is classified as an offset medialness function, because the function measures contour information at points equidistant to the center. The medialness function is given by:

$$R(\mathbf{x}, \sigma, \theta) = \frac{1}{2\pi} \int_{\alpha=0}^{2\pi} -\sigma^\gamma \nabla I^{(\sigma)}(\mathbf{x} + \theta\sigma \mathbf{v}_\alpha) \cdot \mathbf{v}_\alpha d\alpha , \quad (4)$$

where  $\mathbf{v}_\alpha = \cos(\alpha)\mathbf{v}_1 + \sin(\alpha)\mathbf{v}_2$  and  $\alpha$  is the angle of the rotating phasor  $\mathbf{v}_\alpha$ . Thus, the function measures the gradient of the smoothed image along a circle with radius  $\theta\sigma$  in the plane defined by  $\mathbf{v}_1$  and  $\mathbf{v}_2$ . In analogy to Eq. (1), the term  $\sigma^\gamma$  is used to normalize the first order derivatives.

The authors use a cylindrical model with Gaussian cross section to analytically compute the value of the proportionality constant  $\theta$ , at which the medialness function gives a maximal response. Having set  $\gamma$  to the value 1, the proportionality constant equals  $\theta = \sqrt{3}$ . Furthermore, for radius estimation, the authors derive a relation between the radius of the vessel  $\sigma_0$  and the scale  $\sigma_{max}$  at which it is detected:

$$\theta\sqrt{\sigma_{max}^2} = \sqrt{\sigma_0^2 + \sigma_{max}^2} \Rightarrow \sigma_0 = \sqrt{2}\sigma_{max} . \quad (5)$$

Similarly to Frangi's method, the function is applied at different scales and the maximum response is selected.

$$R_{multi}(\mathbf{x}, \theta) = \max_{\sigma_{min} \leq \sigma \leq \sigma_{max}} \{R(\mathbf{x}, \sigma, \theta)\} . \quad (6)$$

In [5], two weighting factors were introduced to increase the robustness of the method, when applied to 3D Ultrasound images of the Aorta.

Once the multiscale medialness response is generated, the centerlines can be extracted using the method of Pizer et al. [11]. Considering the multiscale medialness response, it is obvious that the local maxima correspond to the medial axes of the tubes. Therefore, if all the local maxima are located at central points of the tubes, local maxima extraction is equivalent to centerline detection. The characterization of local extrema is based on the properties of the Hessian matrix. A three dimensional image point  $\mathbf{x}$  is considered to be locally maximal in the multiscale medialness response  $R_{multi}$ , if the following condition is satisfied:

$$R_{multi}(\mathbf{x}) \geq R_{multi}(\mathbf{x} \pm \mathbf{v}_1) \quad \text{and} \quad R_{multi}(\mathbf{x}) \geq R_{multi}(\mathbf{x} \pm \mathbf{v}_2) . \quad (7)$$

The eigenvectors  $\mathbf{v}_1, \mathbf{v}_2$  correspond to the two major components of the principal curvatures of the Hessian matrix, obtained from the scale that provides the maximum response. In order to obtain a skeleton like representation of the centerlines, the result of the local maxima extraction is thinned by deleting the simple points [10].

## 1.2 Contribution

In this paper, a novel robust tube detection filter for 3D centerline extraction is presented. The filter is based on an adaptive multiscale medialness function, which combines offset and central information. The function measures boundariness along a circle in the plane defined by the two largest principal curvatures of the Hessian matrix. We further propose a weighting function that takes the symmetry of the tube into account. Thus, responses resulting from isolated structures or high intensity variations are rejected and no background subtraction is needed. To avoid the need for user-selected thresholds, the function includes an adaptive threshold, which is based on a central medianless function and allows better discrimination between tube and non-tube structures. In contrast to the method of Krissian et al., we do not use the same scale for computing both the image gradient and the Hessian matrix. Therefore, our method is less dependent on the vessel model and provides accurate radius estimates. In order to compare our method to the methods of Frangi et al. and Krissian et al., we are using both synthetic images taken from a public database [6, 7, 8] and a liver CT data set. The results show that our method provides a higher accuracy of the centerline and the radius estimate.

## 2 Method

In [8] the proportional parameter  $\theta$  is used, to define a linear relation between the radius of the tube and the scale space parameter  $\sigma$  (Eq.(4)). With this relation, their method is very inflexible especially in providing radius estimates for non-Gaussian tube models. In order to achieve model independence we are using two different scale spaces for computing the Hessian matrix and the boundariness and define a more flexible relation given by

$$\sigma_{\mathbf{B}} = \sigma_{\mathcal{H}}^{\eta} , \quad (8)$$

where  $\eta$  depends on the amount of noise in the image and ranges between  $[0, 1]$ . Thus, the Hessian and boundariness scale spaces are defined by:

$$\mathcal{H}(\mathbf{x}) = \sigma_{\mathcal{H}}^{2\gamma} \left[ \frac{\partial^2 I^{\sigma_{\mathcal{H}}})}{\partial x_i \partial x_j} \right] \quad \text{and} \quad \mathbf{B}(\mathbf{x}) = \sigma_{\mathcal{H}}^{\gamma} \nabla I^{(\sigma_{\mathcal{H}})}(\mathbf{x}) . \quad (9)$$

The boundariness can be represented by  $b(\mathbf{x}) = |\mathbf{B}(\mathbf{x})|$  and  $\mathbf{g}(\mathbf{x}) = \mathbf{B}(\mathbf{x})/|\mathbf{B}(\mathbf{x})|$ , which are the magnitude and the direction of the gradient.

Based on these scale spaces, the method consists of the following steps: First, the initial medialness is computed, using contour information of the boundariness scale space and the filter kernel is locally adapted by means of the Hessian scale space. Second, the initial medialness is weighted by a function that takes the symmetry of the structure into account. Third, a gradient based central medialness function is used to obtain an adaptive threshold. Finally, the function is evaluated at different scales and local maxima are extracted.

## 2.1 Initial Medialness

The initial medialness is given by averaging the contribution of boundariness around a circle of radius  $r$  in the plane defined by the eigenvectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , to the point  $\mathbf{x}$  in the medialness space:

$$R_0^+(\mathbf{x}, r) = \frac{1}{N} \sum_{i=0}^{N-1} b(\mathbf{x} + r\mathbf{v}_{\alpha_i}) c_i^n , \quad (10)$$

where  $N$  is the number of samples and is calculated by  $N = \lfloor 2\pi\sigma + 1 \rfloor$  and  $\alpha_i = (2\pi i)/N$ , respectively. The circularity,  $c_i$ , measures the contribution of the boundariness in radial direction  $\mathbf{v}_{\alpha_i} = \cos(\alpha_i)\mathbf{v}_1 + \sin(\alpha_i)\mathbf{v}_2$  and is additionally constrained by the circularity parameter  $n$ . The choice of the circularity parameter is not very critical,  $n = 2$  being applicable for most images.

$$c_i = \begin{cases} -\mathbf{g}(\mathbf{x} + r\mathbf{v}_{\alpha_i}) \cdot \mathbf{v}_{\alpha_i} & \text{if } -\mathbf{g}(\mathbf{x} + r\mathbf{v}_{\alpha_i}) \cdot \mathbf{v}_{\alpha_i} > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (11)$$

## 2.2 Symmetry Confidence

Eq. (10) also produces responses for isolated edges and non-tube-like structures of high intensity variation. In order to increase the selectivity of the detection, a criterion that takes the symmetry property of the object into account, is introduced. Defining the  $i^{th}$  boundariness sample by  $b_i = b(\mathbf{x} + r\mathbf{v}_{\alpha_i}) c_i^n$ , Eq. (10) can be rewritten as

$$R_0^+(\mathbf{x}, r) = \frac{1}{N} \sum_{i=0}^{N-1} b_i . \quad (12)$$

Considering the distribution of the values  $b_i$ , it is obvious that symmetric structures have a low variance compared to non-symmetric structures. We introduce the variance of the boundariness samples:

$$s^2(\mathbf{x}, r) = \frac{1}{N} \sum_{i=0}^{N-1} (b_i - \bar{b})^2 , \quad (13)$$

where  $\bar{b} = R_0^+(\mathbf{x}, r)$  is the mean boundariness. To quantify the homogeneity of the boundariness along the circle, the initial medialness is weighted by the symmetry confidence

$$\mathcal{S}(\mathbf{x}, r) = 1 - \frac{s^2(\mathbf{x}, r)}{\bar{b}^2} . \quad (14)$$

For circular symmetric structures,  $s^2$  is very low compared to  $\bar{b}^2$  and hence,  $\mathcal{S}(\mathbf{x}, r)$  is approximately one. The larger  $s^2$ , the smaller the value  $\mathcal{S}(\mathbf{x}, r)$ , which results in a reduction in the response to non circular symmetric structures. Thus, the symmetry constrained medialness is defined as:

$$R^+(\mathbf{x}, r) = R_0^+(\mathbf{x}, r) \mathcal{S}(\mathbf{x}, r) . \quad (15)$$

### 2.3 Adaptive Threshold

One problem associated with medialness functions is to find an appropriate threshold to define a minimum response, needed to reject noise and outliers. Considering an arbitrary symmetric cross section profile of a tubular structure, one can notice that the magnitude of the image gradient vanishes at the tube's centerline. Therefore, an adaptive threshold is obtained by the simple fact that, for the tube's center, the medialness must be larger than the magnitude of the center gradient. For this purpose, we introduce the norm of the center gradient

$$R^-(\mathbf{x}, r) = \sigma_{\mathcal{H}}^\gamma |\nabla I^{(\sigma_{\mathcal{H}})}(\mathbf{x})| , \quad (16)$$

which can be classified as a central medialness function. Here, we use the same scale  $\sigma_{\mathcal{H}}$ , to compute the Hessian matrix and the center gradient. Since both features are central ones, the information of the whole structure should be included to provide accurate estimates. Considering a tube of radius  $r'$ , equation (10) is maximized, if the scale parameter  $r$  is equal to  $r'$ . At the same time, the Hessian matrix and the center gradient are computed most stably, if the characteristic width of the Gaussian convolution kernel approximately corresponds to the radius of the tube. Hence, we set  $\sigma_{\mathcal{H}} = r$ . The final medialness is obtained by combining the offset medialness  $R^+(\mathbf{x}, r)$  and the center medialness  $R^-(\mathbf{x}, r)$ :

$$R(\mathbf{x}, r) = \begin{cases} R^+(\mathbf{x}, r) - R^-(\mathbf{x}, r) & \text{if } R^+(\mathbf{x}, r) > R^-(\mathbf{x}, r) \\ 0 & \text{otherwise} \end{cases} . \quad (17)$$

### 2.4 Multiscale Analysis

To take into account the varying size of the vessels, the medialness function  $R(\mathbf{x}, r)$  is evaluated at different radii  $r$ . The multiscale medialness response is obtained by selecting the maximum response over a range of different radii defined by  $r_{min}$  and  $r_{max}$ :

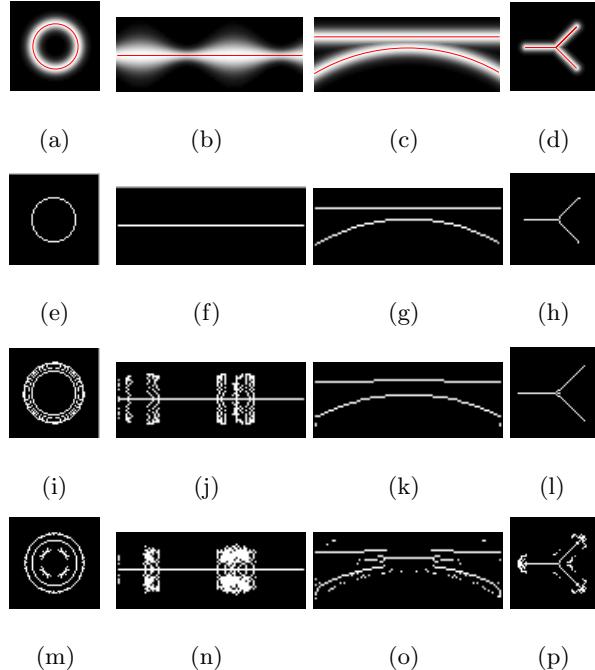
$$R_{multi}(\mathbf{x}) = \max_{r_{min} \leq r \leq r_{max}} \{R(\mathbf{x}, r)\} . \quad (18)$$

The scale  $r$  at which the tube is detected is used to determine the radius of the vessels. In [8], the authors showed how to use the maximum response of their multiscale medialness function to estimate the radius of Gaussian tube models. They also showed, that in the case of bar-like cross sections, the radius estimation fails. However, the use of two separate scale spaces enables tube model independence. In our approach, the accuracy of the radius estimate is only influenced by the amount of image noise.

## 3 Experimental Results

### 3.1 Synthetic Images

We are using four characteristic 3D synthetic images (see Fig. 1) from the public data base of Krissian et al. [6, 7, 8]. Fig. 1 shows the computed centerlines of the



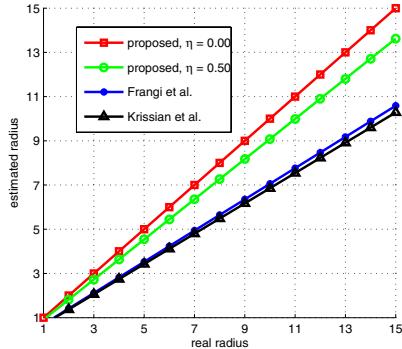
**Fig. 1.** Maximum Intensity Projection (MIP) of the testimages used in evaluation. (a) Tore, (b) Varying Cylinder, (c) Tangent Vessels and (d) Y-Junction. Extracted centerlines: (e)-(h) proposed method, (i)-(l) method of Frangi and (m)-(p) method of Krissian

**Table 1.** Quantitative analysis of centerline accuracy

	Tore			Varying Radius			Tangent Vessels			Y-Junction		
	$R_{pos}$	$\mu_{pos}$	$R_{neg}$	$R_{pos}$	$\mu_{pos}$	$R_{neg}$	$R_{pos}$	$\mu_{pos}$	$R_{neg}$	$R_{pos}$	$\mu_{pos}$	$R_{neg}$
<b>Proposed</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.10</b>	<b>1.00</b>	<b>0.04</b>
Frangi et al.	5.98	5.21	0.00	7.81	11.78	0.00	0.55	1.36	0.49	0.50	3.08	0.06
Krissian et al.	5.62	4.58	0.33	11.12	9.54	0.00	1.03	2.96	0.54	5.40	4.66	0.04

proposed method, the other methods tested. It can be seen that by our method, the centerlines are extracted very accurately. The main reason is, that in regions of high image gradients, the medialness response is eliminated by the adaptive threshold. In addition, the approaches by Frangi and Krissian require thresholds in order to achieve better results.

For quantification, we define three measures,  $R_{pos}$ ,  $\mu_{pos}$  and  $R_{neg}$ , which are the rate of falsely positive centerline voxels, the mean distance in voxel of the falsely centerline voxels to the real centerline, and the rate of falsely negative centerline voxels detected. Table 1 details the quantitative analysis of the extracted centerlines. The centerlines of the models



**Fig. 2.** Results of radius estimation for *bar-like* tubes of radius 1 – 15

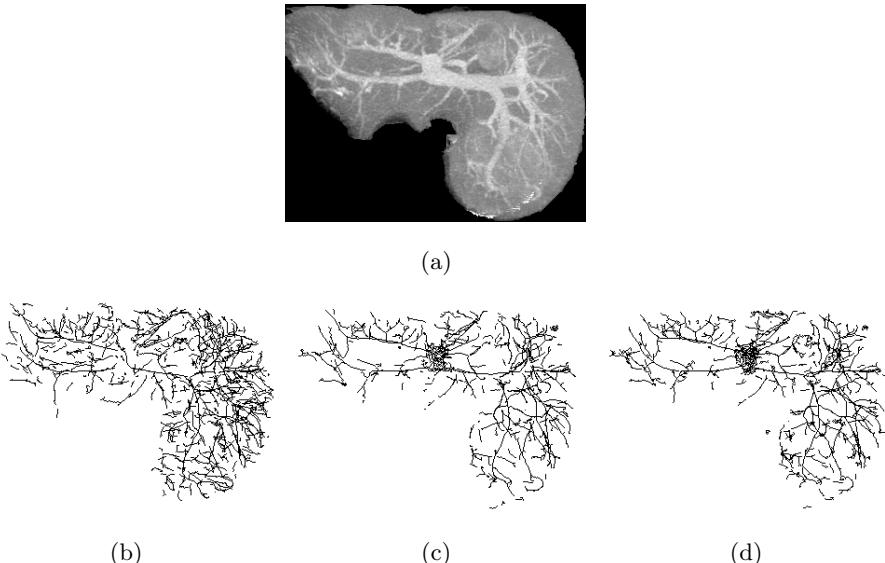
and were accurately extracted. The centerline of the model contains some corrupted voxels, but the mean deviation of the false positive voxels is very small (1 voxel). The following parameters were used:  $\eta = 0.00$  and  $n = 2$ . The parameters of the other approaches were set to standard values as described in [4] and [8].

### 3.2 Performance of Radius Estimation

In real images the intensity variation inside a contrast enhanced vessel is not very high, but the vessel borders are affected by the partial volume effect. Therefore, small vessels are approximated well by a Gaussian model, but big ones are better modeled by a cross section convolved with a Gaussian Kernel with a small standard deviation. For evaluation of the radius estimation we use tube models of radius 1 – 15 voxels, convolved with a Gaussian smoothing kernel of standard deviation 1 voxel. Furthermore, we use different values of the noise parameter  $\eta$ , to demonstrate how this influences the accuracy of the radius estimation. Fig. 2 shows the error of the radius estimation of the proposed method and the methods of Frangi et al. and Krissian et al. The small tubes approximately match the Gaussian model and thus, the methods of Frangi and Krissian do not result in large errors. The more, the tubes approximate the model, the greater is the error associated with these methods. However, the radius estimation of our method does not depend on the specific type of the tube model.

### 3.3 Real Data Sets

We used a liver CT data set which was acquired from routine clinical scan with a voxel size of  $0.55 \times 0.55 \times 2.00 \text{ mm}^3$  and a volume size of  $512 \times 512 \times 97$  voxels. The anisotropic voxels were converted into isotropic voxels using sinc interpolation [3]. For computational speedup, two thresholds  $t_{min}$  and  $t_{max}$  were chosen to define the intensity value range of interest and the computation was limited to regions inside the liver. The following parameters were used:  $\eta =$



**Fig. 3.** Extracted centerlines of a liver portal vein tree. (a) MIP of initial image. Extracted centerlines of (b) the proposed method, (c) method of Frangi and (d) method of Krissian

$0.5$ ,  $n = 2$ ,  $r_{min} = 1$  and  $r_{max} = 15$ . In order to improve the results of the methods of Frangi et al. and Krissian et al., the background voxels defined by  $I(\mathbf{x}) < t_{min}$  were eliminated by setting them to the value  $t_{min}$ . Furthermore, the medialness responses of Frangi and Krissian were thresholded, using the values:  $t_F = 0.05$  and  $t_K = 7.00$ . For both these methods standard parameter settings were used, as described in [4] and [8]. Fig. 3 shows a comparison of the extracted centerlines of a liver portal vein tree for all the different methods. It can be seen, that the complexity and the quality of the centerline extracted by the proposed method is considerably better, than those from the other approaches. In particular, they were less effective at detecting the centerlines of large vessels. Due to intensiy variations inside large vessels, incorrect local maxima in the multiscale medialness response may emerge and result in erroneous centerline pieces. Another disadvantage of the methods of Frangi et al. and Krissian et al. is the need to threshold the medialness response. On one hand, a certain degree of user interaction is necessary to select the threshold, while on the other hand, small vessels providing only a low medialness response are rejected.

## 4 Conclusion and Future Work

In this paper, a novel tube detection filter for 3D centerline extraction was presented. In order to compare our method with the methods of Frangi et al. and

Krissian et al., we used synthetic images and a liver CT data set. The results show the robustness of the method as well as its ability to provide model-independent accurate radius estimates.

Future work will concentrate mainly on special cases of data sets acquired during routine clinical scans with a lot of noise and low contrast between tubes and background.

## References

1. R. Beichel, T. Pock, C. Janko, R. Zotter, B. Reitinger, A. Bornik, K. Palágyi, E. Sorantin, G. Werkgartner, H. Bischof, and M. Sonka. Liver segment approximation in CT data for surgical resection planning. In *Proc. of SPIE*, volume 5370, pages 1435–1446, May 2004.
2. A. Bornik, R. Beichel, B. Reitinger, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Computer aided liver surgery planning: An augmented reality approach. In *SPIE Imaging 2003: Visualization, Image-Guided Procedures and Display*, pages 395–405, February 2003.
3. Y. Du, D. Parker, and W. Davis. Reduction of partial-volume artifacts with zero-filled interpolation in three-dimensional mr angiography,. *Journal of Magnetic Resonance Imaging*, 4(5):733–741, 1995.
4. A. Frangi. *Three-Dimensional Model-Based Analysis of Vascular and Cardiac Images*. PhD thesis, University Medical Center Utrecht, Netherlands, 2001.
5. K. Krissian, J. Ellsmere, K. Vosburgh, R. Kikinis, and C. F. Westin. Multiscale segmentation of the aorta in 3D ultrasound images. In *25th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 638–641, Cancun Mexico, 2003.
6. K. Krissian and G. Farneback. Synthetical test images for vascular segmentation algorithms. <http://lmi.bwh.harvard.edu/research/vascular/SyntheticVessels/SyntheticVesselImages.html>.
7. K. Krissian and G. Farneback. Techniques in the enhancement of 3D angiograms and their application. To appear in Book Chapter.
8. K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trouset. Model-based detection of tubular structures in 3D images. *Computer Vision and Image Understanding*, 80(2):130–171, 2000.
9. Tony Lindeberg and Daniel Fagerstrom. Scale-space with casual time direction. In *ECCV (1)*, pages 229–240, 1996.
10. G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.
11. S. M. Pizer, K. Siddiqi, G. Sékely, J. N. Damon, and S. W. Zucker. Multi-scale medial loci and their properties. *International Journal of Computer Vision*, 55(2/3):155–179, 2003.

# Reconstruction of Probability Density Functions from Channel Representations\*

Erik Jonsson and Michael Felsberg

Computer Vision Laboratory,  
Dept. of Electrical Engineering, Linköping University  
`erijo@isy.liu.se, mfe@isy.liu.se`

**Abstract.** The channel representation allows the construction of soft histograms, where peaks can be detected with a much higher accuracy than in regular hard-binned histograms. This is critical in e.g. reducing the number of bins of generalized Hough transform methods. When applying the maximum entropy method to the channel representation, a minimum-information reconstruction of the underlying continuous probability distribution is obtained.

The maximum entropy reconstruction is compared to simpler linear methods in some simulated situations. Experimental results show that mode estimation of the maximum entropy reconstruction outperforms the linear methods in terms of quantization error and discrimination threshold. Finding the maximum entropy reconstruction is however computationally more expensive.

## 1 Introduction

Many methods in computer vision achieve robustness from a voting and clustering approach. Some examples are the *... , ... , ...* [1, 14], object recognition [10] and view matching [1, 11]. The methods all have in common that each measurement casts a vote, and that clusters in vote space define the output. In high-dimensional spaces, the simple vote histogram as used in the Hough transform is infeasible, since the number of bins will explode.

The construction of soft histograms using the *... , ... , ...* is a practical way to reduce the number of bins required without decreasing the accuracy of the peak detection. Where previous channel decoding methods [6, 4] were designed merely to extract peaks, this paper addresses the more general problem of reconstructing a continuous probability density function from a channel vector. The theoretically sound *... , ... , ...* (Sect. 3) as well as a simpler linear minimum-norm method (Sect. 4.1) is considered. From these continuous reconstructions modes can be extracted, and in Sect. 5 some properties of these modes are compared to those detected by the previous *... , ... , ...* [4].

---

\* This work has been supported by EC Grant IST-2003-004176 COSPAL.

## 2 Channel Representations and Soft Histograms

In the  $\dots, \dots, \dots$  [8, 5, 7], a value  $x$  is encoded into a  $\dots, \dots$   $\mathbf{c}$  using the nonlinear transformation

$$\mathbf{c} = [c_1, \dots, c_N] = [K(x - \xi_1), \dots, K(x - \xi_N)] , \quad (1)$$

where  $K$  is a localized symmetric non-negative kernel function and  $\xi_n, n \in [1, N]$  the  $\dots, \dots, \dots$ , typically located uniformly and such that the kernels overlap. In this paper, we assume integer spacing between channels, and use the quadratic B-spline kernel [15], defined as

$$K(x) = \begin{cases} \frac{3}{4} - x^2 & 0 \leq |x| \leq \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - |x|)^2 & \frac{1}{2} < |x| \leq \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Assuming that we have a number of samples of some feature, each sample can be channel encoded and the corresponding channel vectors summed or averaged. This produces a  $\dots, \dots, \dots$  - a histogram with partially overlapping and smooth bins. In a regular histogram, each sample is simply put in the closest bin, and we cannot expect to locate peaks in such histograms with greater accuracy than the original bin spacing. In a soft histogram however, it is possible to locate peaks with high accuracy at a sub-bin level. A procedure to find such peaks is referred to as a  $\dots, \dots$  of the soft histogram (channel vector). There are several decoding methods [6, 4], all of which are capable of perfect reconstruction of a single encoded value  $x$ .

The representation (1) can be extended into higher dimensions in a straightforward way by letting  $x$  and  $\xi_n$  be vectors, and taking the vector norm instead of the absolute value. Another option is to form the 1D channel representation in each dimension and take the outer product, giving a separable representation.

To make a connection between channel vectors and probability density functions, consider a sample set  $\{x^{(m)}\}$  of size  $M$  and its encoding into channel vectors using  $N$  channels. Let  $\mathbf{c} = [c_1, \dots, c_N]$  be the average of the channel representations of all samples, such that

$$c_n = \frac{1}{M} \sum_{m=1}^M K(x^{(m)} - \xi_n) . \quad (3)$$

If we assume that the samples  $x^{(m)}$  are drawn from a distribution with density  $p$ , the expectation of  $c_n$  is

$$\mathbb{E}[c_n] = \int_{-\infty}^{\infty} p(x) K(x - \xi_n) dx , \quad (4)$$

showing that the elements in  $\mathbf{c}$  estimate some linear features of the density function  $p$  [4]. The main focus of this paper is on how to reconstruct  $p$  from the channel coefficients  $c_n$ .

### 3 Maximum Entropy Method

The problem of reconstructing a continuous distribution from a finite number of feature values is clearly underdetermined, and some regularization has to be used. The natural regularizer for density functions is the  $\dots$ , which measures the information content in a distribution, such that the distribution with maximal entropy is the one which contains a minimum of spurious information [2].

#### 3.1 Problem Formulation

Using the  $\dots$ , the problem becomes the following: Find the distribution  $p$  that maximizes the (differential) entropy

$$H(p) = - \int_{-\infty}^{\infty} p(x) \log p(x) dx \quad (5)$$

under the constraints

$$\int_{-\infty}^{\infty} p(x) K(x - \xi_n) dx = c_n, \quad 1 \leq n \leq N \quad (6)$$

$$\int_{-\infty}^{\infty} p(x) dx = 1. \quad (7)$$

Using variational calculus, it can be shown that the solution is of the form [2, 9]

$$p(x) = k \exp \left( \sum_{n=1}^N \lambda_n K(x - \xi_n) \right), \quad (8)$$

where  $k$  is determined by the constraint (7). It remains now to find the set of  $\lambda_n$ 's which fulfills the constraints (6-7). This is a non-linear problem, and we must resort to numerical methods. In the next section, a Newton method for finding the  $\lambda_n$ 's is presented.

#### 3.2 Newton Method

To make the notation more compact, we introduce a combined notation for the constraints (6 - 7) by defining feature functions  $f_n$  and residuals

$$f_n(x) = K(x - \xi_n) \quad \text{for } 1 \leq n \leq N \quad (9)$$

$$f_{N+1} \equiv 1 \quad (10)$$

$$r_n = \int_{-\infty}^{\infty} p(x) f_n(x) dx - d_n \quad \text{for } 1 \leq n \leq N + 1, \quad (11)$$

where  $\mathbf{d} = [c_1, \dots, c_N, 1]^T$ . In this notation, (8) becomes

$$p(x) = \exp \left( \sum_{n=1}^{N+1} \lambda_n f_n(x) \right). \quad (12)$$

We can now apply a Newton method on the system of equations  $\mathbf{r} = 0$ . Noting that

$$\frac{dr_i}{d\lambda_j} = \int_{-\infty}^{\infty} \frac{dp(x)}{d\lambda_j} f_i(x) dx = \quad (13)$$

$$= \int_{-\infty}^{\infty} f_i(x) f_j(x) p(x) dx , \quad (14)$$

the Jacobian becomes

$$\mathbf{J} = \left[ \frac{dr_i}{d\lambda_j} \right]_{ij} = \left[ \int_{-\infty}^{\infty} f_i(x) f_j(x) p(x) dx \right]_{ij} . \quad (15)$$

The update in the Newton method is now  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \mathbf{s}$ , where the increment  $\mathbf{s}$  in each step is obtained by solving the equations  $\mathbf{Js} = -\mathbf{r}$ .

Since our feature functions  $f_n$  are localized functions with compact support, most of the time  $f_i$  and  $f_j$  will be non-zero at non-overlapping regions such that  $f_i f_j \equiv 0$ , making  $\mathbf{J}$  band-shaped and sparse, and hence easy to invert. The main work load in this method is in the evaluation of the integrals, both for  $\mathbf{J}$  and  $\mathbf{r}$ . These integrals are non-trivial, and must be evaluated numerically.

## 4 Linear Methods

Since the MEM solution is expensive to calculate, we would like to approximate it a simpler and faster approach. In Sect. 4.1, a computationally efficient linear method for density reconstruction is presented, and in Sect. 4.2, the decoding method from [4] is reviewed.

### 4.1 Minimum-Norm Reconstruction

If we relax the positivity constraints on  $p(x)$ , we can replace the maximum-entropy regularization with a  $\| \cdot \|_2$  (MN) regularization, which permits the use of linear methods for the reconstruction. For the derivations, we consider the vector space  $L_2(\mathbb{R})$  of real square-integrable functions [3], with scalar product

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x) g(x) dx \quad (16)$$

and norm  $\|f\|^2 = \langle f, f \rangle$ . The minimum norm reconstruction problem is now posed as

$$p_* = \arg \min_p \|p\|^2 \quad \text{subject to } \langle p, f_n \rangle = d_n \quad \text{for } 1 \leq n \leq N+1 . \quad (17)$$

Reconstructing  $p$  from the  $d_n$ 's resembles the problem of reconstructing a function from a set of frame coefficients [12]. The reconstruction  $p_*$  of minimum norm is in  $Q_1 = \text{span}\{f_1, \dots, f_{N+1}\}$ , which can easiest be seen by decomposing

$p_*$  into  $p_* = q_1 + q_2$ , where  $q_1 \in Q_1$  and  $q_2 \in Q_1^\perp$ . Since  $q_1 \perp q_2$ , we have  $\|p_*\|^2 = \|q_1\|^2 + \|q_2\|^2$ . But  $q_2 \perp f_n$  for all feature functions  $f_n$ , so  $q_2$  does not affect the constraints and must be zero in order to minimize  $\|p_*\|^2$ . Hence  $p_* = q_1 \in Q_1$ , which implies that  $p_*$  can be written as

$$p_*(x) = \sum_n \alpha_n f_n(x) . \quad (18)$$

To find the set of  $\alpha_n$ 's making  $p_*$  fulfill the constraints in (17), write

$$d_n = \langle p_*, f_n \rangle = \left\langle \sum_k \alpha_k f_k, f_n \right\rangle = \sum_k \alpha_k \langle f_k, f_n \rangle , \quad (19)$$

giving the  $\alpha_n$ 's as a solution of a linear system of equations. In matrix notation, this system becomes

$$\Phi \boldsymbol{\alpha} = \mathbf{d} , \quad (20)$$

where  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{N+1}]^T$  and  $\Phi$  is the Gram matrix

$$\Phi = [\langle f_i, f_j \rangle]_{ij} . \quad (21)$$

Note that since  $\Phi$  is independent of our feature values  $\mathbf{d}$ , it can be calculated analytically and inverted once and for all for a specific problem class. The coefficients  $\boldsymbol{\alpha}$  can then be obtained by a single matrix multiplication.

A theoretical justification of using the minimum norm to reconstruct density functions can be given in the case where  $p$  shows just small deviations from a uniform distribution, such that  $p(x)$  is defined on  $[0, K]$ , and  $p(x) \approx K^{-1}$ . In this case, we can approximate the entropy by linearizing the logarithm around  $K^{-1}$  using the first terms of the Taylor expansion:

$$H(p) = - \int_0^K p(x) \log p(x) dx \approx - \int_0^K p(x) (\log K^{-1} + Kp(x)) dx = \quad (22)$$

$$= \log(K) \int_0^K p(x) dx - K \int_0^K p(x)^2 dx . \quad (23)$$

Since  $\int_0^K p(x) dx = 1$ , maximizing this expression is equivalent to minimizing  $\int_0^K p(x)^2 dx = \|p\|^2$ . This shows that the closer  $p(x)$  is to being uniform, the better results should be expected from the minimum-norm approximation.

## 4.2 Efficient Mode Seeking

Assume that we are not interested in reconstructing the exact density function  $p(x)$ , but just need to locate modes of the distribution with high accuracy. An efficient method for this problem, called the ..., is introduced in [4] and briefly reviewed in this section. Let  $\mathbf{c} = [c_1, \dots, c_N]$  be the averaged

channel vector of samples drawn from a distribution  $p(x)$  as earlier. For simplicity, assume that the channel centers are located at the positive integers. We want to do M-estimation on  $p(x)$  by finding minima of the risk function

$$\mathcal{E}(x) = (\rho * p)(x) \quad (24)$$

where  $\rho$  is some robust error norm<sup>1</sup>. The exact shape of this error norm is not that important; what is desired is to achieve good performance in terms of e.g. low channel quantization effects [4]. Choose an error norm  $\rho$  with derivative

$$\rho'(x) = K(x - 1) - K(x + 1) , \quad (25)$$

where  $K$  is our quadratic B-spline kernel from (2). To find extrema of  $\mathcal{E}(x)$ , we seek zeros of the derivative

$$\mathcal{E}' = \rho' * p = (K(\cdot - 1) - K(\cdot + 1)) * p = \quad (26)$$

$$= K(\cdot - 1) * p - K(\cdot + 1) * p . \quad (27)$$

Our channel coefficients  $c_k$  can be seen as the function  $K * p$ , sampled at the integers. By letting  $c'_n = c_{n-1} - c_{n+1}$ , the new coefficients  $c'_n$  become samples of the function  $\mathcal{E}'(x)$  at the integers. To find the zeros of  $\mathcal{E}'$ , we can interpolate these sampled function values using quadratic B-splines again. Since the interpolated  $\mathcal{E}'$  can be written as a linear combination of a number of B-spline kernels (which are piecewise quadratic polynomials), the zeros of this expansion can then be found analytically.

In practise, a recursive filtering [15] or the FFT is used to find the interpolation of  $\mathcal{E}'$ , and the analytic solution of the zero crossings is only determined at positions where the original channel encoding has large values from the beginning, which leads to a computationally efficient method. We will not go into more detail about this, but refer to [4]. The main point is that in this mode seeking scheme, it is the derivative of some risk function which is reconstructed, and not the actual underlying probability density.

## 5 Experimental Results

In this section, the qualitative and quantitative behavior of the two reconstruction methods are compared. Since the MEM requires the numerical evaluation of integrals, all methods are discretized using 400 samples per unit distance. The continuous functions  $p$  and  $f_n$  are replaced by vectors, and the integrals replaced by sums. The channel centers are still assumed to be at the integers of the original continuous domain.

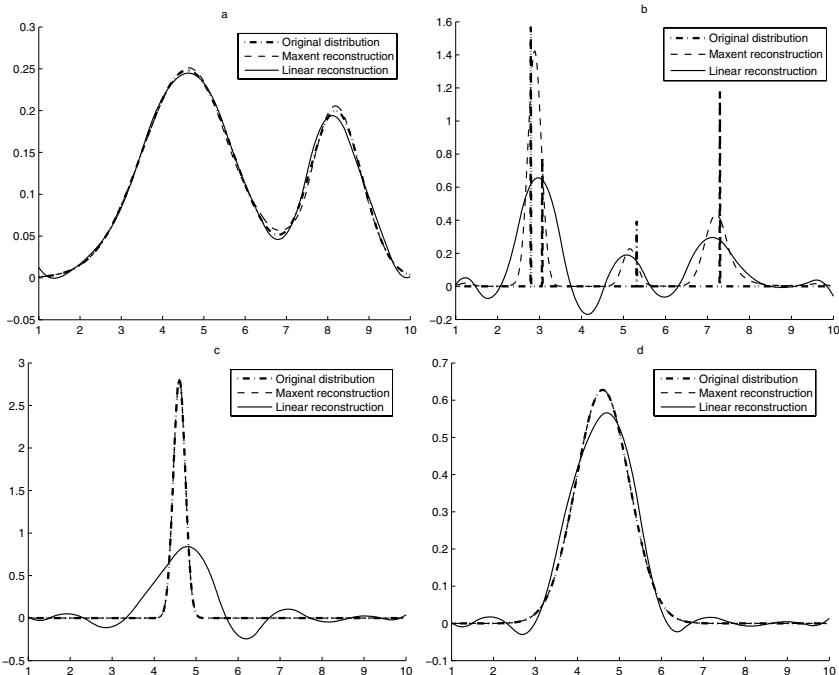
As a channel coefficient  $c_n$  gets closer to zero, the corresponding  $\lambda_n$  from the MEM tends towards  $-\infty$ , leading to numerical problems. To stabilize the solution in these cases, a small background DC level is introduced ( $\epsilon$ -regularization).

---

<sup>1</sup> Some function which looks like a quadratic function close to zero but saturates for large argument values.

### 5.1 Qualitative Behavior

In Fig. 1, the qualitative behavior of the MEM and MN reconstructions are examined. The feature vector  $\mathbf{d}$  was calculated for some different distributions (with the channel centers located at the integers). The two Gaussians (c-d) are reconstructed almost perfectly using MEM, but rather poorly using MN. In (b), the two rightmost peaks are close enough to influence each other, and here even the maximum entropy reconstruction fails to find the exact peak locations. For the slowly varying continuous distribution (a), both methods perform quite well.



**Fig. 1.** The MEM and MN reconstruction of (a) Sum of two Gaussians, (b) 4 Diracs of different weights, (c-d) Single Gaussians with different variance

### 5.2 Quantitative Behavior

To make a more quantitative comparison, we focused on two key properties; the [13] and the [4] of channel decoding. These properties can be measured also for the virtual shift mode seeking.

Recall that the latter decoding finds the minimum of an error function, which is equivalent to finding the maximum of the density function convolved with some kernel. To estimate a similar error function minimum from the continuous reconstructions, our estimated  $p$  should likewise be convolved with some kernel

prior to the maximum detection. Let  $K_{VS} = \rho_{\max} - \rho(x)$  be the kernel implicitly used in the virtual shift decoding. For all continuous reconstruction experiments, peaks were detected from the raw reconstruction  $p$  as well as from  $K * p$  (with the standard B-spline kernel  $K$ ) and  $K_{VS} * p$ .

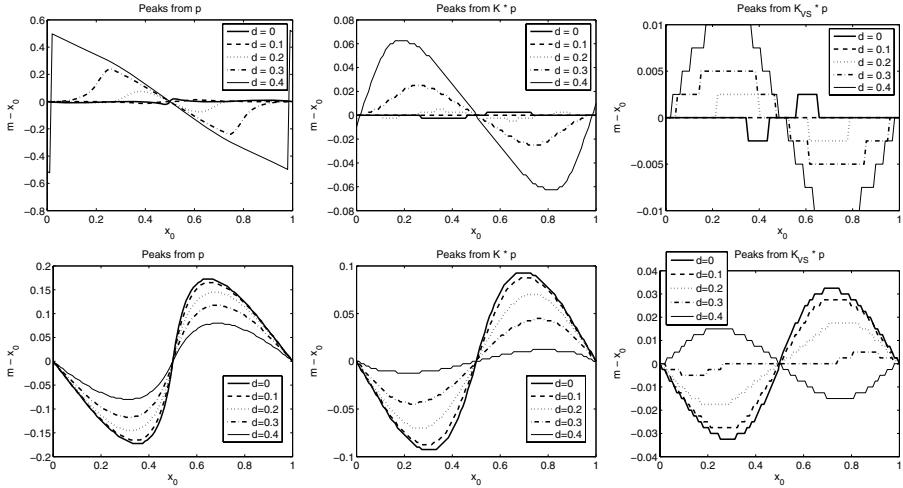
To measure the discrimination threshold, two values  $x_0 \pm d$  were encoded. The discrimination threshold in this context is defined as the minimum value of  $d$  which gives two distinct peaks in the reconstruction. As the background DC level increases, the distribution becomes closer to uniform, and the performances of the MEM and MN methods are expected to become increasingly similar. To keep this DC level low but still avoid numerical problems, we chose a regularization level corresponding to 1% of the entire probability mass. The discrimination threshold was calculated for both reconstruction methods and the different choices of convolution kernels, and the results are summarized in Table 1 for  $x_0$  at a channel center ( $\Delta x_0 = 0$ ) as well as in the middle between two centers ( $\Delta x_0 = 0.5$ ).

**Table 1.** Discrimination thresholds

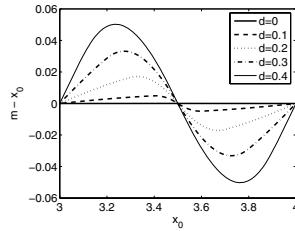
Method		$\Delta x_0 = 0$	$\Delta x_0 = 0.5$
MEM	$p$	0.34	0.57
	$K * p$	0.53	0.71
	$K_{VS} * p$	1.00	1.00
MN	$p$	0.57	0.71
	$K * p$	0.64	0.81
	$K_{VS} * p$	0.95	1.00
Virtual Shift		1.00	1.00

With quantization effect, we mean that two distributions  $p$  differing only in shift relative to the grid of basis functions are reconstructed differently. To measure this effect, two distinct impulses of equal weight located at  $x_0 \pm d$  with  $d$  below the discrimination threshold were encoded. Ideally, the peak of the reconstructed distribution would be located at  $x_0$ , but the detected position  $m$  actually varies depending on the location relative to the grid. Figure 2 shows the difference between the the maximum of the reconstruction and the ideal peak location for varying positions and spacing of the two peaks, using the MEM and MN reconstruction. Figure 3 shows the same effect for the virtual shift decoding. Note the different scales of the plots. Also note that as the error becomes small enough, the discretization of  $p(x)$  becomes apparent.

Achieving a quantization error as low as 1% of the channel distance in the best case in a very nice result, but keep in mind that this error is only evaluated for the special case of two Diracs. For smooth distributions, we expect a lower effect. It is not clear to the authors how to measure this effect in a general way, without assuming some specific distribution.



**Fig. 2.** The quantization effect for continuous reconstructions. Top: Maximum entropy. Bottom: Minimum norm. Left: Raw density estimate. Middle, Right: Density estimate convolved with  $K$ ,  $K_{VS}$



**Fig. 3.** The quantization effect for virtual shift decoding

## 6 Conclusions and Outlook

The maximum entropy reconstruction is theoretically appealing, since it provides a natural way of reconstructing density functions from partial information. In most applications however, the exact shape of the density function is not needed, since we are merely interested in locating modes of the distribution with high accuracy. Efficient linear methods for such mode detection can be constructed, but generally perform worse in terms of quantization error and discriminating capabilities than possible using an entropy-based method. However, as the distributions become more uniform, the performances of the approaches are expected to become increasingly similar.

In general, for wider convolution kernels in the mode extraction, we get better position invariance but worse discriminating capabilities. Thus, there is a trade-off between these two effects. The possibility of achieving as little quantization error as  $\pm 1\%$  of the channel distance opens up for the use of channel-based

methods in high-dimensional spaces. In e.g. Hough-like ellipse detection, the vote histogram would be 5-dimensional, and keeping a small number of bins in each dimension is crucial. Unfortunately, turning the MEM reconstruction method into a practical and efficient mode seeking algorithm is nontrivial, and finding high-performance decoding methods for high-dimensional data is an issue for future research.

## References

1. D.H. Ballard. Generalizing the Hough transform to detect arbitrary patterns. *Pattern Recognition*, 2(13):111–122, 1981.
2. Adam Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):790–799, 1996.
3. Lokenath Debnath and Piotr Mikusiński. *Introduction to Hilbert Spaces with Applications*. Academic Press, 1999.
4. M. Felsberg, P.-E. Forssén, and H. Scharr. Efficient robust smoothing of low-level signal features. Technical Report LiTH-ISY-R-2619, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, August 2004.
5. M. Felsberg and G. Granlund. Anisotropic channel filtering. In *Proc. 13th Scandinavian Conference on Image Analysis*, LNCS 2749, pages 755–762, Gothenburg, Sweden, 2003.
6. Per-Erik Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, March 2004. Dissertation No. 858, ISBN 91-7373-876-X.
7. Per-Erik Forssén and Gösta Granlund. Sparse feature maps in a scale hierarchy. In *AFPAC, Algebraic Frames for the Perception Action Cycle*, Kiel, Germany, September 2000.
8. G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.
9. Simon Haykin. *Neural Networks, A Comprehensive Foundation*. Prentice Hall, second edition, 1999.
10. Björn Johansson. *Low Level Operations and Learning in Computer Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, December 2004. Dissertation No. 912, ISBN 91-85295-93-0.
11. David G. Lowe. Object recognition from local scale-invariant features. In *CVPR’01*, 2001.
12. Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
13. Herman P. Snippe and Jan J. Koenderink. Discrimination thresholds for channel-coded systems. *Biological Cybernetics*, 66:543–551, 1992.
14. M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks / Cole, 1999.
15. Michael Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999.

# Non-rigid Registration Using Morphons

Andreas Wrangsjö, Johanna Pettersson, and Hans Knutsson

Medical Informatics, Department of Biomedical Engineering,  
and Center for Medical Image Science and Visualization,  
Linköping University, Sweden

**Abstract.** The Morphon, a non-rigid registration method is presented and applied to a number of registration applications. The algorithm takes a prototype image (or volume) and morphs it into a target image using an iterative, multi-resolution technique. The deformation process is done in three steps: *displacement estimation*, *deformation field accumulation* and *deformation*. The framework could be described in very general terms, but in this paper we focus on a specific implementation of the Morphon framework. The method can be employed in a wide range of registration tasks, which is shown in four very different registration examples; 2D photographs of hands and faces, 3D CT data of the hip region, and 3D MR brain images.

## 1 Introduction

Image registration is a necessary process in many applications, for example when fusing images from different medical imaging modalities, or correcting geometrically distorted or disaligned images. The goal with the registration procedure is to find a transformation from one image (or volume) to another, such that the difference between the deformed image and the target image is minimised in some sense. There exist a considerable amount of methods for solving this problem. It is not straightforward how to classify different registration methods. They differ in the types of image features they work on, the way they measure similarity, the optimisation scheme, and the types of deformations they allow. A good survey can be found in e.g. [9].

There are several ways to choose what image features to compare. The most basic procedure is to simply compare the image intensity between the two images. More elaborate features are based on intensity gradients, local structure tensors or geometrical landmarks. Furthermore, a way to find the correspondence between the image features must be introduced. This measure should be chosen carefully and with the application in mind to give a good measure of the similarity between the images. Common methods are based on (normalised) cross correlation or mutual information between the image features [8]. With a similarity measure the registration process can be stated as an optimisation problem, or, on the other hand, as a minimisation of the dissimilarity. The transformation model defines how the registered image is allowed to deform. These are typically referred to as global/local or rigid/non-rigid models. A global model

implies that the same transformation model is applied to the entire image, while a local model indicates that different parts of the image may have different transformation models. The rigidity of the model refers to the degrees of freedom of the transformations. A rigid model has a very low number of degrees of freedom and a non-rigid model allows for a higher amount of deformation.

## 2 The Morphon Method

The Morphon essentially attempts to iteratively deform a prototype image (or volume) into a target image by morphing the prototype. The process can be divided into three steps: and

- The displacement estimation aims to find local indications on how to deform the prototype to make it more similar to the target image. In the implementation presented here, these estimates are based on quadrature phase differences.
- The deformation field accumulation uses the displacement estimates to update a deformation field. This step involves two processes: updating the accumulated deformation field and regularisation of the displacement estimates and/or accumulated field. This regularisation is used to fit the observed deformation to a local and/or global deformation model.
- Finally, the deformation morphs the original prototype according to the accumulated deformation field. In this implementation conventional bi/trilinear interpolation has been used for this purpose.

These three steps are iterated for as long as the displacement estimates indicate further morphing to be beneficial. Furthermore, a scale space scheme has been included in the Morphon. This enables successful morphing also between quite dissimilar images.

The method is so far indeed quite general. Its usability is, however, quite dependent on what methods one makes use of in following the steps above. The remainder of this section is devoted to the methods we have chosen in our implementation.

### Quadrature Phase Difference

Estimating optical flow, motion, orientation or displacement between images are essentially equivalent problems. Being very typical image analysis issues, a multitude of methods have evolved to cope with them. There are, among others, methods based on gradient [7, 1], polynomial expansions [2] and quadrature phase [4, 5, 6]). We have chosen to use quadrature phase differences to estimate the local displacement between the two images. Among the benefits of this method are invariance to image intensity and weak gradients.

Quadrature phase can be described as a measure of local structure. Edges between dark and bright areas in the image have one phase, lines on dark background have one, bright patches have one and dark lines have one. The transition

between local phase values is continuous as we move e.g. from a dark patch across an edge and into a bright patch. The difference in local phase between prototype and target images is therefore a good measure of how much the prototype needs to be moved (locally) to fit the target image. The phase is, however, by definition one-dimensional. For multi-dimensional signals this means that the phase needs to either be associated with a certain direction [5] or redefined to be meaningful in a multi-dimensional sense [3]. By using a set of multidimensional quadrature filters, each associated with a direction and an angular function, we can obtain phase and magnitude estimates corresponding to the filter directions. If these are chosen carefully and the angular functions are chosen to constitute a spherical harmonics basis, we can estimate any local orientation and phase of the image. If each region of the image was a  $\dots$ , where the image intensity could be described as a function of only one variable, we would be able to find a perfect estimate of the local orientation and an ideal measure of local phase (given the spatial frequency function of the filters). The local displacement is then found as a function of the local phase along its associated direction. Images are not simple, however, and we need to settle with an approximation of the local phase and its direction. Here, we have used a least square estimate of the local displacement.

$$\min_{\mathbf{v}} \sum_i [w_i(\hat{\mathbf{n}}_i^T \mathbf{v} - v_i)]^2 \quad (1)$$

where  $\mathbf{v}$  is the sought displacement field estimate,  $\hat{\mathbf{n}}_i$  is the direction of filter  $i$ ,  $v_i$  is the displacement field associated to filter direction  $i$ , and  $w_i$  is a certainty measure, derived from the magnitude of the phase difference. By using this estimation scheme we obtain a fairly good estimate of local displacement. Note, however, that these estimates are always perpendicular to the local structure. This is due to the  $\dots$ . If the images are e.g. two lines of the same orientation, we can only estimate the orthogonal distance between the lines. Displacement along the lines is, and will always be, impossible to estimate.

Using quadrature phase in displacement estimation gives us a quite simple and useful certainty measure, briefly mentioned in Eq. (1) above. If the filter response magnitude at some position in the image is large, we are more inclined to trust that phase estimate than if it is small. It is thus beneficial a function of the magnitude as certainty for the displacement estimates.

## Deformation

Eq. (1) gives us a displacement field for the current iteration and scale. The displacement field should be used to interpolate a new, deformed version of the prototype. However, the interpolation step introduces small errors in the data. To avoid these errors to escalate when the prototype is iteratively interpolated, we accumulate the displacement fields into one total field that represents how the original prototype should be morphed. For each iteration the original prototype is deformed according to the accumulated field and then compared to the target data to estimate a displacement field for the current iteration.

The updated accumulated field,  $\mathbf{d}'_a$ , is obtained by combining the accumulated field,  $\mathbf{d}_a$ , and the displacement field from the current iteration,  $\mathbf{d}_k$ . This accumulation also includes certainty measures for both the accumulated field,  $c_a$ , and the temporary field,  $c_k$ .

$$\mathbf{d}'_a = \frac{\mathbf{d}_a c_a + (\mathbf{d}_a + \mathbf{d}_k) c_k}{c_a + c_k} \quad (2)$$

## Regularisation

The local displacement estimates could be interpreted as local forces trying to pull the prototype their way. If we would simply deform the prototype by moving each pixel as suggested by the displacement estimate, we would surely end up with a completely disrupted image. Hence, there is need for regularisation of the displacement estimates.

Our deformation field can be regularised in a multitude of ways. As described in the introducing section, registration methods incorporate a great variety of deformation models. We could, e.g. find a (weighted) least squares fit to some global deformation model. In the simplest case, we could fit it to a global rotation and translation model. This would correspond to rigid registration. The only allowed deformations would be those that can be described as a global rotation and/or translation. We could also give the model more degrees of freedom. Scaling, skewing, etc. could be added to our global model. The model could also be regional, e.g. a FEM model, where the observed displacements would be fitted according to their nearest landmark (node) points in a mesh model. The model could also be local. That is, we can fit the displacement estimates locally according to some neighbourhood function. The simplest case is to perform a local averaging of the estimates. The regularisation method could also be altered or modified during the registration. We could e.g. start out using a global model to find a good initial deformation. The regularisation could then be adapted towards a local model after a few iterations. Thus the benefits of global and local registration can be incorporated in the same model.

## Certainty Measures

As mentioned earlier, certainty measures can be computed for the displacement estimates. Typically, edges, lines and surfaces are structures where the quadrature filters give strong responses. Using the response amplitude as a certainty measure would, thus, yields large certainties in such regions. If e.g. a global least squares fit to some deformation model is used in the regularisation, these certainties are used as weights in weighted LSQ. If, on the other hand, we use a local averaging or similar, the conventional averaging convolution operator is replaced by normalised averaging [5].

## Scale Spaces

Local displacement is usually scale dependent. It is perfectly possible for two images to be horizontally displaced when observed on a fine scale but vertically

if you look on a coarse scale. A typical problem in registration is to know what features of the prototype image to associate with what features of the target image. Our approach is to start comparing the images on a very coarse scale. Once the images are registered on that scale we move on to a finer scale. This simple scale space scheme has proven to be quite successful.

### 3 Experiments

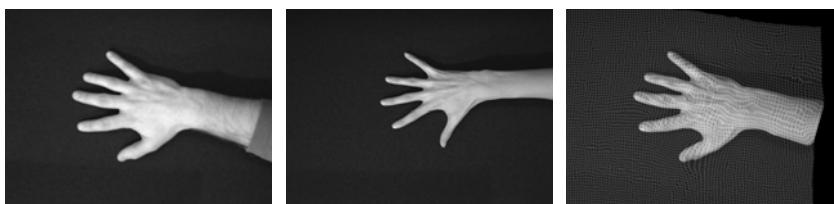
To demonstrate some of the capabilities of Morphon registration, a set of four quite different registration tasks are presented along with experimental results using the Morphon.

#### Hands

A set of hands were photographed. Since the hands belong to different people, there are differences in size and shape. There are also differences in pose between the photos. Clearly, any rigid registration would fail in this case. Two experiments were carried out on the hands. The first one makes use of a local normalised Gaussian averaging model in the regularisation of the deformation field. The second uses a global affine regularisation model (which limits the allowed deformation quite a lot more than the Gaussian one). Figures 1 and 2 show the results when using the normalised Gaussian averaging and figures 3 and 4 show the same results using the affine model.

#### Faces

The next example deals with registration of images of faces. We show results both for registration between the faces of two different persons, Fig. 5, as well as registration of the same face but with different facial expression, Fig. 6. In both cases, a local transformation model based on local normalised Gaussian averaging, has been used. This example is more difficult than the hand example. Here, the images consist of objects that do not necessarily look at all the same. Even in the case where the same person is used as both target data and prototype data the images can look quite different just by changing the pose somewhat. An ear might show in one image and not in the other.



**Fig. 1. Left:** The target data hand, **Middle:** The hand to be deformed, **Right:** The deformed hand (with an added grid to show the deformation). Here, a normalised Gaussian regularisation has been used



**Fig. 2.** The results from normalised Gaussian hand registration. **Left:** The registered hand (mirrored), **Right:** The 'real' hand



**Fig. 3.** **Left:** The target data hand, **Middle:** The hand to be deformed, **Right:** The deformed hand (with an added grid to show the deformation). Here, an affine regularisation has been used



**Fig. 4.** The results from affine hand registration. **Left:** The registered hand (mirrored), **Right:** The 'real' hand

## Hips

The hip example demonstrates the Morphon method applied to 3D CT data. The goal is to automatically segment the femoral bone and the pelvic bone as two separate objects from the data. Classification of tissue classes in CT images is usually done by simple thresholding, but due to osteoporosis, the bone density



**Fig. 5.** **Left:** The image to deform, **Middle:** The target image, **Right:** The result after morphing the leftmost image onto the middle image

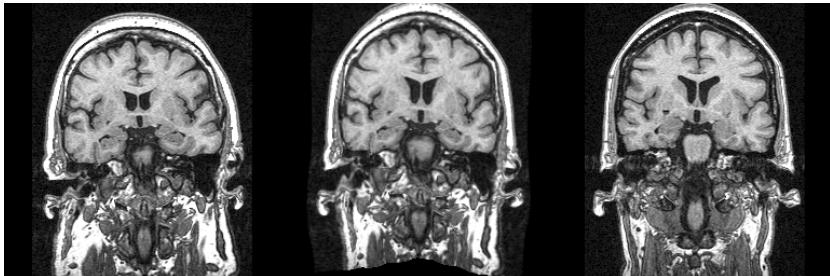


**Fig. 6.** **Left:** The prototype image with a happy expression, **Middle:** The target image with a sad expression, **Right:** Result from morphing the happy face onto the sad face



**Fig. 7.** 2D coronal slices of the hip data. **Left:** The hand-segmented data used as prototype, **Middle:** The CT data, **Right:** The prototype after deformation

is reduced to levels below that of the surrounding soft tissue. This gives CT images where the bone and the surrounding tissue have similar intensity levels, and where the border between the two is not always clearly distinguishable. Moreover, the joint space separating the pelvis and the femoral head is in most cases not possible to detect because of the low resolution of the image data. A prototype containing a hand segmented femur and pelvis, with basically two



**Fig. 8.** **Left:** Prototype volume (brain a), **Middle:** Deformed prototype (brain a)  
**Right:** Target volume (brain b)

gray levels, is registered to the patient data. When the registration process is completed the femur and pelvis, now transformed into the shape of the specific patient, can be segmented from the deformed prototype. 2D coronal slices of the original prototype, the target data and the deformed prototype are shown in figure 7. A local transformation model has been used.

### Brains

The final experiment was performed on 3D MRI-data of two brains. This is quite a challenge since no two brains look the same. One should most likely think carefully what one actually wishes to accomplish with such a registration. In this case we are interested only in the region around the left and right hippocampus. That is, we do not even wish to deform every winding section of the brains to look the same. This drastically reduces the complexity of the (still quite tricky) operation.

Figure 8 shows one section of the two original MRI volumes (a and b) along with the deformed version of the prototype volume (b).

## 4 Discussion

The application range for non-rigid registration techniques is wide, which explains the large amount of research devoted to this area. Deformation models, displacement estimation and similarity measures, are some relevant concepts associated to these methods. These concepts can be thought of as research areas of their own, and by combining these elements in new ways novel registration schemes can be found. The Morphon algorithm presented here can be characterized as a general registration method. It can be adapted to quite varying applications, which has been proven by the examples shown in this paper.

This paper does not, however, contain a comparison of the Morphon method to other well-known registration techniques, which is necessary to evaluate its functionality. In spite of this, we will point out some features that in much suggest that the Morphon method is a competitive registration algorithm.

One advantage with the Morphon method is that it uses techniques which makes it robust in several senses. By using the quadrature phase as an input for measuring the similarity between the images, and obtaining the displacement estimation from the differences in the filter outputs, the algorithm becomes less sensitive to variations in the intensity of the data. This is beneficial in cases where the intensity level in the data is not closely related to the content of the data. One example of this is when working with photos, such as the hands or faces in the above examples. Due to different lighting conditions when obtaining the images, and difference in skin colour and reflectance, it would be very difficult to register these images to each other by only taking the gray values into account. However, in cases where the image intensity actually has a significance, such as in CT data where the intensity values directly correspond to the type of tissue, it would be disadvantageous not to utilize this knowledge. Thus, a method considering both the phase and the image intensity would be optimal in such cases.

Furthermore, because of the certainty measures included in the registration process the method becomes more stable. By applying these certainties as weights both in the estimation and the accumulation of the displacement fields the resulting deformation mainly depends on the neighbourhoods where the similarity between the images is large.

## **Future Issues and Applications**

The general Morphon framework is definitely not limited to the implementations presented here. Since the Morphon components are fairly independent of each other, one can easily plug in new displacement estimation or regularisation methods when needed. Among the features one could imagine to incorporate is more prior information on the images. If e.g. images of hands are to be registered, a hand-specific deformation model would be reasonable.

The Morphon framework is here presented as a registration scheme. Its most likely application area would be to use such registration to perform atlas-based segmentation. All the prior information on the studied objects is incorporated in the prototype and morphed along with it. After registration to a target image, the morphed prior information can be used to draw conclusions about the target image.

As mentioned earlier, the Morphon in its current shape makes use of local measures to find a deformation field corresponding to a low cost. Most registration methods are based on a global energy measure of some kind which one would then attempt to minimise. Not seldom will the minimisation process involve a local formulation of some kind, meaning that the practical difference will not be enormous. It could, however, be beneficial to incorporate some kind of global measure also in the case of the Morphon. One problem now is that we do not consider whether the displacement estimates are made between structures that correspond to each other or to some completely different structure. To some extent this could be dealt with by the deformation model. In other cases we might wish to adapt the displacement estimation model itself to look at e.g. the likelihood of a certain voxel belonging to a certain tissue class or similar.

Creating atlases is one very likely application for the Morphon method. By performing a number of user-guided Morphon registrations on a set of data, a database of typical deformation fields is obtained. This database can be used to create problem-specific prototypes where not only the intensity of the depicted objects is known, but also statistical models of shape variations similar to those used in active shape modeling.

## 5 Conclusions

The Morphon method was presented as a 2D and 3D non-rigid registration method. Its capabilities were successfully demonstrated on four very different applications. The results are promising but further evaluation, and comparison to existing registration techniques, must be done to assess its functionality. Finally a number of future extensions to adapt the Morphon method to very challenging problems such as atlas based segmentation were suggested.

## Acknowledgement

We would like to thank Mats Andersson for interesting discussions on Morphons and more. We are also grateful to our project funders; Vetenskapsrådet, Vinova and SSF, along with our research partners; Helge Malmgren at Göteborgs University and Melerit AB, Linköping.

## References

1. J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43–77, 1994.
2. G. Farnebäck. *Polynomial Expansion for Orientation and Motion Estimation*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 2002. Dissertation No 790, ISBN 91-7373-475-6.
3. M. Felsberg. Disparity from monogenic phase. In L. v. Gool, editor, *24. DAGM Symposium Mustererkennung, Zürich*, volume 2449 of *Lecture Notes in Computer Science*, pages 248–256. Springer, Heidelberg, 2002.
4. D. J. Fleet and A. D. Jepson. Computation of Component Image Velocity from Local Phase Information. *Int. Journal of Computer Vision*, 5(1):77–104, 1990.
5. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
6. M. Hemmendorff. *Motion Estimation and Compensation in Medical Imaging*. PhD thesis, Linköping University, Sweden, SE-581 85 Linköping, Sweden, 2001. Dissertation No 703, ISBN 91-7373-060-2.
7. B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.
8. J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.
9. B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

# Hybridization of the Ant Colony Optimization with the K-Means Algorithm for Clustering

Sara Saatchi and Chih Cheng Hung

Department of Computer Science,  
Southern Polytechnic State University,  
1100 South Marietta Parkway, Marietta, GA 30060, USA  
`{ssaatchi, chung}@spsu.edu`

**Abstract.** In this paper the novel concept of ACO and its learning mechanism is integrated with the K-means algorithm to solve image clustering problems. The learning mechanism of the proposed algorithm is obtained by using the defined parameter called pheromone, by which undesired solutions of the K-means algorithm is omitted. The proposed method improves the K-means algorithm by making it less dependent on the initial parameters such as randomly chosen initial cluster centers, hence more stable.

## 1 Introduction

Image segmentation plays an essential role in interpretation of the various kinds of images. Image segmentation techniques can be grouped into several categories such as Edge-based segmentation, Region-oriented segmentation, Histogram-thresholding, and clustering algorithms [1]. The aim of clustering algorithm is to aggregate data pixels into groups or clusters such that pixels in each group are similar to each other and different from other groups. There are various techniques for clustering including the K-means algorithm which is based on the similarity between pixels and the specified cluster centers. The behavior of the K-means algorithm mostly is influenced by the number of clusters specified and the random choice of initial cluster centers. In this study we concentrate on the latter, where the results are less dependent on the initial cluster centers chosen, hence more stabilized, by introducing a hybrid technique using the K-means and the Ant Colony Optimization (ACO) heuristic.

The ACO algorithm was first introduced and fully implemented in [2] on the traveling salesman problem (TSP) which can be stated as finding the shortest closed path in a given set of nodes that passes each node once. The ACO algorithm is one of the two main types of swarm intelligent techniques. Swarm intelligence is inspired from the collaborative behavior of social animals such as birds, fish and ants and their amazing formation of flocks, swarms and colonies. By observing and simulating the interaction of these social animals, a variety of optimization problems are solved. The other type of swarm intelligent techniques is the particle swarm optimization (PSO) algorithm. The algorithm consists of a swarm of particles flying through the search space [3]. Parameters of position and velocity are used to represent the potential solution to the problem and the dynamic of the swarm respectively. The velocity of

each particle is modified according to its experience and that of its neighbors which changes the position of the particle in order to satisfy the objective. The ACO algorithm which we are focused on, is based on a sequence of local moves with a probabilistic decision based on a parameter, called pheromone as a guide to the objective solution. There are algorithms that while they follow the cited procedure of ACO algorithm, they do not necessarily follow all the aspects of it, which we informally refer to as ant-based algorithm or simply ant algorithm.

There are several ant-based approaches to clustering which are based on the stochastic behavior of ants in piling up objects. In most of the approaches [4-8], the ant algorithm is applied to numeric data which are treated as objects. Ants pick up an object and drop it on a heap where there is the most similarity according to a probability. In [4, 5, 6] the ant based algorithm is applied initially to form the initial set of clusters, and then the K-means algorithm is used to remove classification error and to assign objects that are left alone, to a cluster. The ant based algorithm is applied again on heaps of objects rather than single object and then the K-means algorithm is used for the same reason as before. The algorithm resolves the problem of local minimal by the randomness of ants. In [4] the idea of density is introduced to determine where the heaps and scattered objects are to increase the searching speed and efficiency of the algorithm. Although in [4, 5, 6] the K-means algorithm was used, only the stochastic movement of the ants are considered, not the learning mechanism of the ACO through the so called pheromone. In [8] a similar approach is used except that after the ant based initialization, Fuzzy C-Means (FCM) algorithm is used to refine these clusters. In [9] the ACO algorithm is combined with genetic algorithms and simplex algorithms to build a learning model for optimal mask, which is used to discriminate texture objects on aerial images. The ACO has been applied to various problems. An ACO heuristic was introduced for solving maximum independent set problems in [10] where uses the pheromone concept to obtain a learning mechanism.

In this study the ACO algorithm is implemented and it is integrated with the K-means algorithm to solve clustering problems and it is applied to images. For clustering as the goal,  $m$  number of ants is chosen for clustering. Solutions to the problem will be generated by the ants independently. The best solution will be chosen from these solutions and the assigned pheromone to this solution is incremented. The next  $m$  ants are inspired by the previous  $m$  ants and start making their own solution. This is repeated until a certain number of iterations when the optimal solution is achieved. Decision based on the pheromone amount may be able to eliminate the undesired solutions which are probable to occur, making the algorithm more stable and the solutions more desirable. The advantage of this algorithm over the K-means is that the influence of the improperly chosen initial cluster centers will be diminished after the best solution is chosen and marked with the pheromone over a number of iterations. Therefore it will be less dependent on the initial parameters such as randomly chosen initial cluster centers and more stabilized while it is more likely to find the global solution rather than the local.

In Section 2 the K-means algorithm is reviewed briefly. Section 3 describes the ACO algorithm. In Section 4 the proposed hybrid ACO-K-means clustering algorithm is discussed. Experimental results are presented in Section 4 and the conclusion and future work are discussed in Section 5.

## 2 The K-Means Algorithm

The K-means algorithm, first introduced in [11] is an unsupervised clustering algorithm which partitions a set of object into a certain number of clusters. The K-means algorithm is based on the minimization of a performance index which is defined as the sum of the squared distances from all points in a cluster domain to the cluster center [12]. First K random initial cluster centers are chosen. Then each sample is assigned to a cluster based on the minimum distance to the cluster centers. Finally cluster centers are updated by calculating the average of the values in each cluster and this will be repeated until cluster centers no longer change.

The K-means algorithm tends to find the local minima rather than the global therefore it is heavily influenced by the choice of initial cluster centers and the distribution of data. Most of the time the results become more acceptable when initial cluster centers are chosen relatively far apart since the main clusters in a given data are usually distinguished in such a way. If the main clusters in a given data are close in characteristics the K-means algorithm fails to recognize them when it is left unsupervised. For its improvement the K-means algorithm needs to be associated with some optimization procedures in order to be less dependent on a given data and initialization.

## 3 The ACO Algorithm

The ACO heuristic has been inspired by the observation on real ant colony's foraging behavior and on that ants can often find the shortest path between food source and their nest [10]. This is achieved by a deposited and accumulated chemical substance called pheromone by the passing ant which goes towards the food. In its searching the ant uses its own knowledge of where the smell of the food comes from (we call it as heuristic information) and the other ants' decision of the path toward the food (pheromone information). After it decides its own path, it will confirm the path by depositing its own pheromone making the pheromone trail denser and more probable to be chosen by other ants. This is a learning mechanism ants follow beside their own recognition of the path. As a result of this consulting of ants with the ants already done a job, the best path which is the shortest will be marked from the nest towards the food.

ACO uses this learning mechanism. Furthermore, in the ACO algorithm, the pheromone level is updated based on the best solution obtained by a number of ants and the pheromone amount that is deposited by the succeeded ant is defined to be proportional to the quality of the solution it produces. For the real ants, the best solution is the shortest path and it will be marked with a strong pheromone trail. In the short path problem using ACO algorithm, the pheromone amount deposited is inversely proportional to the length of the path. For a given problem the pheromone can be set to be proportional to any criteria of the desired solution. In the clustering method we will introduce below, the criteria includes the similarity of data in each cluster, distinction of the clusters and compactness of them. More details on the ACO algorithm can be found in [2].

## 4 The Proposed Hybrid ACO-K-Means Clustering Algorithm

Our approach to the problem is very similar to the K-means strategy. It starts by choosing the number of clusters and a random initial cluster center for each cluster. ACO plays its part in assigning each pixel to a cluster. This is done according to a probability which is inversely dependent to the distance (similarity) between the pixel and cluster centers and a variable,  $\tau$ , representing the pheromone level. Pheromone is defined to be dependent to minimum distance between each pair of cluster centers and inversely dependent on the distances between each pixel and its cluster center. So the pheromone gets bigger when cluster centers get far apart and clusters tend to be more compact (our criterion for best solution), making the probability of assigning a pixel to that cluster high. Pheromone evaporation is considered to weaken the influence of the previously chosen solutions, which are less likely to be desired. Similar to the K-means algorithm, at this stage new cluster centers are updated by calculating the average of the pixels in each cluster and this will be repeated until cluster centers no longer change. But unlike K-means, this algorithm doesn't stop here. It is assumed that the described clustering job is performed by an ant, and there are  $m$  ants repeating this job, each with their own random initialization and they all will end up with a solution. A criterion is defined to find the best solution and the pheromone level is updated accordingly for the next set of  $m$  ants as a leading guide. A termination criterion will stop the algorithm and an optimal solution is resulted.

The algorithm starts by assigning a pheromone level  $\tau$  and a heuristic information  $\eta$  to each pixel. Then each ant will assign each pixel to a cluster with the probability  $P$ . This assignment of parameters is shown in Table 1, where  $P$  is obtained from Eq. 1 [2]:

$$P_{(i,Xn)} = \frac{\tau_{(i,Xn)}^\alpha \eta_{(i,Xn)}^\beta}{\sum_{i=0}^K \tau_{(i,Xn)}^\alpha \eta_{(i,Xn)}^\beta}. \quad (1)$$

where  $P_{(i,Xn)}$  is the probability of choosing pixel  $X_n$  in cluster  $i$ ,  $\tau_{(i,Xn)}$  and  $\eta_{(i,Xn)}$  are the pheromone and heuristic information assigned to pixel  $X_n$  in cluster  $i$  respectively,  $\alpha$  and  $\beta$  are constant parameters that determines the relative influence of the pheromone and heuristic information, and  $K$  is the number of clusters. Heuristic information  $\eta_{(i,Xn)}$  is obtained from:

$$\eta_{(i,Xn)} = \frac{\kappa}{ColDist(X_n, C_i) * PhysDist(X_n, C_i)}. \quad (2)$$

where  $X_n$  is the  $n^{th}$  pixel and  $C_i$  is the  $i^{th}$  cluster center.  $ColDist(X_n, C_i)$  is the color distance between  $X_n$  and  $C_i$ , and  $PhysDist(X_n, C_i)$  is the physical (geometrical) distance between  $X_n$  and  $C_i$ . Constant  $\kappa$  is used to balance the value of  $\eta$  with  $\tau$ .

The value for the pheromone level  $\tau$  assigned to each pixel is initialized to 1 so that it doesn't have effect on the probability at the beginning. This pheromone should become bigger for the best solution we are looking for.

**Table 1.** Assignment of parameters to each pixel in each clustering solution obtained by an ant

Cluster# \ Pix	X <sub>0</sub>	X <sub>1</sub>	...	X <sub>n</sub>
0	P <sub>(0,X0)</sub>	P <sub>(0,X1)</sub>		P <sub>(0,Xn)</sub>
1	P <sub>(1,X0)</sub>	P <sub>(1,X1)</sub>		P <sub>(1,Xn)</sub>
2	P <sub>(2,X0)</sub>	P <sub>(2,X1)</sub>		P <sub>(2,Xn)</sub>
...				

Suppose  $m$  number of ants is chosen for clustering on an image. Each ant is giving its own clustering solution. After  $m$  ants have done their clustering, the current best solution is chosen and the assigned pheromone to this solution is incremented. Also cluster centers are updated by the cluster centers of the current best solution. The next  $m$  ants inspire from the previous  $m$  ants and start making their own solution. In each of iterations, each one of the  $m$  ants finds its solution based on the best solution found by the previous  $m$  ants, and the best solution is found for the next  $m$  ants. This is repeated until a certain amount of times where the overall best solution is achieved.

The best solution in each of iterations is chosen according to two factors; *distance* between cluster centers and *sum of the color and physical distances* between each pixel and its cluster center (similarity and compactness of clusters). For the best solution to choose: 1) Distance between cluster centers should be *large* so the clusters are further apart, 2) The sum of the color distances between each pixel and its cluster center should be *small* so that each cluster becomes more similar in color and 3) The sum of the distances between each pixel and its cluster center should be *small* so that each cluster becomes more compact. To achieve the first one, for each clustering performed by ant  $k$  ( $k = 1, \dots, m$ ), we compute the distances between every pair of cluster centers and sort these distances then we pick the minimum distance  $Min(k)$ . Now we compare all these minimums performed by all the ants, and pick the maximum of them [ $Min(k')$ ]. To achieve the second and third, for each clustering performed by ant  $k$  we compute the sum of the distances between each pixel and its cluster center, *and* sort these sum of the distances. Then we pick the maximum *and* compare all these maximums performed by all ants, and pick the minimum of them. The second maximum and third maximum of the solutions are compared in the same way and the minimum is picked. Since every ant has its own solution, solutions are being voted based on their advantages and the solution with the larger vote is selected as the best solution.

After the best solution is found, the pheromone value is updated according to Eq. 3 [10]:

$$\tau_{(i,Xn)} \leftarrow (1 - \rho) \tau_{(i,Xn)} + \sum_i \Delta\tau_{(i,Xn)}. \quad (3)$$

Where  $\rho$  is the evaporation factor ( $0 \leq \rho < 1$ ) which causes the earlier pheromones vanish over the iterations. Therefore as the solution becomes better, the corresponding pheromone have more effect on the next solution rather than the earlier pheromones which correspond to the initial undesired solutions found.  $\Delta\tau_{(i,Xn)}$  in Eq. 3 is the amount of pheromone added to previous pheromone by the succeeded ant, which is obtained from:

$$\Delta\tau_{(i,X_n)} = \begin{cases} \frac{Q * \text{Min}(k')}{\text{AvgColDist}(k', i) * \text{AvgPhysDist}(k', i)} & \text{if } X_n \text{ is a member of cluster } i. \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In Eq. 4,  $Q$  is a positive constant which is related to the quantity of the added pheromone by ants,  $\text{Min}(k')$  is the maximum of the minimum distance between every two cluster centers obtained by ant  $k'$ ,  $\text{AvgColDist}(k', i)$  is the average sum of the color distances and  $\text{AvgPhysDist}(k', i)$  is the average sum of the physical distances, between each pixel and its cluster center  $i$  obtained by ant  $k'$ .  $\text{Min}(k')$  causes the pheromone become bigger when clusters get more apart and hence raise the probability.  $\text{AvgColDist}(k', i)$  and  $\text{AvgPhysDist}(k', i)$  cause the pheromone become bigger when the cluster has more similar pixels and is more compact respectively. In other words the more the  $\text{Min}(k')$  the more apart our clusters are which is desired and the bigger pheromone, and the less the  $\text{AvgColDist}(k', i)$  and  $\text{AvgPhysDist}(k', i)$ , the more similar and compact our clusters are which is desired and the bigger the pheromone.

Next, cluster centers are updated by the cluster centers of the best solution. This is repeated until a certain amount of times where the best of the best solution is achieved. The algorithm is described below:

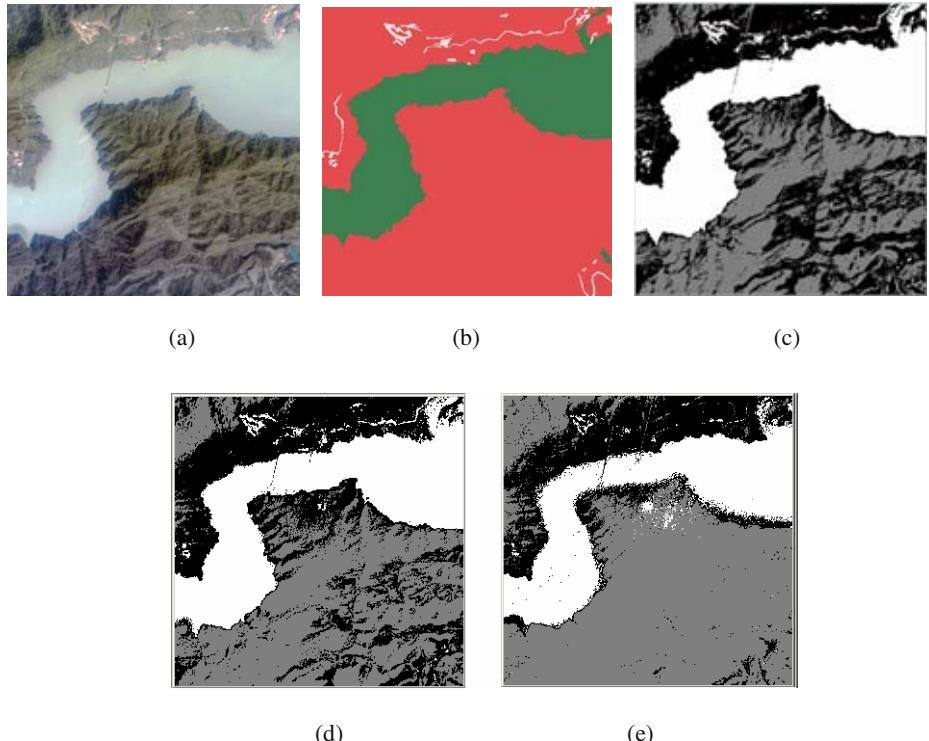
- Step 1:* Initialize pheromone level to 1, and the number of clusters to  $K$  and number of ants to  $m$ .
- Step 2:* Initialize  $m$  sets of  $K$  different random cluster centers to be used by  $m$  ants.
- Step 3:* For each ant, let each pixel  $x$  belong to one cluster with the probability given in Eq. 1.
- Step 4:* Calculate new cluster center; If the new cluster centers converge to the old ones, go to next step otherwise, go to Step 3.
- Step 5:* Save the best solution among the  $m$  solutions found.
- Step 6:* Update the pheromone level on all pixels according to the best solution.
- Step 7:* Update cluster centers by the cluster center values of the best solution.
- Step 8:* If the termination criterion is satisfied go to next step otherwise, go to Step3.
- Step 9:* Output the optimal solution.

## 5 Simulation Results

In the simulation the parameters involved are set as follows. Parameters  $\alpha$ ,  $\beta$  and  $\kappa$  are used to keep the values of  $\tau$  and  $\eta$  in the same order. The values set was  $\alpha = 2$ ,  $\beta = 5$ , and  $\kappa = 1000$ . The parameters  $Q$  controls the added amount of pheromone and  $\rho$  eliminates the influence of the earlier added pheromone and they are set to be,  $Q = 10$  and  $\rho = 0.8$ . The number of ants is chosen to be  $m = 10$ .

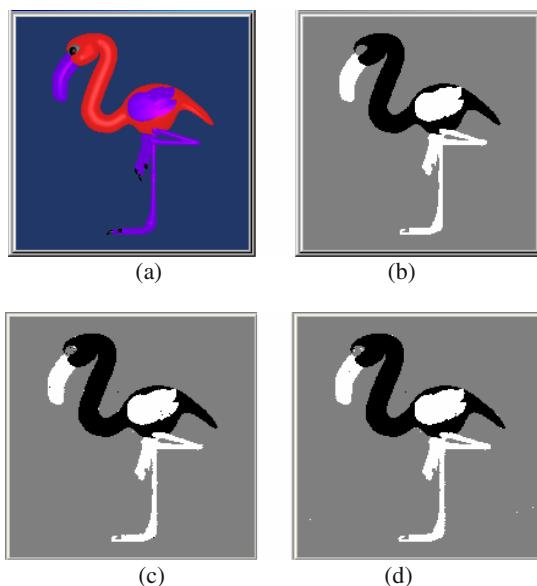
The results for both the K-means and Hybrid ACO-K-means are compared in Figs. 1 and 2 on two different images. In both cases the number of clusters to be found is 3. Different runs of the proposed algorithm are shown in Figs. 1 (d) and (e), which compared to the ground truth data shown in Fig. 1 (b) show a better result with respect to the K-means. Figs. 2 (c) and (d) are the results for another image. As it can

be seen there are some scattered dots over the segmented images. In the first look this can be thought of the instability of the algorithm, but contradiction of this is proved in Figs. 3 and 4 which we will discuss later. These dotted regions do not influence the big picture of the clustering and it is due to the randomness of the algorithm.

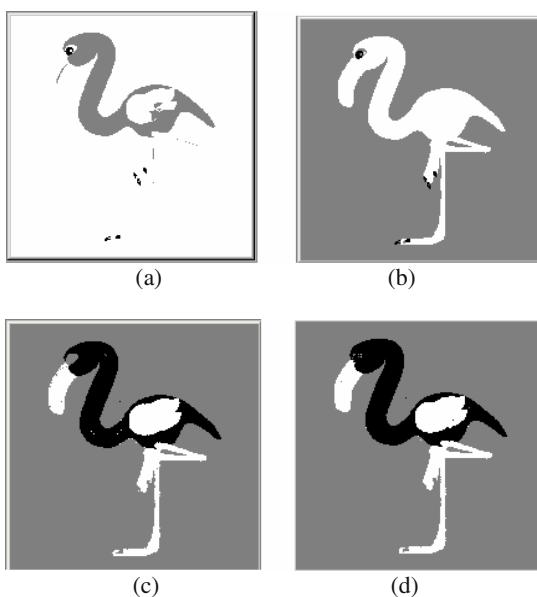


**Fig. 1.** a) An original image, b) The ground truth data, c) K-means segmentation results, d & e) Hybrid ACO-K-means results with  $K = 3$  and different runs

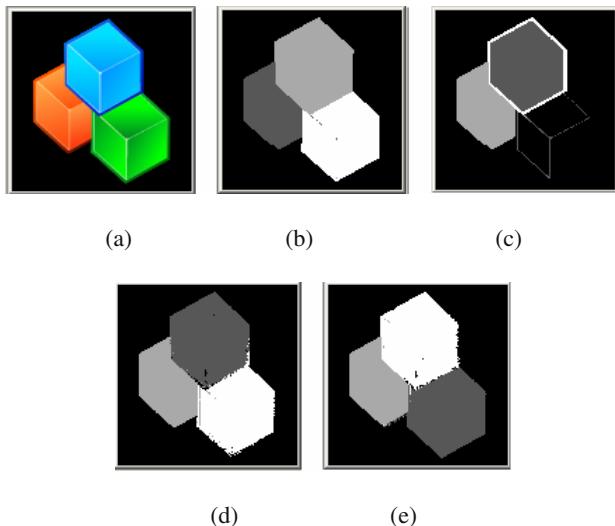
In Fig. 3 improper seeds are forced to show the case in which initial seed values are not properly chosen. Fig. 3 (a) and (b) are the only results of the K-means in this case which shows some information loss, where Figs. 3 (c) and (d) show the results of ACO-K-means which shows that the information is kept. This shows that even when improper seeds are forced to the Hybrid ACO-K-means, since there are several solutions to compete, eventually the improper solution will be omitted and the proper one wins as the answer. This indicates the stability of the Hybrid ACO-K-means algorithm. The stability of the proposed algorithm is further confirmed in Fig. 4 where different runs of both K-means and ACO-K-means are shown. There are some runs of the K-means algorithm which lose the information Fig. 4 (c), due to improper initialization which it is not the case in ACO-K-means.



**Fig. 2.** a) An original image, b) K-means results with  $K = 3$ , c & d) Hybrid ACO-K-means results with  $K = 3$  and different runs



**Fig. 3.** a & b) K-means results with  $K = 3$  and different runs, where the initial seed values are not properly chosen, c & d) Hybrid ACO-K-means results with  $K = 3$  and different runs, where the initial seed values are not properly chosen



**Fig. 4.** a) An original image, b & c) K-means results with  $K = 4$  and different runs, d & e) Hybrid ACO-K-means results with  $K = 4$  and different runs

## 6 Conclusion and Future Work

In this paper a Hybrid ACO-K-means algorithm is proposed. In this algorithm ants are used to obtain solutions based on their own knowledge and the best solution found by previous ants. The advantage of this algorithm over the K-means algorithm is revealed when initial seeds are chosen improperly. Since a solution is obtained for a number of iterations, most of the possible solutions are available to choose from as the best solution. Therefore over a number of iterations, the influence of the improperly chosen initial cluster centers will be diminished. This best solution is marked with the pheromone and eventually becomes denser causing the best result to become as the final result. Therefore this algorithm is less dependent on the initial parameters such as randomly chosen initial seeds hence more stabilized and it is more likely to find the global solution rather than the local.

The proposed ACO-K-means algorithm is not limited to a particular problem. For a given problem the main problem is to define the heuristic information, the pheromone level and the criteria for the best solution. Therefore with this learning mechanism the problem of finding the desired number of clusters with unsupervised information can also be solved by defining the heuristic and pheromone information and desired criterion for the best solution, as an extension of this work.

## Acknowledgments

The authors would like to thank Arash Karbaschi and Sara Arasteh for valuable discussions and helps.

## References

1. Gonzalez, R.C., Woods, R.E., Digital Image Processing, Addison-Wesley (1992)
2. Dorigo, M., Maniezzo, V., Colorni, A., Ant system: optimization by a colony of cooperating agents, In: IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 26, (1996) 29-41
3. Kaewkamnerpong, B., Bentley, P.J., Perceptive Particle Swarm Optimization,
4. <http://www.cs.ucl.ac.uk/staff/B.Kaewkamnerpong/bkpb-icannga05.pdf>
5. Yuqing, P., Xiangdan, H., Shang, L., The K-means clustering Algorithm Based on Density and Ant Colony, In: Proceedings of the International Conference on Neural Networks and Signal Processing, Vol. 1, Nanjing, China, (2003) 457-460
6. Monmarché, N., On data clustering with artificial ants, In: AAAI-99 and GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, A.A. Freitas, ed., Orlando, Florida, (1999) 23-26
7. Monmarché, N., Slimane, M., Venturini, G., AntClass: discovery of clusters in numeric data by an hybridization of ant colony with k-means algorithm, Internal Report no. 213 Laboratoire d'Informatique, E3i, Université de Tours, Tours, France, (1999)
8. Bin, W., Yi, Z., Shaohui, L., Zhongzhi, S., CSIM: A Document Clustering Algorithm Based on Swarm Intelligence, In: Proceedings of the Congress on Evolutionary Computation, Vol. 1, Honolulu, HI, (2002) 477-482
9. Kanade, P.M., Hall, L.O., Fuzzy Ants as a Clustering Concept, In: Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society, Chicago, IL, (2003) 227-232
10. Zheng, H., Zheng, Z., Xiang, Y., The application of ant colony system to image texture classification [texture read texture], In: Proceedings of the 2nd International Conference on Machine Learning and Cybernetics, Vol. 3, Xi'an, China, (2003) 1491-1495
11. Li, Y., Xu, Z., An Ant Colony Optimization Heuristic for Solving Maximum Independent Set Problems, In: Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications, Xi'an, China, (2003) 206-211
12. MacQueen, J.B., Some Methods For Classification and Analysis of Multivariate Observations. In L. M. LeCam and J. Neyman, editors, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability, University of California Press, Berkley, CA, (1967) 281-297
13. Tou, J.T., Gonzalez, R.C., Pattern Recognition Principles, Addison-Wesley, (1974)

# Incremental Locally Linear Embedding Algorithm

Olga Kouropteva\*, Oleg Okun, and Matti Pietikäinen

Machine Vision Group,

Infotech Oulu and Department of Electrical and Information Engineering,  
P.O.Box 4500, FI-90014 University of Oulu, Finland  
`{kouropte, oleg, mkp}@ee.oulu.fi`

**Abstract.** A number of manifold learning algorithms have been recently proposed, including locally linear embedding (LLE). These algorithms not only merely reduce data dimensionality, but also attempt to discover a true low dimensional structure of the data. The common feature of the most of these algorithms is that they operate in a batch or offline mode. Hence, when new data arrive, one needs to rerun these algorithms with the old data augmented by the new data. A solution for this problem is to make a certain algorithm online or incremental so that sequentially coming data will not cause time consuming recalculations. In this paper, we propose an incremental version of LLE and experimentally demonstrate its advantages in terms of topology preservation. Also, compared to the original (batch) LLE, the incremental LLE needs to solve a much smaller optimization problem.

## 1 Introduction

Dimensionality reduction serves to eliminate irrelevant information while preserving the important one. In many cases dimensionality reduction is able to lessen the curse of dimensionality, raise the accuracy rate when there is not enough data (compared to data dimensionality), and improve performance and clustering quality of feature sets. Such improvements are possible since the data lie on or close to a low dimensional manifold, which is embedded in a high dimensional space. Consider, for example, a set of grayscale facial images of resolution  $m \times n$  taken under different views with fixed illuminating conditions. Each of the images can be represented with brightness pixel values as a point in  $\mathbb{R}^{mn}$  space. However, the intrinsic dimensionality of the manifold formed by these facial images is equal to the degree of freedom of the camera. Therefore, it is much smaller than the image size.

To obtain a relevant low dimensional representation of high dimensional data, several manifold learning algorithms [1, 2, 3, 4, 5] have been recently proposed.

---

\* Olga Kouropteva is grateful to the Infotech Oulu Graduate School and the Nokia Foundation.

Manifold learning is a perfect tool for data mining that discovers structure of large high dimensional datasets and, hence, provides better understanding of the data. Nevertheless, most of the manifold learning algorithms operate in a batch mode, hence they are unsuitable for sequentially coming data. In other words, when new data arrive, one needs to rerun the entire algorithm with the original data augmented by the new samples.

Recently, an incremental version of one of the manifold learning algorithms called isometric feature mapping (Isomap) [5] has been proposed in [6], where the authors suggested that it can be extended to the online versions of other manifold learning algorithms. Unfortunately, LLE does not belong to this group of algorithms. First of all, as remarked in [7], it is much more challenging to make LLE incremental than other manifold learning algorithms. Secondly, LLE aims at bottom eigenvectors and eigenvalues rather than at top ones. It is well known that ill-conditioning of eigenvalues and eigenvectors frequently occurs in the former case and it is impossible in the latter case. Ill-conditioning means that eigenvalues or/and eigenvectors are susceptible to small changes of a matrix for which they are computed. As a result, problems one faces with when making LLE incremental are more formidable than those for other manifold learning algorithms searching for the top eigenvalues/eigenvectors. This leads to the necessity of inventing another generalization method for LLE. In this paper we propose such a method, called incremental LLE, which is based on the intrinsic properties of LLE. Additionally, we compare the incremental LLE with two previously proposed non-parametric generalization procedures for LLE [8, 9]. Promising and encouraging results are demonstrated in the experimental part.

The paper is organized as follows. A brief description of the LLE algorithm is given in Section 2. Section 3 presents all incremental versions of LLE, including the new one. They are compared on several datasets and the obtained results are discussed in Section 4. Section 5 concludes the paper.

## 2 Locally Lineal Embedding Algorithms

As input, LLE requires  $D$  dimensional points (one point per pattern) assembled in a matrix  $\mathbf{X}$ :  $\mathbf{X} = \{x_1, x_2, \dots, x_N\}, x_i \in \mathbb{R}^D, i = 1, \dots, N$ . As output, it produces  $d$  dimensional points ( $d << D$ ) assembled in a matrix  $\mathbf{Y}$ :  $\mathbf{Y} = \{y_1, y_2, \dots, y_N\}, y_i \in \mathbb{R}^d, i = 1, \dots, N$ . The  $i^{th}$  column of  $\mathbf{X}$  corresponds to the  $i^{th}$  column of  $\mathbf{Y}$ .

The LLE algorithm consists of three steps [4]:

1. For each  $x_i \in \mathbb{R}^D, i = 1, \dots, N$  find its  $K$  nearest neighbors:  $x_i^1, x_i^2, \dots, x_i^K$  by using the Euclidean distance as a similarity measure. A technique for selecting the optimal parameter  $K$  for LLE was proposed in [10].
2. Compute weights that best reconstruct each  $x_i$  from its  $K$  nearest neighbors,  $x_i^1, x_i^2, \dots, x_i^K$ , by minimizing the following cost function:

$$\varepsilon(\mathbf{W}) = \sum_{i=1}^N \| x_i - \sum_{j=1}^N w_{ij} x_j \|^2 , \quad (1)$$

subject to constraints:  $w_{ij} = 0$ , if  $x_j \notin \{x_i^1, x_i^2, \dots, x_i^K\}$ , and  $\sum_{j=1}^N w_{ij} = 1$ . The first (sparseness) constraint assures that each point  $x_i$  is reconstructed only from its neighbors, while the second condition enforces translation invariance of  $x_i$  and its neighbors. Moreover, as follows from Eq. 1, the constrained weights are invariant to rotation and rescaling, but not to local affine transformations, such as shears.

3. Set  $d$  - the number of dimensions in the embedding space (see [11] for automatic computing of  $d$ ). Fix the reconstruction weights  $w_{ij}$  and compute low-dimensional embeddings by minimizing the embedding cost function:

$$\delta(\mathbf{Y}) = \sum_{i=1}^N \|y_i - \sum_{j=1}^N w_{ij} y_j\|^2, \quad (2)$$

subject to  $\frac{1}{N} \sum_{i=1}^N y_i y_i^T = I$  (normalized unit covariance) and  $\sum_{i=1}^N y_i = 0$  (translation-invariant embedding), which provide a unique solution.

Finding a low dimensional embedding under these constraints is equivalent to computing the bottom  $d+1$  eigenvectors associated with the  $d+1$  smallest eigenvalues of a sparse and symmetric matrix  $\mathbf{M} = (I - \mathbf{W})^T(I - \mathbf{W})$ . The first eigenvector (composed of 1's) whose eigenvalue is close to zero is excluded. The remaining  $d$  eigenvectors yield the final embedding  $\mathbf{Y}$ .

### 3 Incremental LLE

LLE operates in a batch or offline mode, that is, it obtains a low-dimensional representation for a certain number of high-dimensional data points to which the algorithm is applied. When new data points arrive, one needs to completely rerun the original LLE for the previously seen dataset augmented by the new data points. In other words, the original LLE lacks generalization to new data. This makes the algorithm to be less attractive especially for large datasets of high dimensionality in a dynamic environment, where a complete rerun of LLE becomes prohibitively expensive.

In [12], an attempt was made to adapt LLE to a situation when the data come incrementally point-by-point. Two simple techniques were proposed, in which the adaptation to a new point can be done either by updating the weight matrix  $\mathbf{W}$  or the cost matrix  $\mathbf{M}$ , respectively. In both these cases, an expensive eigenvector calculation is required for each query. There are two ways to lower the complexity of the LLE generalization: 1) to derive and use a transformation between the original and projected data, and 2) to solve an incremental eigenvalue problem. In this section, we describe two known generalization algorithms belonging to the former case, and propose the incremental version of LLE that uses the latter approach.

Suppose we are given already processed data  $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$ , corresponding projected points  $\mathbf{Y} = \{y_1, y_2, \dots, y_N\}$ , and a new point  $x_{N+1} \in \mathbb{R}^D$ , which is sampled from the same data manifold as  $\mathbf{X}$ . We are asked to find a new embedding point  $y_{N+1} \in \mathbb{R}^d$  corresponding to the point  $x_{N+1}$ .

### 3.1 Linear Generalization

In order to obtain the new embedded coordinates, the LLE intuition is used, i.e. any nonlinear manifold can be considered as locally linear. This linearity is used to build a linear relation between high and low dimensional points belonging to a particular neighborhood of the data. There are two possibilities of linear generalization [8, 9]:

1. Let us put the  $K$  nearest neighbors of  $x_{N+1}$  and the corresponding embedded points into the matrices:  $\mathbf{X}^{N+1} = \{x_{N+1}^1, x_{N+1}^2, \dots, x_{N+1}^K\}$  and  $\mathbf{Y}^{N+1} = \{y_{N+1}^1, y_{N+1}^2, \dots, y_{N+1}^K\}$ . By taking into consideration the assumption that the manifold is locally linear, the following equation is approximately true:  $\mathbf{Y}^{N+1} = \mathbf{Z}\mathbf{X}^{N+1}$ , where  $\mathbf{Z}$  is an unknown linear transformation matrix of size  $d \times D$ , which can be straightforwardly determined as  $\mathbf{Z} = \mathbf{Y}^{N+1}(\mathbf{X}^{N+1})^{-1}$ . Because  $\mathbf{X}^{N+1}$  is the neighborhood of  $x_{N+1}$  and LLE preserves local structures, i.e. points close in the original space remain close in the embedded space, the new projection can be found as  $y_{N+1} = \mathbf{Z}x_{N+1}$ . Here we multiply the new input by the found transformation matrix, since the underlying manifold must be well sampled and, hence, the neighboring points give sufficient information about the new point [8].
2. To find  $y_{N+1}$ , first, the  $K$  nearest neighbors of  $x_{N+1}$  are detected among points in the high dimensional space:  $x_i \in \mathbf{X}, i = 1, \dots, N$ . Then, the linear weights,  $w_{N+1}$ , that best reconstruct  $x_{N+1}$  from its neighbors, are computed by using Eq. 1 with the sum-to-one constraint:  $\sum_{j=1} w_{N+1j} = 1$ . Finally, the new output  $y_{N+1}$  is found:  $y_{N+1} = \sum_{j=1} w_{N+1j}y_j$ , where the sum is over the  $y_j$ 's corresponding to the  $K$  nearest neighbors of  $x_{N+1}$  [9].

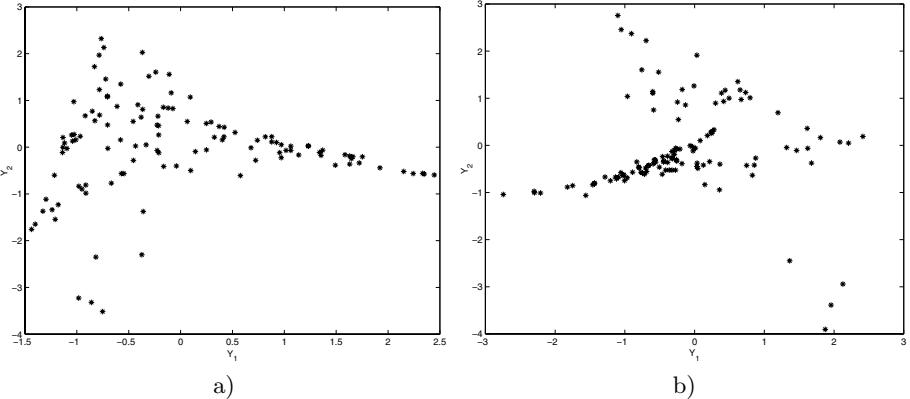
### 3.2 Incremental LLE

In order to construct embedding LLE searches for the smallest eigenvalues and corresponding eigenvectors of the Hermitian matrix of size  $N \times N$ . Hence, one has to deal with ill-conditioned eigenproblem [13]. Ill-conditioning means that eigenvalues and/or eigenvectors of a particular matrix are very sensitive to small perturbations of the matrix. For example, changing of the matrix in norm by at most  $\epsilon$  can change any eigenvalue by at most  $\epsilon$ , i.e. computing  $\lambda_i = 10^{-5}$  to within plus or minus  $\epsilon = 10^{-4}$  means that no leading digits of the computed  $\lambda_i$  may be correct.

Eigenvectors and eigenspaces they span are ill-conditioned if small change of the matrix, e.g. changing

$$A_0 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & \epsilon \\ 0 & \epsilon & 1 \end{pmatrix} \text{ to } A_1 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 + \epsilon \end{pmatrix}$$

rotates the two eigenvectors corresponding to the two eigenvalues near one by  $\pi/4$ , no matter how small  $\epsilon$  is. Thus they are very sensitive to small changes. Note, that if the eigenvalues are ill-conditioned, then the corresponding eigenvectors are also ill-conditioned; the opposite is not always true.



**Fig. 1.** 2D spaces formed by two bottom eigenvectors of a) the initial matrix, and b) the modified matrix. The difference between norms of these matrices is equal to  $-2.36 \cdot 10^{-33}$

Fig. 1 demonstrates the consequence of ill-conditioning. First, we find two bottom eigenvectors of a particular matrix. These eigenvectors form 2D space in Fig. 1 (a). Then we change the elements of the matrix by adding or subtracting very small values. Finally, we compute two bottom eigenvectors of the modified matrix, which are shown in Fig. 1 (b). One can see that these plots dramatically differ from each other, while the difference between norms of the initial and modified matrices is equal to  $-2.36 \cdot 10^{-33}$ .

When a new data point arrives, the main goal of the incremental LLE is to compute the new cost matrix  $\mathbf{M}_{new}$  to be exactly the same as if it would be computed by LLE applied to the old data augmented by the new data point. This can be done by applying the following operations: first, distances between points, which either belong to the  $K$  nearest neighbors of the new point or contain the new point as one of their  $K$  nearest neighbors, are recalculated. Then the weights for the points whose distances have been changed are updated by solving Eq. 1 and the new matrix  $\mathbf{M}_{new}$  of size  $(N+1) \times (N+1)$  is calculated by using these weights.

The classical eigenproblem is defined as the solution of the equation  $\mathbf{M}y_i^T = \lambda_i y_i^T$  or in matrix form  $\mathbf{M}\mathbf{Y}^T = diag\{\lambda_1, \lambda_2, \dots, \lambda_d\}\mathbf{Y}^T$ . Since typical eigenvectors are orthogonal, we can rewrite the eigenproblem:  $\mathbf{Y}\mathbf{M}\mathbf{Y}^T = diag\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ . Without loss of generality, we assume that the eigenvalues of the new cost matrix,  $\mathbf{M}_{new}$  are the same as for the cost matrix computed for  $N$  points. This can be done since the eigenvalues, we are dealing with, are very close to zero, usually they are of order  $10^{-p}$ , where  $p$  is large enough (practically about 10). Therefore we can write  $\mathbf{Y}_{new}\mathbf{M}_{new}\mathbf{Y}_{new}^T = diag\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ , where  $\{\lambda_i\}, i = 1, \dots, d$  are the smallest eigenvalues of the cost matrix computed for  $N$  points. The new coordinates are obtained by solving  $d \times d$  minimization problem:

$$\min_{\mathbf{Y}_{new}} (\mathbf{Y}_{new}\mathbf{M}_{new}\mathbf{Y}_{new}^T - diag\{\lambda_1, \lambda_2, \dots, \lambda_d\}). \quad (3)$$

The LLE constraints imposed on the embedding coordinates should be kept. Thus, the  $N \times N$  problem of the third LLE step was reduced to the  $d \times d$  problem, where  $d \ll N$ . Since  $d$  is usually very small, say 10 or so, the minimization is not time consuming and can be done for every arriving point.

## 4 Experiments

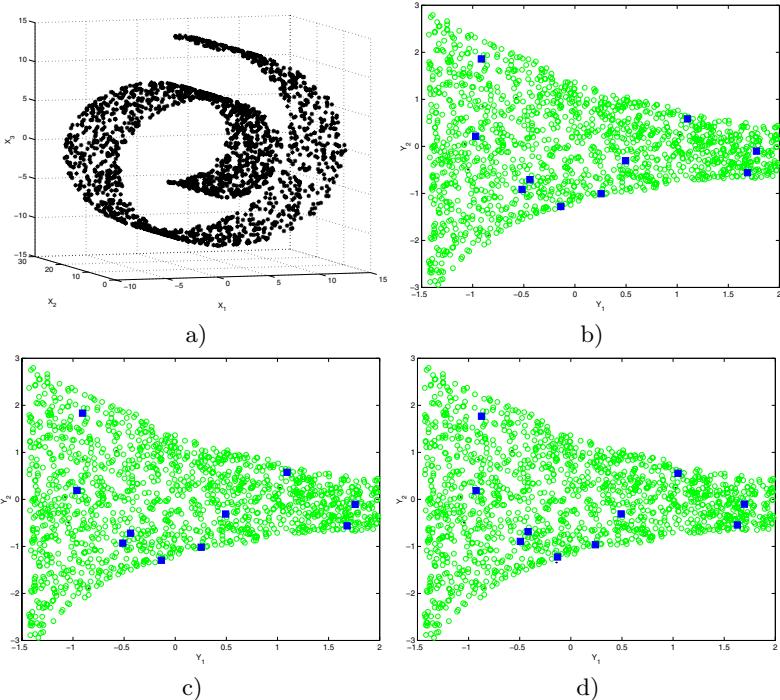
In the experiments we applied the three LLE generalization algorithms described in the previous section to the datasets represented in Table 1. The Olga's and Oleg's faces datasets were taken by a Sony DFW-X700 digital camera under fixed illumination conditions. Each sequence consists of images showing one person slowly rotating the head from left to right, while trying to fix a chin at the same level in order to obtain one degree of freedom: angle of rotation. In spite of the fact that initial conditions for capturing these datasets were the same, the Oleg's faces data is uniformly distributed, while Olga's faces data is not. This is due to the velocity of head rotation: Olga rotated her head from the left to frontal view slower than from frontal view to the right; therefore, there are more frames for the former case than for the latter one. Hence, we consider Olga's faces dataset to be non-uniform. The description of other datasets can be found from the corresponding references.

**Table 1.** Datasets used in the experiments

Data	N points	Dimensionality	Features
Swissroll [9]	2000	3	Coordinates
S-curve [9]	2000	3	Coordinates
Wine [14]	178	13	Chemical measurements
Fray faces [15]	1965	560	Grayscale pixels values
MNIST digits(3&8) [16]	1984	784	Grayscale pixels values
Coil-20 [17]	1440	4096	Grayscale pixels values
Oleg's faces	1130	6300	Grayscale pixels values
Olga's faces	1200	6300	Grayscale pixels values

All datasets were divided into training (70%) and test (30%) sets. The training sets were projected by LLE to two dimensional spaces and the test sets were mapped to the corresponding space by the generalization algorithms described in Section 3.

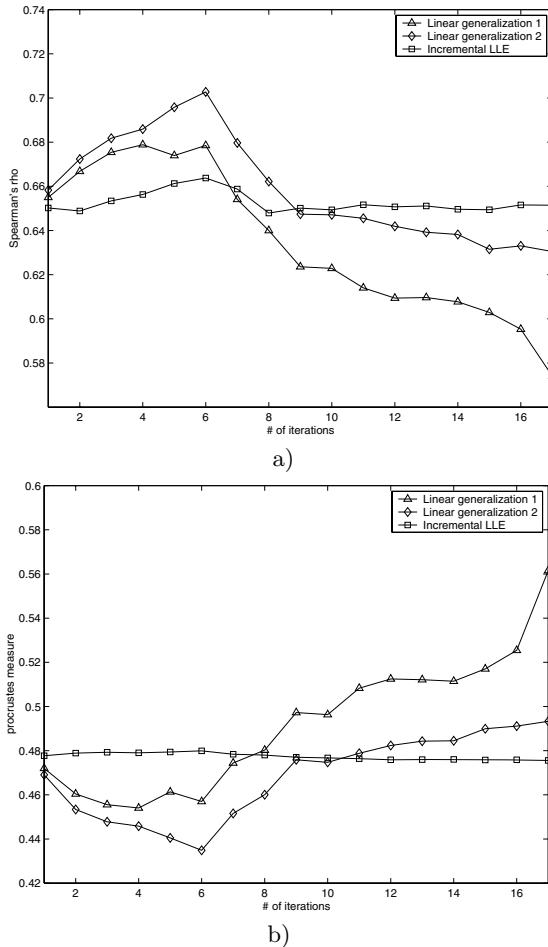
The first experiment is done with the swissroll dataset (Fig. 2 (a)). At the beginning, we project the initial data containing 1,400 points by the conventional LLE algorithm ( $K = 15$ ), and then we add the test data points in a random order. In Fig. 2 (b, c, d) results of generalizing ten new data points are shown. As one can see the projections of the new points are visually almost the same for all methods.



**Fig. 2.** LLE generalization on the swissroll dataset. (a) The original 3D data points sampled from the corresponding data manifold. The circles ( $\circ$ ) depict the target coordinates, i.e. those, which are obtained by applying the conventional LLE to the pooled dataset, including both old and new data points. The dots ( $\cdot$ ) show the estimated coordinates for b) linear generalization 1; c) linear generalization 2; d) incremental LLE. The filled squares ( $\square$ ) correspond to the projections of the new points

That is why, in order to quantitatively compare the generalization methods, we calculate two characteristics, namely, Spearman's rho and procrustes measure. The Spearman's rho estimates the correlation of rank order data, i.e. how well the corresponding low dimensional projection preserves the order of the pairwise distances between the high dimensional data points converted to ranks. The best value of Spearman's rho is equal to one. In its turn, the procrustes measure determines how well a linear transformation (translation, reflection, orthogonal rotation, and scaling) of the points in the projected space conforms to the points in the corresponding high dimensional space. The smaller the value of procrustes measure, the better fitting is obtained. Both Spearman's rho and procrustes measure are commonly used for estimating topology preservation when doing dimensionality reduction.

In another experiment, 119 wine data training points are projected by the conventional LLE ( $K = 15$ ) and the other 51 test points are added during 17 iterations ( $n = 3$  points per time) by three generalization methods: linear



**Fig. 3.** Spearman's rho (a) and procrustes measure (b) for the wine dataset

generalization 1 (LG1), linear generalization 2 (LG2), and incremental LLE (ILLE). In Fig.3 plots show the Spearman's rho and procrustes measure estimated after each iteration. One can see that the incremental LLE performs better than other generalization procedures when the number of new samples increases. But this is not always the case. In order to make the final conclusion, we estimate the Spearman's rho and procrustes measure after each iteration for all datasets and count the number of iterations for which a particular method outperforms others in terms of the Spearman's rho and procrustes measure. The number of iterations, number of points per iteration, and the results for all datasets are listed in Table 2. The largest resulting values are underlined.

By looking at Table 2, a number of interesting observations can be done.

**Table 2.** Spearman's rho ( $\rho_{Sp}$ ) and procrustes measure ( $Procr$ ) for the datasets

Data	N of iterations	N of points per iteration		LG1	LG2	ILLE
S-curve	60	10	$\rho_{Sp}$	<u>47</u>	12	1
			$Procr$	<u>43</u>	16	1
Wine	17	3	$\rho_{Sp}$	0	8	<u>9</u>
			$Procr$	0	<u>10</u>	7
Fray faces	28	21	$\rho_{Sp}$	1	8	<u>19</u>
			$Procr$	0	14	14
MNIST digits(3&8)	22	27	$\rho_{Sp}$	0	0	<u>22</u>
			$Procr$	0	<u>22</u>	0
Coil-20	27	16	$\rho_{Sp}$	0	<u>27</u>	0
			$Procr$	1	5	<u>21</u>
Oleg's faces	33	10	$\rho_{Sp}$	2	<u>31</u>	0
			$Procr$	16	<u>17</u>	0
Olga's faces	36	10	$\rho_{Sp}$	0	6	<u>30</u>
			$Procr$	14	5	<u>17</u>

- When manifold is well and evenly sampled, and the relationships between original and embedded datasets are close to be locally linear as in case of S-curve and Oleg's faces, LG1 and especially LG2 are sufficient for successful generalization of test points, and this fact is confirmed by both Spearman's rho and procrustes measure.
- However, if the data manifold is non-uniformly sampled and the relationships are locally nonlinear as in case of Olga's and Fray's faces, ILLE emerges as a clear winner, since it does not rely on linear relationships. This is again supported by both Spearman's rho and procrustes measure.

## 5 Conclusion

LLE belongs to the class of manifold learning algorithms, which reduce dimensionality by learning structure of data manifolds. The deficiency of LLE is that if new inputs arrive, one needs to rerun the algorithm for the pool of the old and new data. The main difficulty of making LLE incremental is that it obtains embeddings by searching for the smallest eigenvectors, which are ill-conditioned. In this paper we proposed a new method for LLE generalization, called incremental LLE, and compared it with linear generalization methods. The results demonstrate that ILLE is a powerful generalization method for data whose distribution is not uniform and the local linearity constraints do not hold. In contrast, the linear generalization methods deal perfectly with well-sampled manifolds of artificially generated data but may have problems with real-world datasets.

One of the directions of the future work is to compare the classification performance of the linear generalization algorithms and ILLE on different datasets.

## References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Technical Report TR-2002-01, University of Chicago, Department of Computer Science (2002)
2. DeCoste, D.: Visualizing Mercer kernel feature spaces via kernelized locally-linear embeddings. In: Proc. of the 8th Int. Conf. on Neural Information Processing, Shanghai, China. (2001)
3. Donoho, D., Grimes, G.: Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. Proc. of National Academy of Sciences **100** (2003) 5591–5596
4. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326
5. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323
6. Law, M., Zhang, N., Jain, A.: Nonlinear manifold learning for data stream. In Berry, M., Dayal, U., Kamath, C., Skillicorn, D., eds.: Proc. of the 4th SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA. (2004) 33–44
7. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M.: Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16, Cambridge, MA, MIT Press (2004)
8. Kouropteva, O., Okun, O., Hadid, A., Soriano, M., Marcos, S., Pietikäinen, M.: Beyond locally linear embedding algorithm. Technical Report MVG-01-2002, University of Oulu (2002)
9. Saul, L., Roweis, S.: Think globally, fit locally: unsupervised learning of nonlinear manifolds. Journal of Machine Learning Research **4** (2003) 119–155
10. Kouropteva, O., Okun, O., Pietikäinen, M.: Selection of the optimal parameter value for the locally linear embedding algorithm. In: Proc. of 2002 Int. Conf. on Fuzzy Systems and Knowledge Discovery, Singapore. (2002) 359–363
11. de Ridder, D., Duin, R.: Locally linear embedding for classification. Technical Report PH-2002-01, Delft University of Technology (2002)
12. Kouropteva, O.: Unsupervised learning with locally linear embedding algorithm: an experimental study. Master's thesis, University of Joensuu, Finland (2001)
13. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H.: Templates for the solution of algebraic eigenvalue problems. SIAM, Philadelphia (2000)
14. (<http://www.ics.uci.edu/~mlearn/>)
15. (<http://www.cs.toronto.edu/~roweis/data.html>)
16. (<http://yann.lecun.com/exdb/mnist/index.html>)
17. ([http://www1.cs.columbia.edu/CAVE/research/softlib/coil\\_20.html](http://www1.cs.columbia.edu/CAVE/research/softlib/coil_20.html))

# On Aligning Sets of Points Reconstructed from Uncalibrated Affine Cameras

A. Bartoli<sup>1</sup>, H. Martinsson<sup>2</sup>, F. Gaspard<sup>2</sup>, and J.-M. Lavest<sup>1</sup>

<sup>1</sup> LASMEA (CNRS / UBP) – Clermont-Ferrand, France

<sup>2</sup> CEA LIST – Gif sur Yvette, France

[Adrien.Bartoli@gmail.com](mailto:Adrien.Bartoli@gmail.com) – [Hanna.Martinsson@cea.fr](mailto:Hanna.Martinsson@cea.fr)

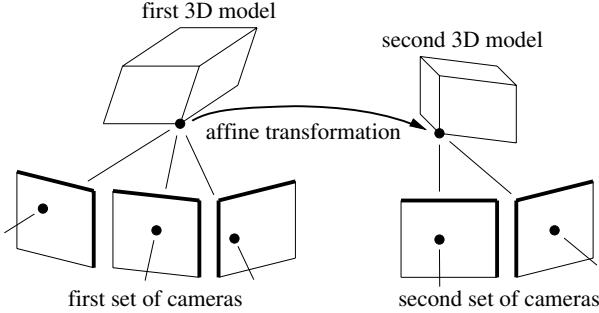
**Abstract.** The reconstruction of rigid scenes from multiple images is a central topic in computer vision. Approaches merging partial 3D models in a hierarchical manner have proven the most effective to deal with large image sequences. One of the key building blocks of these hierarchical approaches is the alignment of two partial 3D models by computing a 3D transformation. This problem has been well-studied for the cases of 3D models obtained with calibrated or uncalibrated pinhole cameras.

We tackle the problem of aligning 3D models – sets of 3D points – obtained using uncalibrated affine cameras. This requires to estimate 3D affine transformations between the 3D models. We propose a factorization-based algorithm estimating simultaneously the aligning transformations and corrected points, exactly matching the estimated transformations, such that the reprojection error over all cameras is minimized.

We experimentally compare our algorithm to other methods using simulated and real data.

## 1 Introduction

Three dimensional reconstruction from multiple images of a rigid scene, often dubbed Structure-From-Motion, is one of the most studied problems in computer vision. The difficulties come from the fact that, using only feature correspondences, both the 3D structure of the scene and the cameras have to be computed. Most approaches rely on an initialisation phase optionally followed by self-calibration and bundle adjustment. Existing initialisation algorithms can be divided into three families, namely  $\dots$ ,  $\dots$  and  $\dots$  processes. Hierarchical processes [1] have proven the most successful for large image sequences. Indeed, batch processes such as the factorization algorithms [2] which reconstruct all features and cameras in a single computation step, do not easily handle occlusions, while sequential processes reconstruct each view on turn, may typically suffer from accumulation of the errors. Hierarchical processes merge partial 3D models obtained from sub-sequences, which allows to distribute the error over the sequence, and efficiently handle open and closed sequences. A key step of hierarchical processes is the fusion or the  $\dots$  of partial 3D models, by  $\dots$ . This problem has been extensively studied in the projective and metric cases.



**Fig. 1.** This paper deals with the estimation of 3D affine transformations between two (or more) affine reconstructions obtained from uncalibrated affine cameras

We focus on the affine camera model, which is a reasonable approximation to the perspective camera model when the depth of the observed scene is small compared to the viewing distance. Partial 3D models obtained from sub-sequences, multiple subsets of cameras, are related by 3D affine transformations. We deal with the computation of such transformations from point correspondences, as illustrated on figure 1. We propose a Maximum Likelihood Estimator based on factorizing modified 3D point coordinates. We compute a 3D affine transformation and a set of 3D point correspondences which perfectly match, such that. The method can be embedded in a robust RANSAC-like [3] framework to deal with data sets containing outliers. It is intended to fit in hierarchical affine Structure-From-Motion processes of which the basic reconstruction block is, the affine factorization [2]. Our method, based on the new concept of, requires a single Singular Value Decomposition (SVD) in the occlusion-free case.

This paper is organized as follow. We give our notation and preliminaries in §2. In §3, we review the factorization approach to uncalibrated affine Structure-From-Motion. Our alignment method is described in §4, while other methods are summarized in §5. Experimental results are reported in §6. Our conclusions are given in §7.

## 2 Notation and Preliminaries

Vectors are typeset using bold fonts,  $\mathbf{x}$ , and matrices using sans-serif, calligraphic and greek fonts,  $A$ ,  $Q$  and  $\Lambda$ . We do not use homogeneous coordinates, image point coordinates are 2-vectors:  $\mathbf{x}^T = (x \ y)$ , where  $T$  is transposition. The different sets of cameras are indicated with primes,  $P_1$ ,  $P'_1$  and  $P''_1$  are the first cameras of the three first camera sets. Index  $i = 1 \dots n$  is used for the cameras of a camera set and index  $j = 1 \dots m$  is used for the 3D points. The identity matrix is denoted  $I$  and the zero matrix and vector by  $0$  and  $\mathbf{0}$ . The Frobenius or  $L_2$  norm of a matrix  $A$  or a vector  $\mathbf{x}$  are respectively

denoted  $\|\mathbf{A}\|$  and  $\|\mathbf{x}\|$ . The mean vector of a set of vectors, say  $\{\mathbf{Q}_j\}$ , is denoted  $\bar{\mathbf{Q}}$ . The Moore-Penrose pseudoinverse of matrix  $\mathbf{A}$  is denoted  $\mathbf{A}^\dagger$ .

Let  $\mathbf{Q}_j$  be a 3-vector and  $\mathbf{x}_{ij}$  a 2-vector representing respectively a 3D and an image point. The uncalibrated affine camera is modeled by a  $(2 \times 3)$  matrix  $\mathbf{P}_i$  and a  $(2 \times 1)$  translation vector  $\mathbf{t}_i$ , giving the projection equation:

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{Q}_j + \mathbf{t}_i. \quad (1)$$

Calligraphic fonts are used for the measurement matrices:  $\mathcal{X}_{(2n \times m)}$  is made with measured point coordinates  $\mathbf{x}_{ij}$  and  $\mathcal{X} = (\mathcal{Y}_1 \cdots \mathcal{Y}_m)$ , where  $\mathcal{Y}_j$  contains all the measured image coordinates for the  $j$ -th point. The so-called  $(2n \times 3)$  ‘joint projection’ and  $(3 \times m)$  ‘joint structure’ matrices are defined by  $\mathcal{P}^T = (\mathbf{P}_1^T \cdots \mathbf{P}_n^T)$  and  $\mathcal{Q} = (\mathbf{Q}_1 \cdots \mathbf{Q}_m)$ . We assume that the noise on image point positions is i.i.d., centred Gaussian. Under these hypotheses minimizing the reprojection error yields Maximum Likelihood Estimates.

### 3 Structure-From-Motion Using Factorization

Given a set of point matches  $\{\mathbf{x}_{ij}\}$ , the factorization algorithm is employed to recover all cameras  $\{\hat{\mathbf{P}}_i, \hat{\mathbf{t}}_i\}$  and 3D points  $\{\hat{\mathbf{Q}}_j\}$  at once [2]. Under the aforementioned hypotheses on the noise distribution, this algorithm computes Maximum Likelihood Estimates by minimizing the reprojection error:

$$\mathcal{R}^2(\mathcal{P}, \mathcal{Q}, \{\mathbf{t}_i\}) = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m d^2(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{Q}_j + \mathbf{t}_i), \quad (2)$$

where  $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$  is the Euclidean distance between  $\mathbf{x}$  and  $\mathbf{y}$ . The problem is thus formulated as  $\min_{\hat{\mathcal{P}}, \hat{\mathcal{Q}}, \{\hat{\mathbf{t}}_i\}} \mathcal{R}^2(\hat{\mathcal{P}}, \hat{\mathcal{Q}}, \{\hat{\mathbf{t}}_i\})$ .

Given the uncalibrated affine projection (1), the first step of the algorithm is to compute the translation  $\hat{\mathbf{t}}_i$  of each camera in order to cancel it out from the projection equation. This is achieved by nullifying the partial derivatives of the reprojection error (2) with respect to  $\hat{\mathbf{t}}_i$ :  $\frac{\partial \mathcal{R}^2}{\partial \hat{\mathbf{t}}_i} = 0$ . A short calculation shows that if we fix the arbitrary centroid of the 3D points to the origin, then  $\hat{\mathbf{t}}_i = \bar{\mathbf{x}}_i$ . Each set of image points is therefore centred on its centroid,  $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \bar{\mathbf{x}}_i$ , to obtain  $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{Q}_j$ . Henceforth, we work in centred coordinates which allows to write the  $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{Q}_j$  from (1).

We rewrite  $\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m d^2(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{Q}_j)$  the reprojection error. The problem is thus reformulated as  $\min_{\hat{\mathcal{P}}, \hat{\mathcal{Q}}} \mathcal{R}^2(\hat{\mathcal{P}}, \hat{\mathcal{Q}})$ . The reprojection error can be rewritten by gathering the terms using the measurement, the ‘joint projection’ and the ‘joint structure’ matrices as  $\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) \propto \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2$ , and the problem is solved by computing the Singular Value Decomposition (SVD) [4] of matrix  $\mathcal{X}$ :  $\mathcal{X}_{2n \times m} = \mathbf{U}_{2n \times m} \Sigma_{m \times m} \mathbf{V}_{m \times m}^T$ , where  $\mathbf{U}$  and

$\mathbf{V}$  are orthonormal matrices and  $\Sigma$  is diagonal and contains the singular values of  $\mathcal{X}$ . Let  $\Sigma = \Sigma_u \Sigma_v$  be any decomposition of matrix  $\Sigma$ ,  $\Sigma_u = \Sigma_v = \sqrt{\Sigma}$ . The motion and structure are obtained by, loosely speaking, ‘truncating’ the decomposition or nullifying all but the 3 first singular values, which leads to  $\mathcal{P} = \psi(\mathbf{U}\Sigma_u)$  and  $\mathcal{Q} = \psi^\top(\mathbf{V}\Sigma_v^\top)$ , where  $\psi(W)$  returns the matrix formed with the 3 leading columns of matrix  $W$ . Note that the alternative solution  $\mathcal{P} = \psi(\mathbf{U})$  and  $\mathcal{Q} = \psi^\top(\mathbf{V}\Sigma)$  has the property  $\mathcal{P}^\top\mathcal{P} = \mathbf{I}$  which is useful for our alignment method, see §4. The 3D model is obtained only up to a global affine transformation. Indeed, let  $\mathbf{B}$  be a  $(3 \times 3)$  invertible matrix:  $\tilde{\mathcal{P}} = \hat{\mathcal{P}}\mathbf{B}$  and  $\tilde{\mathcal{Q}} = \mathbf{B}^{-1}\hat{\mathcal{Q}}$  give the same reprojection error as  $\mathcal{P}$  and  $\mathcal{Q}$  since  $\mathcal{R}^2(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}) = \|\mathcal{X} - \tilde{\mathcal{P}}\tilde{\mathcal{Q}}\| = \|\mathcal{X} - \hat{\mathcal{P}}\mathbf{B}\mathbf{B}^{-1}\hat{\mathcal{Q}}\|^2 = \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 = \mathcal{R}^2(\mathcal{P}, \mathcal{Q})$ . As presented above, the factorization algorithm do not handle occlusions. Though some algorithms have been proposed, see [5], they are not appropriate for Structure-From-Motion from large image sequences.

## 4 Alignment of 3D Affine Reconstructions

We formally state the alignment problem in the two camera set case and present our algorithm, dubbed ‘FACTMLE’. Its extension to the multiple camera set case is trivial and is omitted.

### 4.1 Problem Statement

Consider two sets of cameras  $\{(\mathbf{P}_i, \mathbf{t}_i)\}_{i=1}^n$  and  $\{(\mathbf{P}'_i, \mathbf{t}'_i)\}_{i=1}^{n'}$  and associated structures<sup>1</sup>  $\{\mathbf{Q}_j \leftrightarrow \mathbf{Q}'_j\}_{j=1}^m$  obtained by reconstructing a rigid scene using the above-described factorization algorithm. The reprojection error over these two sets is given by:

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2nm} (\mathcal{R}^2(\mathcal{P}, \mathcal{Q}, \{\mathbf{t}_i\}) + \mathcal{R}'^2(\mathcal{P}', \mathcal{Q}', \{\mathbf{t}'_i\})). \quad (3)$$

Let  $(\hat{\mathbf{A}}, \hat{\mathbf{t}})$  represent the aligning  $(3 \times 3)$  affine transformation. The Maximum Likelihood Estimator is formulated by:

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}'}} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}'}) \quad \text{s.t.} \quad \hat{\mathbf{Q}}'_j = \hat{\mathbf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}. \quad (4)$$

### 4.2 A Factorization-Based Algorithm

Our method to solve problem (4) uses a three-step factorization strategy. We describe it in the occlusion-free case only. An iterative extension for the missing data case will be proposed in a forthcoming paper.

---

<sup>1</sup> Without loss of generality, we assume the same number of points to be present in the two reconstructions since only the correspondences are used for alignment.

We propose the important concept of orthonormalizing. We define a reconstruction to be in an orthonormal basis if the joint projection matrix is column-orthonormal. Given a joint projection matrix  $\mathcal{P}$ , one can find a 3D affine transformation  $N_{(3 \times 3)}$  such that  $\mathcal{P}N$  is column-orthonormal, such that  $N^T \mathcal{P}^T \mathcal{P}N = I_{(3 \times 3)}$ . We call  $N$  an orthonormalizing transformation. The set of orthonormalizing transformations is 3-dimensional since for any 3D rotation matrix  $U$ ,  $NU$  still is an orthonormalizing transformation for  $\mathcal{P}$ . We use the QR decomposition  $\mathcal{P} = QR$ , see [4], giving an upper triangular orthonormalizing transformation  $N = R^{-1}$ . Other choices are possible for computing an  $N$ , if  $\mathcal{P} = U\Sigma V^T$  is an SVD of  $\mathcal{P}$ , then  $N = V\Sigma^{-1}$  has the required property. Henceforth, we assume that all 3D models are expressed in orthonormal bases:  $\mathcal{P} \leftarrow \mathcal{P}N$ ,  $\mathcal{P}' \leftarrow \mathcal{P}'N'$ ,  $\mathcal{Q} \leftarrow N^{-1}\mathcal{Q}$  and  $\mathcal{Q}' \leftarrow N'^{-1}\mathcal{Q}'$ . An interesting property of orthonormal bases is that  $\mathcal{P}^\dagger = \mathcal{P}^T$ . Hence, triangulating points in these bases is simply done by  $\mathcal{Q} = \mathcal{P}^T\mathcal{X}$ .

Note that the matrix  $\mathcal{P}$  computed by factorization, see §3, may already satisfy  $\mathcal{P}^T \mathcal{P} = I$ . However, if at least one of the cameras is not used for the alignment, if none of the 3D point correspondences project in this camera, or if the cameras come as the result of the alignment of partial 3D models, then  $\mathcal{P}$  will not satisfy  $\mathcal{P}^T \mathcal{P} = I$ , thus requiring the orthonormalization step.

The translation part of the sought-after transformation can not be computed directly, but can be eliminated from the equations. First, centre the image points to eliminate the translation part of the cameras:  $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathbf{t}_i$  and  $\mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathbf{t}'_i$ . Second, consider that the partial derivatives of the reprojection error (3) with respect to  $\hat{\mathbf{t}}$  must vanish:  $\frac{\partial \mathcal{C}^2}{\partial \mathbf{t}} = 0$ . By using the constraint  $\hat{\mathbf{Q}}'_j = \hat{\mathbf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}$  from equation (3) and expanding:  $\sum_{i=1}^{n'} \sum_{j=1}^m \left( \mathbf{P}'^T \mathbf{P}'_i \hat{\mathbf{t}} - \mathbf{P}'^T \mathbf{x}'_{ij} + \mathbf{P}'^T \mathbf{P}'_i \hat{\mathbf{A}}\hat{\mathbf{Q}}_j \right) = m\mathcal{P}'^T \mathcal{P}' \hat{\mathbf{t}} - m\mathcal{P}'^T \bar{\mathcal{Y}}' + m\mathcal{P}'^T \mathcal{P}' \hat{\mathbf{A}}\bar{\hat{\mathbf{Q}}} = 0$ , which leaves us with  $\hat{\mathbf{t}} = (\mathcal{P}'^T \mathcal{P}')^{-1}(\mathcal{P}'^T \bar{\mathcal{Y}}' - \mathcal{P}'^T \mathcal{P}' \hat{\mathbf{A}}\bar{\hat{\mathbf{Q}}})$  that further simplifies to  $\hat{\mathbf{t}} = \mathcal{P}'^\dagger \bar{\mathcal{Y}}' - \hat{\mathbf{A}}\bar{\hat{\mathbf{Q}}}$  and, thanks to the orthonormal basis property  $\mathcal{P}'^\dagger = \mathcal{P}'^T$ , we get:

$$\hat{\mathbf{t}} = \mathcal{P}'^T \bar{\mathcal{Y}}' - \hat{\mathbf{A}}\bar{\hat{\mathbf{Q}}}, \quad (5)$$

Note that if the same entire sets of reconstructed points are used for the alignment, then we directly obtain  $\hat{\mathbf{t}} = \mathbf{0}$  since  $\bar{\mathcal{Y}}' = \mathbf{0}$  and  $\bar{\hat{\mathbf{Q}}} = \mathbf{0}$ . This is rarely the case in practice, especially if the alignment is used to merge partial 3D models.

Third, consider that the  $m$  partial derivatives of the reprojection error (3) with respect to each  $\hat{\mathbf{Q}}_j$  must vanish as well:  $\frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{Q}}_j} = 0$ , and expand as above:  $\mathcal{P}^T \mathcal{P} \hat{\mathbf{Q}}_j - \mathcal{P}^T \mathcal{Y}_j + \hat{\mathbf{A}}^T \mathcal{P}'^T \mathcal{P}' \hat{\mathbf{A}}\hat{\mathbf{Q}}_j - \hat{\mathbf{A}}^T \mathcal{P}'^T \mathcal{Y}'_j + \hat{\mathbf{A}}^T \mathcal{P}'^T \mathcal{P}' \hat{\mathbf{t}} = 0$ . The sum over  $j$  of all these derivatives also vanishes:  $(\forall j, \frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{Q}}_j} = 0) \Rightarrow \left( \sum_{j=1}^m \frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{Q}}_j} = 0 \right)$ , giving  $\mathcal{P}^T \mathcal{P} \bar{\hat{\mathbf{Q}}} - \mathcal{P}^T \bar{\mathcal{Y}} + \hat{\mathbf{A}}^T \mathcal{P}'^T \mathcal{P}' \bar{\hat{\mathbf{Q}}} - \hat{\mathbf{A}}^T \mathcal{P}'^T \bar{\mathcal{Y}}' + \hat{\mathbf{A}}^T \mathcal{P}'^T \mathcal{P}' \hat{\mathbf{t}} = 0$ . By replacing  $\hat{\mathbf{t}}$  by its expression (5), and after some minor algebraic manipulations, we obtain

$\mathcal{P}^\top \mathcal{P} \bar{\mathcal{Q}} - \mathcal{P}^\top \bar{\mathcal{Y}} = 0$  and  $\bar{\mathcal{Q}} = \mathcal{P}^\dagger \bar{\mathcal{Y}}$ . By substituting in equation (5) and using the orthonormal basis property  $\mathcal{P}^\dagger = \mathcal{P}^\top$ , we get:

$$\hat{\mathbf{t}} = \mathcal{P}'^\top \bar{\mathcal{Y}}' - \hat{\mathbf{A}} \mathcal{P}^\top \bar{\mathcal{Y}}. \quad (6)$$

It is common in factorization methods to centre the data with respect to their centroid to cancel the translation part of the transformation. Equation (6) means that, according to the reprojection error criterion, the data must be centred with respect to the centroid of the image points, not with respect to the actual 3D centroid.

Obviously, if the 3D models have been obtained by the factorization method of §3, then the centroid of the 3D points corresponds to the reconstructed centroid,  $\hat{\mathbf{Q}} = \mathcal{P}^\top \bar{\mathcal{Y}}$  and  $\hat{\mathbf{Q}}' = \mathcal{P}'^\top \bar{\mathcal{Y}}'$ , provided that the same sets of views are used for reconstruction and alignment.

To summarize, we cancel the translation part out of the sought-after transformation by translating the reconstructions and the image points by  $\mathbf{Q}_j \leftarrow \mathbf{Q}_j - \mathcal{P}^\top \bar{\mathcal{Y}}$  and  $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathbf{P}_i \mathcal{P}^\top \bar{\mathcal{Y}}$ , and similarly for the second image set. The reprojection error (3) is rewritten:

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2nm} (\|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 + \|\mathcal{X}' - \mathcal{P}'\mathcal{Q}'\|^2), \quad (7)$$

and problem (4) is reformulated as  $\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}')$  s.t.  $\hat{\mathbf{Q}}'_j = \hat{\mathbf{A}} \hat{\mathbf{Q}}_j$ .

Thanks to the orthonormal basis property  $\mathcal{P}^\top \mathcal{P} = \mathbf{I}$ , and since for any column-orthonormal matrix  $\mathcal{A}$ ,  $\|\mathcal{A}\mathbf{x}\| = \|\mathbf{x}\|$ , we can rewrite the reprojection error on a single set of cameras as  $\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) \propto \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 = \|\mathcal{P}^\top \mathcal{X} - \mathcal{Q}\|^2$ . This allows to rewrite the reprojection error (7) as:

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \|\mathcal{P}^\top \mathcal{X} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{P}'^\top \mathcal{X}' - \hat{\mathcal{Q}}'\|^2 = \underbrace{\|\begin{pmatrix} \mathcal{P}^\top \mathcal{X} \\ \mathcal{P}'^\top \mathcal{X}' \end{pmatrix}\}_{\Lambda}}_{\Delta} - \underbrace{\begin{pmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{pmatrix}}_{\tilde{\mathcal{M}}} \|^2.$$

By introducing the constraint  $\hat{\mathcal{Q}}' = \hat{\mathbf{A}} \hat{\mathcal{Q}}$  and, as in §3, an unknown global affine transformation  $\mathbf{B}$ :

$$\Delta = \begin{pmatrix} \mathbf{I} \\ \hat{\mathbf{A}} \end{pmatrix} \mathbf{B} \mathbf{B}^{-1} \hat{\mathcal{Q}} = \underbrace{\begin{pmatrix} \mathbf{B} \\ \hat{\mathbf{A}} \mathbf{B} \end{pmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathbf{B}^{-1} \hat{\mathcal{Q}}}_{\tilde{\mathcal{Q}}}.$$

The problem is reformulated as  $\min_{\tilde{\mathcal{M}}, \tilde{\mathcal{Q}}} \|\Delta - \tilde{\mathcal{M}} \tilde{\mathcal{Q}}\|^2$ . A solution is given by SVD of matrix  $\Delta$ :  $\Delta_{(6 \times m)} = \mathbf{U}_{(6 \times 6)} \Sigma_{(6 \times 6)} \mathbf{V}_{(6 \times m)}^\top$ . As in §3, let  $\Sigma = \Sigma_u \Sigma_v$  be any decomposition of matrix  $\Sigma$ . We obtain  $\tilde{\mathcal{M}} = \psi(\mathbf{U} \Sigma_u)$  and  $\tilde{\mathcal{Q}} = \psi^\top(\mathbf{V} \Sigma_v^\top)$ . Using the partitioning  $\tilde{\mathcal{M}}^\top = (\tilde{\mathbf{M}}^\top \tilde{\mathbf{M}}'^\top)$ , we get  $\mathbf{B} = \tilde{\mathbf{M}}$ ,  $\hat{\mathbf{A}} = \tilde{\mathbf{M}}' \mathbf{B}^{-1}$  and  $\hat{\mathcal{Q}} = \mathbf{B} \tilde{\mathcal{Q}}$ . Obviously, one needs to undo the effect of the orthonormalizing transformations:  $\hat{\mathbf{A}} \leftarrow \mathbf{N} \hat{\mathbf{A}} \mathbf{N}^{-1}$  and  $\hat{\mathcal{Q}} \leftarrow \mathbf{N} \hat{\mathcal{Q}}$ . A minimal  $m \geq 4$  point correspondences is required.

Note that it is possible to solve the problem without using the orthonormalizing transformations. This solution requires however to compute the SVD of a  $(2(n + n') \times m)$  matrix, made by stacking the measurement matrices  $\mathcal{X}$  and  $\mathcal{X}'$ , and is therefore much more computationally expensive than the algorithm above, and may be intractable for large sets of cameras and points.

## 5 Other Algorithms

We briefly describe two other alignment algorithms. They do not yield Maximum Likelihood Estimates under the previously-mentioned hypotheses on the noise distribution. They rely on 3D measurements and naturally handle missing data.

### 5.1 Minimizing the Non-Symmetric Transfer Error

This algorithm, dubbed ‘TRERROR’, is specific to the two camera set case. It is based on minimizing a non-symmetric 3D transfer error  $\mathcal{E}(\hat{\mathbf{A}})$ :  $\min_{\hat{\mathbf{A}}, \hat{\mathbf{t}}} \mathcal{E}^2(\hat{\mathbf{A}}, \hat{\mathbf{t}})$  with  $\mathcal{E}^2(\hat{\mathbf{A}}) = \frac{1}{m} \sum_{j=1}^m \|\mathbf{Q}'_j - \hat{\mathbf{A}}\mathbf{Q}_j - \hat{\mathbf{t}}\|^2$ . Differentiating  $\mathcal{E}^2$  with respect to  $\hat{\mathbf{t}}$  and nullifying the result yields  $\hat{\mathbf{t}} = \hat{\mathbf{Q}}' - \hat{\mathbf{A}}\hat{\mathbf{Q}}$ . Henceforth, we assume that the translation has been eliminated by translating each 3D point set on its centroid. By rewriting the error function as  $\mathcal{E}^2(\hat{\mathbf{A}}) \propto \|\mathcal{Q}' - \hat{\mathbf{A}}\mathcal{Q}\|^2$  and applying standard linear least-squares, one obtains the solution  $\hat{\mathbf{A}} = \mathcal{Q}'\mathcal{Q}^\dagger$ .

### 5.2 Direct 3D Factorization

This algorithm, dubbed ‘FACT3D’, is based on directly factorizing the 3D reconstructed points. It is not restricted to the two camera set case, but for simplicity, we only describe this case. Generalization to multiple camera sets is trivial. The algorithm computes the aligning transformation  $(\hat{\mathbf{A}}, \hat{\mathbf{t}})$  and perfectly corresponding points  $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$ . The reconstructed cameras are not taken into account by this algorithm, which entirely relies on 3D measures on the reconstructed points. This algorithm is equivalent to the proposed FACTMLE under certain conditions.

The problem is stated by  $\min_{\hat{\mathbf{Q}}, \hat{\mathbf{Q}}'} \mathcal{D}^2(\hat{\mathbf{Q}}, \hat{\mathbf{Q}}')$  s.t.  $\hat{\mathbf{Q}}'_j = \hat{\mathbf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}$ , here the 3D error function employed is defined by  $\mathcal{D}^2(\hat{\mathbf{Q}}, \hat{\mathbf{Q}}') = \frac{1}{2m} (\|\mathcal{Q} - \hat{\mathbf{Q}}\|^2 + \|\mathcal{Q}' - \hat{\mathbf{Q}}'\|^2)$ . Minimizing this error function means that if the noise were Gaussian, centred and i.i.d., which is the case with our actual hypotheses (the noise distribution in 3D depends on the noise distribution in the images and the reconstruction method – hence it is not a priori Gaussian), then this algorithm would yield the Maximum Likelihood Estimate.

By nullifying the partial derivatives of the error function  $\mathcal{D}^2$  with respect to  $\hat{\mathbf{t}}$  and with respect to the  $\hat{\mathbf{Q}}_j$ , and substituting the latter expressions into the former one, we obtain  $\hat{\mathbf{t}} = \bar{\mathbf{Q}}' - \hat{\mathbf{A}}\bar{\mathbf{Q}}$ . This equation

means that, as in most factorization methods, cancelling the translation part out according to the error function  $\mathcal{D}$  is done by centring each set of 3D points on its actual centroid:  $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j - \bar{\mathbf{Q}}$  and  $\hat{\mathbf{Q}}'_j \leftarrow \hat{\mathbf{Q}}'_j - \bar{\mathbf{Q}}'$ . Henceforth, we assume to work in centred coordinates. The problem is rewritten as  $\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}')$  s.t.  $\hat{\mathbf{Q}}'_j = \hat{\mathbf{A}}\hat{\mathbf{Q}}_j$ .

Following the approach in §4.2, we rewrite  $\mathcal{D}$  as:

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \| \begin{pmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{pmatrix} - \begin{pmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{pmatrix} \|^2 = \| \underbrace{\begin{pmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{pmatrix}}_{\Lambda} - \underbrace{\begin{pmatrix} \mathbf{B} \\ \mathbf{AB} \end{pmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathbf{B}^{-1}\hat{\mathcal{Q}}}_{\tilde{\mathcal{Q}}} \|^2.$$

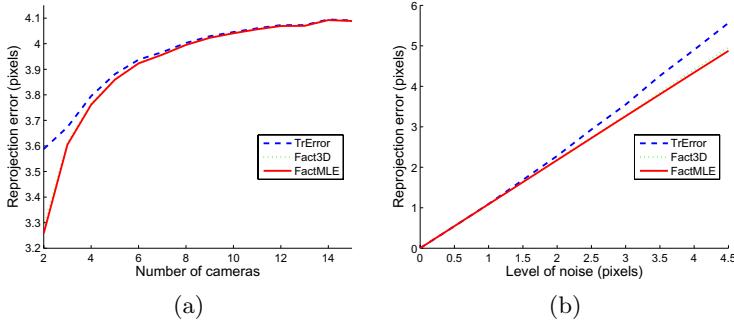
Using SVD of matrix  $\Lambda = \mathbf{U}\Sigma\mathbf{V}^\top$ , we obtain  $\tilde{\mathcal{M}} = \psi(\mathbf{U}\Sigma_u)$  and  $\tilde{\mathcal{Q}} = \psi^\top(\mathbf{V}\Sigma_v^\top)$ . By partitioning  $\tilde{\mathcal{M}}^\top = (\tilde{\mathbf{M}}^\top \ \tilde{\mathbf{M}}'^\top)$ , we get  $\mathbf{B} = \tilde{\mathbf{M}}$ ,  $\hat{\mathbf{A}} = \tilde{\mathbf{M}}'\mathbf{B}^{-1}$  and  $\hat{\mathcal{Q}} = \mathbf{B}\tilde{\mathcal{Q}}$ .

## 6 Experimental Evaluation

### 6.1 Simulated Data

We generated  $m$  3D points and two sets of  $n$  weak perspective cameras:  $\mathbf{P}_i = \mathbf{A}_i\bar{\mathbf{R}}_i$ , where  $\mathbf{A}_i$  is the internal calibration matrix  $\mathbf{A}_i = k_i \text{diag}(\tau_i, 1)$ ,  $\bar{\mathbf{R}}_i$  a  $(2 \times 3)$ , truncated, 3D rotation matrix and  $\mathbf{t}_i$  is a 2-vector. The scale factor  $k_i$  models the average depth of the object and the focal length of the camera, and  $\tau$  models the aspect ratio that we choose very close to 1. The 3D points are chosen from a uniform distribution inside a thin rectangular parallelepiped with dimensions  $1 \times 1 \times (1-d)$ , and the internal camera scale factors  $k_i$  are chosen so that the points are uniformly spread in  $400 \times 400$  pixel images. We use  $m$  points to perform Structure-From-Motion on each camera set and  $m_c$  points for the alignment. A gaussian noise with zero mean and standard deviation  $\sigma$  is added in the images. We define the overlap ratio of the two camera sets to be  $\theta = m_c/m$ . for  $\theta = 1$  all points are seen in all views, while for  $\theta = 0$ , the two sets of cameras do not share corresponding points. The comparison of the algorithms being based on the reprojection error, the point clouds used to compute it need to be re-estimated so that this error is minimized, given an estimated transformation. This must be done for TRERROR and FACT3D.

The default setting is:  $n = 2$  views,  $m = 250$  points,  $\theta = 0.2$  ( a 20% overlap and  $m_c = 50$  points common to the two 3D models),  $\sigma = 3.0$  pixels,  $d = 0.95$  (flat 3D scene) and  $a = 1$  (perfectly affine projections). Figure 2 shows the reprojection error averaged over 500 simulations for varying the number  $n$  of cameras and the level of noise  $\sigma$ . Whereas FACTMLE and FACT3D have similar behaviors, TRERROR is less robust with regard to both of these parameters. Other experiments concern varying the overlap ratio and the number of points  $m_c$ , the former from 10% to 100% and the latter from 4 to 60, corresponding respectively to  $m = 20$  and  $m = 300$  points. For small values of  $m_c$ , FACTMLE



**Fig. 2.** Reprojection error versus (a) the number  $n$  of cameras and (b) the noise  $\sigma$

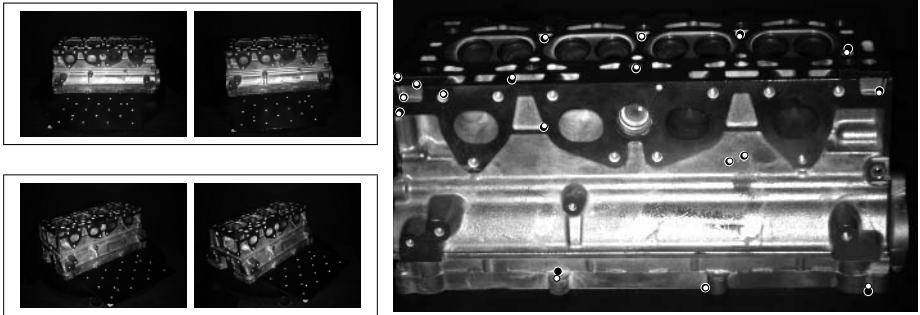
and FACT3D yield very similar errors, whereas for higher numbers of points, the difference gets larger. We see that for  $m_c > 20$ , the number of points has a much smaller influence on the errors. However, neither the error nor the difference between the methods seems to change when the overlap changes and we thus conclude that the overlap does not much influence the alignment algorithms, when the number of points is high enough. In all cases, method TRERROR performs very badly compared to the two other ones. Finally, the flatness of the simulated data  $d$  is varied from 0 to 1, that is from a cube to a plane. The flatness of the scene does not change the result of the alignment, except for very flat scenes where TRERROR turns out to be unstable. This result was expected since planar scenes are singular for the computation of a 3D affine transformation. The result means that TRERROR is much more sensitive to noise than the two other methods. We conclude that although FACTMLE consistently outperforms the other two algorithms, it is in critical situation that the difference seems to be the most important. In particular, TRERROR is less robust with regard to the number of cameras, the noise and the flatness of the scene.

## 6.2 Real Data

We applied the algorithms to real image sequences. For one of them, the ‘cylinder head’ sequence, we show results. The video camera is not calibrated, that is, the internal parameters are unknown but constant throughout the video footage.

The images are shown on figure 3 with the  $m_c = 13$  original, manually entered, and reprojected points after we applied the FACTMLE method. The pictures were taken with a camera with 12 mm focal length, at a distance of approximately 60 cm of the object, which is 40 cm long. The reprojection errors we obtained are: 3.7683 pixels for FACTMLE, 3.7692 pixels for FACT3D and 3.7764 pixels for TRERROR.

The ‘statuette sequence’ is made of 4 images with  $m_c = 14$  points lying very close to a plane, and  $m = m' = 25$ , giving a 56% overlap. The points were manually entered. The reprojection errors we obtained are: 2.8402 pixels for FACTMLE, 2.8415 pixels for FACT3D and 2.8446 pixels for TRERROR.



**Fig. 3.** (left) Both sets of images of the cylinder head sequence and (right) closeup overlaid with the original points (black) and reprojected points (white) from FACTMLE

The ‘book sequence’ consists of 5 images with  $m_c = 196$ ,  $m = 628$  and  $m' = 634$  points given by an automatic correlation-based tracker, giving a 31% overlap. The reprojection errors we obtained are: 1.8961 pixels for FACTMLE, 1.9737 pixels for FACT3D and 2.1690 pixels for TRERROR.

In accordance with the results on simulated data, we observe that in critical situations, FACTMLE outperforms the other two methods. The reprojection errors of the order of a few pixels indicate that the data are well-modeled.

## 7 Conclusions

We presented a method to compute the Maximum Likelihood Estimate of 3D affine transformations, under standard hypotheses on the noise distribution, aligning sets of 3D points obtained from uncalibrated affine cameras.

Future work could be devoted to the experimental validation of the method in the missing data case, and the incorporation of other types of features, namely line, planar curve and plane correspondences.

## References

1. Fitzgibbon, A., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: European Conference on Computer Vision. (1998) 311–326
2. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization method. International Journal of Computer Vision **9** (1992) 137–154
3. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Graphics and Image Processing **24** (1981) 381 – 395
4. Golub, G., van Loan, C.: Matrix Computation. The Johns Hopkins University Press, Baltimore (1989)
5. Jacobs, D.: Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In: Computer Vision and Pattern Recognition (1997) 206–212

# A New Class of Learnable Detectors for Categorisation

Jiri Matas and Karel Zimmermann

Center for Machine Perception,  
Faculty of Electrotechnical Engineering,  
Czech Technical University in Prague

**Abstract.** A new class of image-level detectors that can be adapted by machine learning techniques to detect parts of objects from a given category is proposed. A classifier (e.g. neural network or adaboost trained classifier) within the detector selects a relevant subset of extremal regions, i.e. regions that are connected components of a thresholded image. Properties of extremal regions render the detector very robust to illumination change. Robustness to viewpoint change is achieved by using invariant descriptors and/or by modeling shape variations by the classifier.

The approach is brought to bear on three problems: text detection, face segmentation and leopard skin detection. High detection rates were obtained for unconstrained (i.e. brightness, affine and font invariant) text detection (92%) with a reasonable false positive rate.

The time-complexity of the detection is approximately linear in the number of pixels and a non-optimized implementation runs at about 1 frame per second for a  $640 \times 480$  image on a high-end PC.

## 1 Introduction

Methods relying on correspondences of local affine or scale covariant regions have furthered research in a number of areas of computer vision including object recognition [15, 8, 13, 5], wide-baseline stereo [14, 17, 4, 10, 11], tracking [3], categorisation [16, 1, 7] and texture recognition [6]. As a first step, the cited approaches detect a set of transformation-covariant regions that are stable both under illumination variations and local geometric transformations (either similarity or affine) induced by a viewpoint change. The detectors are generic and they have been shown to perform well in a wide range of environments.

In categorisation, the problem we focus on, state-of-the-art approaches represent categories as probabilistic configurations of classified transformation-covariant regions [2, 16, 7, 12]. The (soft) classification of the



**Fig. 1.** Text detection based on category-specific extremal regions

transformation-covariant regions into components (parts) is based on rules learned in a training stage. The region detectors used in categorisation are generic, e.g. the salient regions of Kadir and Brady in the categorisation systems of Fergus et al.[2] and Fei-Fei et al.[1] or the affine-invariant interest points of Mikolajczyk and Schmid[11] and MSER regions [10] in the VideoGoogle system of Sivic and Zisserman [16].

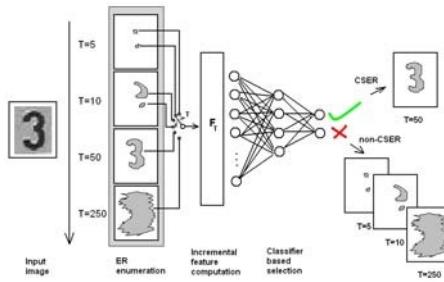
As a main contribution of the paper, a new class of machine learnable category-specific detectors of covariant regions is presented. Machine learning techniques have been applied in the context of categorisation to find a representation of the configuration [16, 18] and to train classifiers for recognition of regions — components of the configuration [2, 16]. In this paper, machine learning is newly introduced to the image processing level i.e. it becomes part of the design of a category-specific detector. The benefits of learning at the detector level are demonstrated on two classical categorisation problems: text detection in images and licence plate recognition.

The proposed category-specific class of detectors is trained to select a relevant subset of extremal regions. A robust category-specific detector of extremal regions can be implemented as follows. Enumerate all extremal regions, compute efficiently a description of each region and classify the region as relevant or irrelevant for the given category. In a learning stage, the classifier is trained on examples of regions — components of objects from a given class. Such detection algorithm is efficient only if features (descriptors) for each region are computed in constant time. We show there is a sufficiently discriminative class of ‘incrementally computable’ features on extremal regions satisfying this requirement.

The proposed detector is robust to many image transformations. The affine invariance is achieved in reasonable scale by learning. The partial occlusion robustness, depicted in Figure 5, is caused by decomposition of the object to small individually detectable regions. The illumination invariance is demonstrated in Figure 1. Two images of a scene with different contrast levels are shown. A class-specific detector of character-like regions (the arrow and the pound sign are not in the training set) processed the two images. An object belonging to a ‘text’ class is defined as a (approximately) linear configuration of more than one character-like extremal regions. The hand-written text is detected even in the extremely low contrast image at the bottom of Fig. 1.

## 2 Category-Specific Extremal Region Detection

Our objective is to select from the set of extremal regions those with shape belonging to a given category. The model of the category is acquired in a separate training stage. Let us assume for the moment that the learning stage produced a classifier that, with some error, is able to assign to each extremal region one of two labels: ‘interesting’, i.e. is a component of our category, or ‘non-interesting’ otherwise. The detection of category-specific extremal regions can be then arranged as three interleaved steps: (1) generate a new extremal



**Fig. 2.** The detection is implemented as interleaved enumeration of extremal regions, computation of incremental features and classification

region, (2) describe the region and (3) classify it. The interleaved computation is schematically depicted in Figure 2.

Extremal regions are connected components of an image binarised at a certain threshold. More formally, an extremal region  $r$  is a contiguous set of pixels such that for all pixels  $p \in r$  and all pixels  $q$  from the outer boundary  $\partial r$  of region  $r$  either  $I(p) < I(q)$  or  $I(p) > I(q)$  holds. In [10], it is shown that extremal regions can be enumerated simply by sorting all pixels by intensity either in increasing or decreasing order and marking the pixels in the image in the order. Connected components of the marked pixels are the extremal regions. The connected component structure is effectively maintained by the union-find algorithm.

In this process, exactly one new extremal region is formed by marking one pixel in the image. It is either a region consisting of a single pixel (a local extremum), a region formed by a merge of regions connected by the marked pixel, or a region that consists of union of an existing region and the marked pixels. It is clear from this view of the algorithm that there are at most as many extremal regions as there are pixels in the image. The process of enumeration of extremal regions is nearly linear in the number of pixels<sup>1</sup> and runs at approximately 10 frames per second on 2.5 GHz PC for a  $700 \times 500$  image.

To avoid making the complexity of the detection process quadratic in the number of image pixels, the computation of region description must not involve all of its pixels. Fortunately, a large class of descriptors can be computed incrementally in constant time even in the case of a merge of two or more extremal regions (the other two situations are special cases). Importantly, combinations of incrementally computable features include affine and scale invariants. Incrementally computable features are analysed in Section 3.

The final step of the CSER detection, the selector of category-specific regions, is implemented as a simple neural network trained on examples of regions - components of the category of interest. The neural network selects rel-

<sup>1</sup> The (negligibly) non-linear term is hidden in the "maintenance of connected component structure".

event regions in constant time. The overall process of marking a pixel, recalculating descriptors and classifying is thus constant time. The choice of neural network is arbitrary and any other classifier such as SVM or AdaBoost could replace it.

### 3 Incrementally Computable Region Descriptors

In the CSER detection process the descriptors of a connected component that evolves have to be computed. The evolution has two forms: growing and merging of regions. It is easy to see that if we can compute the description of a union  $r_1 \cup r_2$  of two regions  $r_1$  and  $r_2$  then we can compute it in each step of the evolution (we use  $r_i$  to identify both the region and its set of pixels). The following problem arises: what image features computed on the union of the regions can be obtained in constant time from some characterisation  $g$  of  $r_1$  and  $r_2$ ?

For example, let us suppose that we want to know the second central moment of the merged region. It is known that the second central moment (moment of inertia) can be computed in constant time from the first and second (non-central) moments and first and second (non-central) moments can be updated in the merge operation in constant time. A region descriptor (feature)  $\phi$  will be called if the following three functions exists: a characterising function  $g : 2^{Z^2} \rightarrow \mathcal{R}^m$ , a characterisation update function  $f : (\mathcal{R}^m, \mathcal{R}^m) \rightarrow \mathcal{R}^m$ , and a feature computation function  $\phi : \mathcal{R}^m \rightarrow \mathcal{R}^n$ , where  $m$  is constant,  $n$  is the dimension of the feature and  $Z^2$  is the image domain.

For each region, the characterising function  $g$  returns the information necessary for computing feature  $\phi$  in a real vector of dimension  $m$ . The dimension  $m$  of the characteristic vector depends on the feature, but is independent of region size. Given the characterisation returned by  $g$ , the  $n$ -dimensional feature of interest (region descriptor) is returned by  $\phi$ . Function  $f$  computes the characterisation of the merged region given the characterisation of the regions  $r_1$ ,  $r_2$ . For efficiency reasons, we are looking for features with the smallest characterisation dimension  $m^*$ . An incremental feature is a triplet of functions  $(g^*, f^*, \phi^*)$  defined as

$$g^* = \arg \min_g \{\dim(g(2^{Z^2}))\} \text{ subject to } \phi(g(r_1 \cup r_2)) = \phi(f(g(r_1), g(r_2))).$$

**Example 1.** Minimum intensity  $I$  of all pixels in a region is an incrementally computable feature with dimension  $m^* = 1$ . Given regions  $r_1$  and  $r_2$  with pixels  $r_1^i \in r_1, r_2^j \in r_2$ , the description of the union regions  $r_1, r_2$  is

$$\phi(g(r_1 \cup r_2)) = \underbrace{1}_{\phi} \cdot \underbrace{\min_f \{ \underbrace{\min_{r_1^i \in r_1} I(r_1^i)}, \underbrace{\min_{r_2^j \in r_2} I(r_2^j)} \}}_{g(r_1) \quad g(r_2)}$$

**Example 2.** The center of gravity ( $m^* = 2$ ) of a union of regions  $r_1, r_2$  with pixels  $r_1^i, r_2^j$  for  $i = 1 \dots k_1, j = 1 \dots k_2$  is

$$\phi(g(r_1 \cup r_2)) = \underbrace{\frac{1}{k_1 + k_2}}_{\phi} \left( \underbrace{\sum_{i=1}^{k_1} r_1^i}_{g(r_1)} + \underbrace{\sum_{j=1}^{k_2} r_2^j}_{g(r_2)} \right).$$

In this paper we use the following incrementally computable features: . . . . . with  $m^* \sim (k)^2$  where  $k$  is an moment order (calculation based on algebraic moments), . . . . . with  $m^* = 2$  (using the area and the border), . . . . . of a region with  $m^* = 2$ , . . . . . with  $m^* = 2$ . Features that we are not able to compute incrementally are e.g. the number convexities and the area of convex hull.

## 4 Experiments - Applications and Properties of CSER Detection

### 4.1 Text Detection and Properties of CSER

$\theta \setminus \phi$	$0^\circ$	$15^\circ$	$30^\circ$	$45^\circ$
$0^\circ$	2.6	2.8	2.8	3.0
$10^\circ$	3.2	3.2	3.2	3.8
$20^\circ$	3.2	3.6	4.0	7.8
$30^\circ$	7.6	8.4	15.2	26.5

**Fig. 3.** (a) False negative rate (missed characters) as a function of viewing angles  $\phi$  (elevation),  $\theta$  (azimuth); in percentage points

The favorable properties (e.g. bright and affine invariance or speed) of CSER detector are demonstrated in this experiment. We have decided for text detection problem only of one font to present mentioned properties of the detector.

The category of texts is modeled as a linear constellation of CSERs. The feed-forward neural network for CSER selection was trained by a standard back-propagation algorithm on approximately 1600 characters semi-automatically segmented from about 250 images acquired in unconstrained conditions. The region descriptor was formed by scale-normalised algebraic moments of the characteristic function up the fourth order, compactness and entropy of the intensity values. Intentionally, we did not restrict the features to be either rotation or affine invariant and let the neural network with 15 hidden nodes to model feature variability.

The detection of text proceeds in two steps. First, relevant CSER selected as described above. Second, linear configurations of regions are found by Hough transform. We impose two constraints on the configurations: the CSER regions must be formed from more than three regions and the regions involved must have a similar height.

**Detection Rate.** On an independent test set of 70 unconstrained images of scenes the method achieved 98% detection rate with a false positive appearing in approximately 1 in 20 images.

**Speed.** The detection time is proportional to the number of pixels. For a 2.5 GHz PC the processing took 1.1 seconds for a  $640 \times 480$  image and 0.25 seconds for  $320 \times 240$  image.

**Robustness to viewpoint** change was indirectly tested by the large variations in the test data where scales of texts differed by a factor of 25 (character 'heights' ranged from approximately 7-8 to 150 pixels) and were viewed both frontally and at acute angles. We also performed systematic evaluation of the CSER detector. Images of texts were warped to simulate a view from a certain point on the viewsphere. The false negative rates for the CSER detector (missed character percentages) with approximately 10 false positive regions per background image are shown in Table 3a). The CSER detector is stable for almost the whole tested range. Even the 27% false negative at the  $30^\circ$ - $45^\circ$  elevation-azimuth means that three quarters of characters on the text sign are detected which gives high probability of text detection.

**Robustness to illumination** change was evaluated in a synthetic experiment. Intensity of images taken in daylight was multiplied by a factor ranging from 0.02 to 1. As shown in Figure 4b, the false negative (left) and false positive (right) rates were unchanged for both the detector of CSER (bottom) and whole text signs (top) in the (0.1, 1) range! For the 0.1 intensity attenuation, the image has at most 25 intensity levels, but thresholds still exist that separate the CSERs. The experiment also suggests that the interleaving of extremal region enumeration, description and classification cannot be simply replaced by detection of MSERs followed by MSER description and classification.

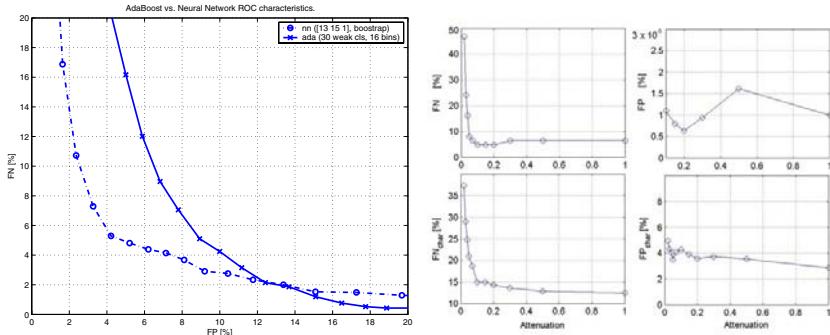
**Robustness to occlusion** is a consequence of modeling the object as a configuration of local components. Occlusion of some components does not imply the object is not detected.

**Independence of internal classifier** is presented by implementation of different classifier. The ROC characteristic in the Fig.4a shows the comparison of achieved results with built-in neural network and adaboost classifiers. Considering that detector is originally designed as filter we can see that for acceptable rate of FP bigger than 15% adaboost provides lower false negative rate than neural network. In the other hand, achieved results renders detector to be able to work alone (without any post-processing considering only linear constellation constraints) and for a such application neural network brings better results in the interval of  $FP \leq FN$ .

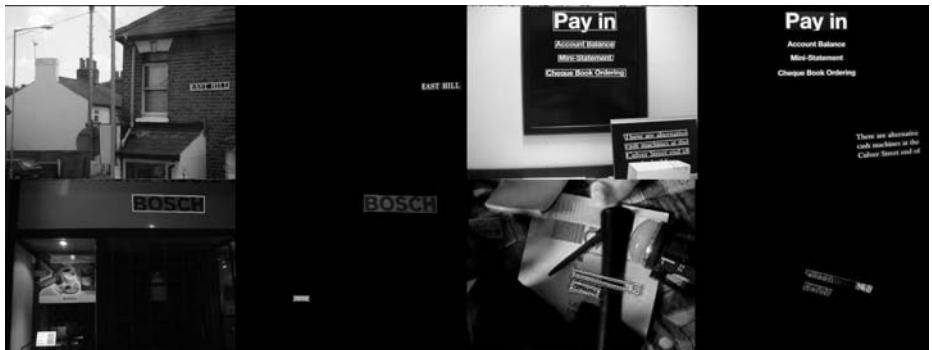
## 4.2 Text Detection in Unconstrained Conditions

We applied the CSER to the problem of text detection in images for which standard datasets are available. We used part of the ICDAR03 text detection competition set maintained by Simon Lucas at the University of Essex [9].

An object from the 'text category' was modeled as an approximately linear configuration of at least three 'text-like' CSERs. The neural network selector



**Fig. 4.** (a)The ROC characteristic of the proposed detector comparing results with built-in neural network and adaboost classifiers.(b) Text detection in images with attenuated intensity



**Fig. 5.** Text detection results

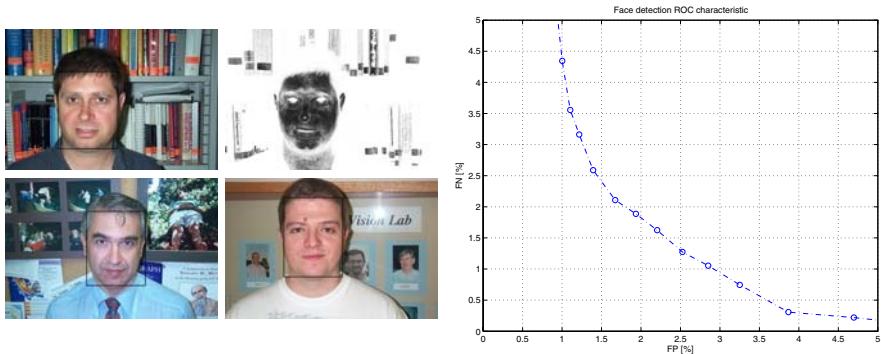
was trained on examples from the 54 images from Essex (the `ifsofa` subset) and 200 images of licence plates. Compared to the preceding experiment, the neural network (again with 15 hidden nodes) has to select CSER corresponding to letters of much higher variability (different fonts, both handwritten and printed characters).

The text detector was tested on 150 images from the `ryoungt` subset of the Essex data. The false negative rate (missed text) was 8% and 0.45 false positives were detected per image. No post-filtering of the result with an OCR method was applied to reduced false positives. Examples of text detection on the ICDAR Essex set are shown in Figures 1 (top) and 5 (top row).

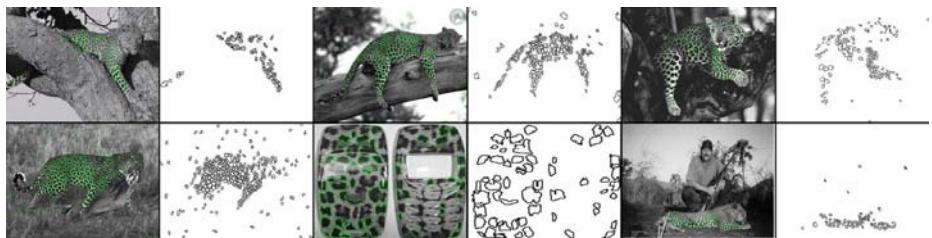
Further informal experiments were carried out to test insensitivity to lighting (Figure 1, center and bottom) and occlusion (Figure 5, bottom right). The image in the bottom left of Figure 5 includes two texts that have different scales and contrast; both are detected.

### 4.3 Face Detection

Extremal regions can be defined in colour images with respect to any ordering of RGB values. Intensity is just a particular scalar function that orders RGB values. This section describes and experiment, where human faces are detected as category-specific extremal regions. In this case, the scalar function is the likelihood ratio  $\lambda(RGB) = P(RGB|skin)/P(RGB|non-skin)$ . The assumption is that for a face there exists a threshold  $\theta$  on the likelihood ratio  $\lambda$  separating the face from the background. As the enumeration of extremal region with respect to skin likelihood ratio  $\lambda(RGB)$  proceeds, descriptors of the connected component are passed on to a classifier trained to detect face-like regions.



**Fig. 6.** Face detection:(a) Results and thresholding in the direction of skin probability, (b) ROC characteristic



**Fig. 7.** Leopard skin detection; (b) sample results

The results on Caltech Human face (front) dataset are summarized by ROC characteristic in Fig.6, where false positive rate is normalized with respect to all extremal regions in the image. In this ROC curve at 99.5% detection rate, only 3.5% of windows have to be verified. The results present detector as rapid region pre-selector , i.e. weak classifier with false negative rate close to zero.

#### 4.4 Leopard Skin Detection

The experiment on leopard skin detection shows whether CSERs can support detection of objects from the given category. We did not attempt to model the complex and flexible spatial configuration. The neural network was trained on spots from only four images. The spot-specific CSER detector than processed a number of images from the WWW. Sample results are shown in the Fig. 7. The density of CSER is high in the leopard skin area (skin-like area in the case of the mobile phone) and low elsewhere. The result suggest that learned CSER may be useful in viewpoint-independent texture detection.

### 5 Conclusions

We presented a new class of detectors that can be adapted by machine learning methods to detect parts of objects from a given category. The detector selects a category-relevant subset of extremal regions. Properties of extremal regions render the detector very robust to illumination change. Robustness to viewpoint change can be achieved by using invariant descriptors and/or by modeling shape variations by the classifier.

The detector was tested in three different tasks (e.g. text detection, face segmentation or texture detection) with successful results. The task of text detection presents affine and brightness invariance, the experiment of face detection introduces the ability of detector to process color images by thresholding in the learnable direction in RGB space and texture detection experiment demonstrates variability of the proposed detector.

The method can only detect regions that are extremal regions. It is not clear whether this is a significant limitation. Certainly many objects can be recognised from suitably locally threshold images, i.e. from extremal regions. Also note that different extremal sets can be defined by ordering pixels according to totally ordered quantities other than intensity, e.g. saturation. Efficiency of the method requires that the CSER are selected on the basis of incrementally computable features. This restriction can be overcome by viewing the interleaved classifier as a fast pre-selector in a cascaded (sequential) classification system.

### Acknowledgements

Karel Zimmermann was supported by The Czech Academy of Sciences Multi-Cam project 1ET101210407 and Jiri Matas was supported by The European Commission under COSPAL project IST-004176.

### References

1. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV03*, pages 1134–1141, 2003.

2. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*, pages II: 264–271, 2003.
3. V. Ferrari, T. Tuytelaars, and L. Van Gool. Real-time affine region tracking and coplanar grouping. In *CVPR*, pages II:226–233, 2001.
4. V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *CVPR*, 2003.
5. T. Kadir and M. Brady. Saliency, scale and image description. *IJCV01*, 45(2):83–105, 2001.
6. S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *ICCV03*, pages 649–655, 2003.
7. B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR03*, pages II: 409–415, 2003.
8. D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV99*, pages 1150–1157, 1999.
9. S. Lucas. Icdar03 text detection competition datasets. In <http://algoval.essex.ac.uk/icdar/Datasets.html>, 2003.
10. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC02*, volume 1, pages 384–393, London, UK, 2002.
11. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV01*, pages 525–531, 2001.
12. G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *CVPR01*, pages I:723–730, 2001.
13. S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *BMVC*, volume 1, pages 113–122, London, UK, 2002.
14. P. Pritchett and A. Zisserman. Matching and reconstruction from widely separated views. In *SMILE98*, 1998.
15. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–535, 1997.
16. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV03*, pages 1470–1477, 2003.
17. T. Tuytelaars and L. van Gool. Content-based image retrieval based on local affinely invariant regions. In *VIIS*, pages 493–500, 1999.
18. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV00*, pages I: 18–32, 2000.

# Overlapping Constraint for Variational Surface Reconstruction

Henrik Aanæs<sup>1</sup> and Jan Erik Solem<sup>2</sup>

<sup>1</sup> Informatics and Mathematical Modelling,  
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark  
[haa@imm.dtu.dk](mailto:haa@imm.dtu.dk)

<sup>2</sup> Jan Erik Solem, School of Technology and Society,  
Malmö University, 205 06 Malmö, Sweden  
[jes@ts.mah.se](mailto:jes@ts.mah.se)

**Abstract.** In this paper a counter example, illustrating a shortcoming in most variational formulations for 3D surface estimation, is presented. The nature of this shortcoming is a lack of an overlapping constraint. A remedy for this shortcoming is presented in the form of a penalty function with an analysis of the effects of this function on surface motion. For practical purposes, this will only have minor influence on current methods. However, the insight provided in the analysis is likely to influence future developments in the field of variational surface reconstruction.

## 1 Introduction

The reconstruction of surfaces from a set of calibrated cameras, or the so called (multiple view) stereo problem, is one of the main cornerstones of computer vision research. Of the magnitude of 3D reconstruction approaches it is often this surface reconstruction, which is needed in order for the end user to perceive a result as acceptable.

An approach to formulating the surface reconstruction problem, which has become popular within the last years, is by casting the problem in a variational framework c.f. e.g. [1, 2, 3, 4, 5, 6]. The problem is thus formulated using some functional, which has to be minimized. The variational approach to surface reconstruction has many advantages, where one of its main virtues is that it ensures theoretical soundness and a clear understanding of which objective function is being used<sup>1</sup>.

Surface reconstruction is closely related to the feature tracking problem and can in fact be viewed as registering the pixels in the images across all the images, and then reconstructing the surface by simple triangulation. When extracting and matching features across images, as in e.g. [7], there is however, an implicit assumption that what is seen in one image is to some degree the

---

<sup>1</sup> Data fitting problems almost always boil down to implicitly or explicitly minimizing some objective function.

same as in the next. We will here argue that an (view) overlapping constraint of some sort similar to the tracking case is also needed in the variational approach to surface reconstruction. We will do this by demonstrating that such an overlapping constraint the minimum of the variational problem is by no means the sought solution. Following this we propose a remedy addressing this issue. Since it is noted, that the present variational approaches work well the proposed overlapping constraint will have no practical implications on present day variational methods, but will address their theoretical shortcoming.

In relation to previous work it is noted, that the concept of encouraging overlap between the images is incorporated in many two view stereo algorithms as an occlusion cost, and it is e.g. eloquently formulated in [8]. This overlapping constraint has, however, seemingly been lost in the transition to the multiple view case and the corresponding variational formulations.

## 2 Variational Surface Estimation

As mentioned above, several variational formulations for surface estimation have been proposed in the literature. Most of these formulations fit in the general form of:

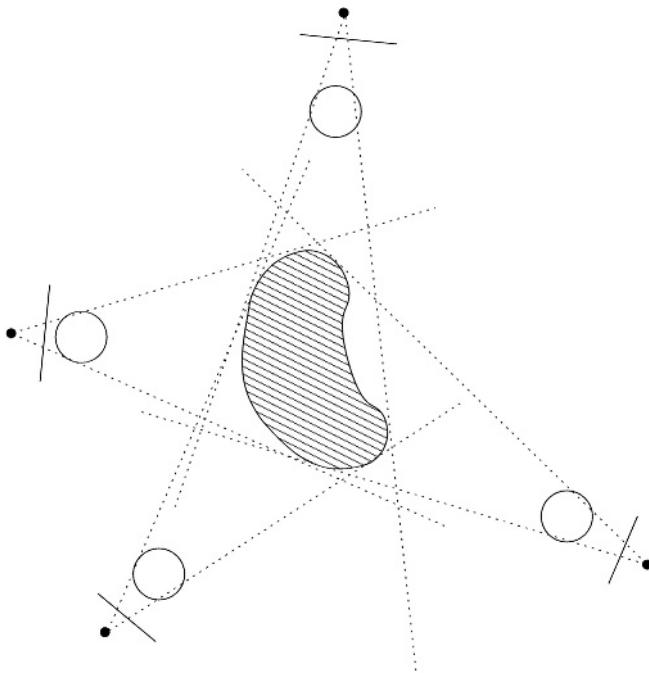
$$E(\Gamma) = \int_{\Gamma} \Phi \, d\sigma + \alpha \int_{\Gamma} d\sigma , \quad (1)$$

where  $\Phi$  is a real-valued functiona, usually defined on the surface  $\Gamma$  (or in a neighborhood of  $\Gamma$ ) which contains a photo consistency term and the last integral is a surface regularization term penalizing surface area. The constant  $\alpha$  determines the weight of the regularizing term. For some methods the regularization is inherent in  $\Phi$  but this does not affect the analysis below. The reader should note, that this formulation, forming the basis for the rest of this paper, is of specific choices for the function  $\Phi$ , the surface representation and the numerical scheme used.

## 3 Counter Examples

The above mentioned methods for surface reconstruction, using (1), have proven themselves by yielding good results in practice. This is usually to a large extent due to a sufficiently good initialization, in that the minimum of (1) found is a local minimum, as described in the following.

Considering Figure 1, it is seen that there exist surfaces,  $\Gamma$ , which have a lower value for (1), than the 'true' underlying surface, but which has no resemblance to this true underlying surface. These undesired surfaces are constructed by placing e.g. small balls in front of each camera covering its field of view. This ensures that each camera exclusively views its part of the surface. The balls or surface area can furthermore be made arbitrary small while moving the balls closer to the camera. This construction for  $\Gamma$  ensures that the first photo consistency part of (1) is zero, in that a perfect match is achievable since . . . . .



**Fig. 1.** A stylized surface reconstruction problem, where the 4 cameras are denoted by their image plane and focal point. The true surface is depicted as the shaded area in the center of the figure and a surface,  $\Gamma$ , giving a better fit to (1) is represented by the circles in front of the cameras. The dotted lines are included to demonstrate that the circles block the cameras field of view

... . Secondly the last part of (1) can be made arbitrarily small. Note that it is implicitly assumed, that the surface area is above zero, in that this is a trivial and again undesirable minimum to (1) which is always present in these formulations.

As such it is demonstrated that the 'true' underlying surface will ... yield a minimum to (1) — since image noise is assumed. Furthermore, minima exist which are not the desired solutions to the problem at hand. In effect (1) does not model the problem faithfully. Note, that these undesirable minima to (1) arise by grossly ignoring the overlapping constraint, and as such this suggests that this needs to be incorporated in the modelling of the problem.

## 4 Overlapping Constraint

As should be apparent from the argument above, an overlapping constraint needs to be incorporated in the modelling of the surface reconstruction problem and as such also in the corresponding variational formulation. In the following analysis, an additional term  $\xi(\Gamma)$  is proposed which added to (1) does exactly that.

Naturally such an overlapping constraint can be modelled or formulated in a multitude of ways. In developing  $\xi(\Gamma)$  we have chosen to focus on the fact that the above mentioned methods using (1), work well in practice, and as such we want the effect of  $\xi(\Gamma)$  around the 'true' surface to be zero or negligible at best. On the other hand — since we have decided on a additive term — the effect of  $\xi(\Gamma)$  should be prohibitive in cases where nearly no overlap exists between the parts of  $\Gamma$  viewed in the various cameras. Secondly, we also note, that in many surface reconstruction tasks only part of the surface is viewed, as such we do not wish to penalize parts of  $\Gamma$  not being viewed in any camera. Hence we chose to take offset in the ratio between the parts of  $\Gamma$  viewed in and the parts of  $\Gamma$  where an overlap exists and  $\Gamma$  is viewed in.

More formally, define  $\Gamma_{\text{Single}}$  as parts of  $\Gamma$  seen by exactly one camera, and  $\Gamma_{\text{Visible}}$  as parts of  $\Gamma$  visible from any camera. Hence  $\Gamma_{\text{Single}} \subseteq \Gamma_{\text{Visible}} \subseteq \Gamma$ . Introduce  $g(\Gamma) \in [0, 1]$  as the ratio

$$g(\Gamma) = \frac{\int_{\Gamma_{\text{Single}}} d\sigma}{\int_{\Gamma_{\text{Visible}}} d\sigma} = \frac{|\Gamma_{\text{Single}}|}{|\Gamma_{\text{Visible}}|}, \quad (2)$$

where we use the notation  $|\Gamma|$  to mean the area of  $\Gamma$ .

Let  $H$  be a smoothed version of the Heaviside step function, e.g. the  $C^2$  approximation as defined in [9]

$$H(x) = \begin{cases} 1 & x > \gamma \\ 0 & x < -\gamma \\ \frac{1}{2}[1 + \frac{x}{\gamma} + \frac{1}{\pi} \sin(\frac{\pi x}{\gamma})] & |x| \leq \gamma \end{cases}. \quad (3)$$

The function  $\xi(\Gamma)$  is then defined to be

$$\xi(\Gamma) = H(g(\Gamma) - \tau), \quad (4)$$

where  $\tau \in [0, 1]$  is a threshold level. This threshold is used to allow  $\Gamma_{\text{Single}}$  some predefined surface fraction.

All that is needed now is to modify (1) to take this overlapping constraint into account. This can be done by formulating the total energy functional for the problem as

$$E_{\text{Tot}}(\Gamma) = E(\Gamma) + \beta H(g(\Gamma) - \tau), \quad (5)$$

where  $\beta$  is a prohibitive weighting constant. In principle one can choose other functions than "thresholding" with  $H$ , we, however, leave this up to the imagination of the reader.

## 5 The Effect on Surface Motion

Adding the term  $\xi(\Gamma)$  to the functional will in theory affect the surface evolution if one uses an iterative approach such as e.g. gradient descent methods. Whether

it is relevant to take this into account or not we leave for the reader to decide. For the sake of completeness, in this section we briefly derive the influence on the normal velocity in a gradient descent implementation. The Gâteaux derivative,  $d\xi(\Gamma)$ , of  $\xi(\Gamma) = H(g(\Gamma) - \tau)$  is

$$d\xi(\Gamma) = H'(g(\Gamma) - \tau)dg(\Gamma) = H'(g(\Gamma) - \tau)\kappa \frac{(1 - g(\Gamma))}{|\Gamma_{\text{Visible}}|}, \quad (6)$$

where  $\kappa$  is the mean curvature of the surface and  $dg(\Gamma)$  denotes the Gâteaux derivative of  $g(\Gamma)$ . This relation is clear from the chain rule and the fact that the Gâteaux derivative of  $g(\Gamma)$  is

$$dg(\Gamma) = \frac{\kappa_{\text{Single}}|\Gamma_{\text{Visible}}| - |\Gamma_{\text{Single}}|\kappa_{\text{Visible}}}{|\Gamma_{\text{Visible}}|^2} = \kappa \frac{\left(1 - \frac{|\Gamma_{\text{Single}}|}{|\Gamma_{\text{Visible}}|}\right)}{|\Gamma_{\text{Visible}}|} = \kappa \frac{(1 - g(\Gamma))}{|\Gamma_{\text{Visible}}|}, \quad (7)$$

where we have used the product rule and the fact that  $\kappa_{\text{Single}} = \kappa_{\text{Visible}} = \kappa$  on  $\Gamma$  and the well known fact that the Gâteaux derivative of area functionals give the mean curvature, cf. [10]. The result is that if one wants to incorporate this in the gradient descent using any surface representation, an extra term  $H'(g(\Gamma) - \tau)\kappa(1 - g(\Gamma))/|\Gamma_{\text{Visible}}|$  in the normal velocity is needed.

## 6 Discussion and Conclusion

The above mentioned surface reconstruction methods employing a variational formulation, all work well without our proposed overlapping constraint (5). One could then ask oneself why this constraint is relevant, since it seemingly has no practical relevance. Firstly, a correct modelling of the problem is important in its own right, and especially with a variational formulation, which is often used to ensure theoretical soundness of ones method. Secondly in the further development of surface estimation methods the insight accompanying the proposed overlapping constraint is likely to become a real concern. An example of the latter is; that the idea to this constraint came while discussing if a certain surface reconstruction algorithm actually reached the global minimum.

In conclusion, an overlapping constraint has been proposed, which helps ensure that the global minimum to functionals used for surface estimation is in fact what is desired. This constraint also encapsulates a better modelling of the surface reconstruction problem, such that it better reflects our implicit expectations.

## References

1. Faugeras, O., Keriven, R.: Complete dense stereovision using level set methods. Computer Vision - ECCV'98. 5th European Conference on Computer Vision. Proceedings (1998) 379–93 vol.1

2. Faugeras, O., Keriven, R.: Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *Image Processing, IEEE Transactions on* 7 (1998) 336–344
3. Jin, H., Yezzi, A., Soatto, S.: Variational multiframe stereo in the presence of specular reflections. Technical Report TR01-0017, UCLA (2001)
4. Jin, H., Soatto, S., Yezzi, A.: Multi-view stereo beyond lambert. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* 1 (2003) 171–178
5. Yezzi, A., Soatto, S.: Structure from motion for scenes without features. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* 1 (2003) 525–532
6. Yezzi, A., Soatto, S.: Stereoscopic segmentation. *International Journal of Computer Vision* 53 (2003) 31–43
7. Hartley, R.I., Zisserman, A.: *Multiple View Geometry*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK (2000)
8. Cox, I., Higorani, S., S.B.Rao, Maggs, B.: A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding* 63 (1996) 542–67
9. Zhao, H., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *J. Computational Physics* 127 (1996) 179–195
10. do Carmo, M.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall (1976)

# Integration Methods of Model-Free Features for 3D Tracking

Ville Kyrki and Kerstin Schmock

Lappeenranta University of Technology,  
Laboratory of Information Processing,  
P.O.Box 20, 53851 Lappeenranta, Finland  
[kyrki@lut.fi](mailto:kyrki@lut.fi)

**Abstract.** A number of approaches for 3D pose tracking have been recently introduced, most of them utilizing an edge (wireframe) model of the target. However, the use of an edge model has significant problems in complex scenes due to background, occlusions, and multiple responses. Integration of model-free information has been recently proposed to decrease these problems.

In this paper, we propose two integration methods for model-free point features to enhance the robustness and to increase the performance of real-time model-based tracking. The relative pose change between frames is estimated using an optimization approach. This allows the pose change to be integrated very efficiently in a Kalman filter. Our first approach estimates the pose change in a least squares sense while the second one uses M-estimators to decrease the effect of outliers. Experiments are presented which demonstrate that the approaches are superior in performance to earlier approaches.

## 1 Introduction

Real-time tracking of 3D pose is today one of the most important building blocks of computer vision systems in several application areas including augmented reality [1], activity interpretation [2], and robotic manipulation [3]. There is a multitude of approaches as most methods have been designed for a particular application. An important reason for this is that the visual characteristics are different in different applications. For example, some methods are suitable for textured objects [3, 1] while others are better suited for objects of uniform color [4, 2].

In 3D tracking, a 3D geometric model is usually available. Several successful systems for real-time pose tracking using such models have been proposed [4, 5]. However, the model represents just the structure of the object using planes and lines, without the knowledge of surface texture. Because the object is tracked using the edge information, such systems are mainly suited to tracking of objects without texture in a relatively simple background. Otherwise the edges of the object are easily confused with background edges and object texture.

Robustness of tracking can be increased by using multiple cues. Recently, it has been proposed to use automatically generated model-free point features to avoid the problems of the model-based tracking[6]. Thus, several different types of features are used to compensate for each others' weaknesses. A Kalman filter framework can be used in integrating the image-plane point measurements with the model-based 3D pose.

In this paper, we continue examining the integration approach. We present two new integration approaches which aim to increase the robustness and performance. The approaches are based on finding the relative pose change between frames by minimizing a sum of squared errors of the image point measurements. The second approach aims to further increase the robustness by decreasing the effect of possible outliers by using a robust M-estimator. We also experimentally compare the computational complexities and the accuracies of the new approaches to the existing approach.

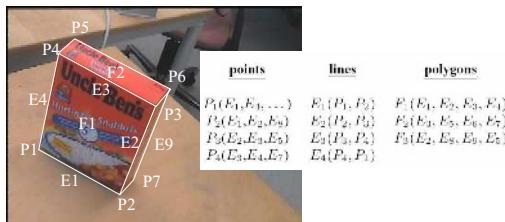
Section 2 presents components of the tracking system including both model-based and model-free trackers. In Sec. 3, we present the Kalman filter model of integration, along with the two new approaches. Experimental evaluation is described in Sec. 4. Finally, in Sec. 5, a short summary and conclusions are given.

## 2 Tracking in 2D and 3D

The tracking approaches are now reviewed independently for model-based and model free cues. A unit focal length is assumed below, i.e., the intrinsic camera calibration has been performed in advance. A more complete description can be found in [6].

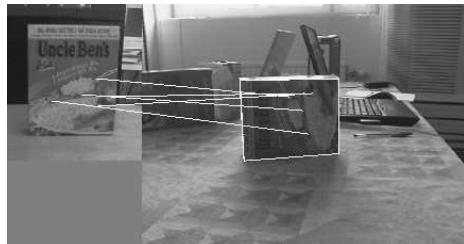
### 2.1 Model-Based 3D Pose Tracking

The model-based system consists of two main components. First, the initialization part, which is only used to find the pose in the first frame, and second, the tracking part, which is run real-time. The tracked objects are modeled using points, lines, and polygons defined in 3D. An example model can be seen in Fig. 1.



**Fig. 1.** Object representation using points, lines, and polygons

In the initialization phase, wide baseline matching is performed in order to initialize the object in a general setting. SIFT features proposed in [7] are extracted off-line from images with the known object pose for each image. During the initialization, SIFT features are extracted from an image and matched with the ones extracted off-line. This is illustrated in Fig. 2. The view for which the number of matches is maximal is then used to estimate the pose of the object. For planar objects, planar homography estimation and decomposition with robust outlier rejection (RANSAC) is used for the pose estimation. This approach is used because for each point it is enough to know that it lies on the same plane, and the exact 3D position is not needed.



**Fig. 2.** Initialization using SIFT point features

The system for model-based pose tracking is based on ideas proposed by Drummond and Cipolla [4]. The basic idea is to estimate the normal flow for a set of points along the edges of the object. The points are distributed with equal image plane intervals along the edges. Lie algebra and Lie group formalism is used to represent the motion of a rigid body such that a six-dimensional Lie group represents the six degrees of freedom of the rigid body motion. The normal flow estimates are then used to find the least squares approximator for the motion by considering the generators of the group at identity. Thus, the motion estimates are good if the change of pose remains small between two frames. The method is very efficient and reasonably accurate if the normal flow estimates are precise.

## 2.2 Model-Free 2D Point Tracking

In addition to the 3D pose, individual interest points are located and tracked in the image. The automatic initialization extracts interest points using Harris corner detection [8]. The current pose estimate is used to only initialize interest points on the frontal plane of the tracked object. New interest points are reinitialized once every 10 frames to guarantee that enough good ones are available at all times. When a new interest point is chosen, its local neighborhood is stored.

The points are tracked in subsequent images by minimizing the sum of squared differences in the RGB values in the local neighborhood. This simple approach is computationally light while still allowing tracking over short time intervals. It should be noted that in contrast to some other proposed methods,

the same interest point does not need to be tracked through the whole sequence. The tracked points are rejected on two conditions: i) if the sum of squared differences grows above a set threshold, or ii) if the point moves out of the frontal plane of the object.

### 3 Kalman Filter Integration

The two types of measurements are integrated in an iterated extended Kalman filter framework. IEKF estimates the state  $\mathbf{x}$  of a system by

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i) + \mathbf{w}_i \quad \mathbf{y}_i = h(\mathbf{x}_i) + \mathbf{v}_i \quad (1)$$

where  $f(\mathbf{x})$  is the system model which describes the time dependencies of the system,  $h(\mathbf{x})$  is the measurement model which links the internal state to measurable quantities,  $\mathbf{y}$  is the measurement, and  $\mathbf{w}$  and  $\mathbf{v}$  are the system noise components modeled as Gaussian random variables. The IEKF estimation consists of two steps: First, the evolution of the system is predicted using the system model. Second, the state is updated using the measurements. For a more thorough description of IEKF, see [6].

The system state consists of the 3D pose of the tracked object,  $\mathbf{x} = (T_X, T_Y, T_Z, \phi_x, \phi_y, \phi_z)^T$ . In this paper, we will concentrate on examining the methods for integrating model free features and thus only a zeroth order moving object system model is considered. The system is thus predicted to remain in the previous state, and the prediction covariance assumes that the object is rotating around its own origin. See [6] for an additional moving camera model.

For integrating the point measurements, three approaches are considered: 1) directly integrating the measurements in the Kalman filter using a suitable measurement model, introduced in [6]; 2) finding the optimal inter-frame transformation using the points, and 3) using M-estimators to find more robust estimates of the pose change. The two latter ones have a linear measurement model. The three approaches have different computational complexities and error characteristics. In all of the approaches, the following simplification is made in contrast to traditional structure-from-motion: Every time a point feature is initialized, its 3D coordinates in the object frame are calculated as the intersection of the line going through the point and camera origin, and the plane defined by the model and the current pose estimate. Thus, the depth and its error are not included in the state, but the associated uncertainty is modeled as part of the measurement noise. This choice allows using the Kalman filter in real-time with a large number of tracked points.

In all of the three approaches, the model-based measurements directly estimate the pose and thus:  $h_0(\mathbf{x}) = \mathbf{x}$ . The measurement errors are assumed to be independent of orientation, therefore the measurement covariance matrix  $\mathbf{S}_0$  can be written

$$\mathbf{S}_0 = \begin{pmatrix} \sigma_t^2 \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \sigma_\phi^2 \mathbf{I}_3 \end{pmatrix} \quad (2)$$

where  $\sigma_t^2$  and  $\sigma_\phi^2$  are the variances of translation and rotation measurements.

### 3.1 Direct Integration

In the direct integration approach, the measurement function is the perspective projection of the known 3D point position  $\mathbf{q}_j$ :

$$\begin{aligned} (X_j \ Y_j \ Z_j)^T &= R(\phi_x, \phi_y, \phi_z) \mathbf{q}_j + t(\mathbf{x}) \\ h_j(\mathbf{x}) &= (X_j/Z_j \ Y_j/Z_j)^T + \mathbf{v}, \end{aligned} \quad (3)$$

where  $R(\cdot)$  is the rotation matrix taking into account the current rotation estimate described by the state, and  $t(\mathbf{x})$  is the translation component of the state. The same form of the function is used for each of the points.

The measurement errors are assumed to be independent with respect to the coordinate axes and thus the measurement covariance matrix can be written  $\mathbf{S}_1 = \sigma_i^2 \mathbf{I}$  where  $\sigma_i^2$  is the image measurement variance. The gradient of the measurement function is calculated analytically from the perspective projection (3) but not shown here for the sake of brevity.

### 3.2 Basic Optimization

The basic optimization approach minimizes the sum of squared errors criterion between image measurements and the initialized 3D points. Thus, the minimized quantity is the 3D-2D projection error  $d(\cdot)$

$$d = \sum_j e_j^2 \quad e_j = \sqrt{\left(x_j - \frac{X_j}{Z_j}\right)^2 + \left(y_j - \frac{Y_j}{Z_j}\right)^2} \quad (4)$$

where  $(x_j, y_j)$  is the measured position of a point, and  $(X_j, Y_j, Z_j)$  are its coordinates in the current camera frame, from Eq. 3. Note that the camera frame coordinates then depend on the current estimated pose.

The optimization is performed using the Polak-Ribiere variant of the conjugate gradient method [9]. Initial search direction is to the negative error function gradient,  $\mathbf{g}_0 = -\nabla d(\mathbf{x}_0)$ . The search direction for successive iterations is determined using

$$\mathbf{g}_{k+1} = -\nabla d(\mathbf{x}_{k+1}) + \gamma_k \mathbf{g}_k \quad (5)$$

where

$$\gamma_k = \frac{(\nabla d(\mathbf{x}_{k+1}) - \nabla d(\mathbf{x}_k)) \cdot \nabla d(\mathbf{x}_{k+1})}{\nabla d(\mathbf{x}_k) \cdot \nabla d(\mathbf{x}_k)} \quad (6)$$

Each line minimization is performed by first bracketing the minimum using golden section bracketing. Then the search for minimum is performed by Brent's method which uses parabolic interpolation and golden section search [10].

### 3.3 M-Estimator Based Optimization

It is well known that a least squares optimization is vulnerable to outliers. To increase the robustness and decrease the effect of outliers, we also present a

method, which employs M-estimators, introduced by Huber [11], for outlier rejection.

In M-estimators, the squared residual is replaced with another function of the residual. The problem can be solved as iteratively reweighted least squares, such that for each measurement, a weight function is determined, standard least squares is performed, and these two steps are iterated. We use the Tukey weight function [12]. Thus, we repeatedly solve the problem

$$\min \sum_j w_{TUK}(e_j) e_j^2 \quad (7)$$

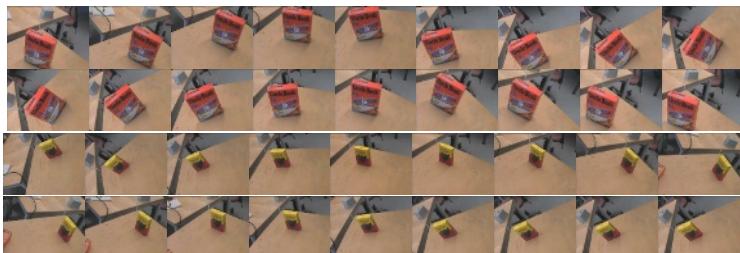
where  $e_j$  is the residual for point  $j$  from (4) and  $w_{TUK}$  is the Tukey weight function

$$w_{TUK}(x) = \begin{cases} (1 - (x/c)^2)^2 & \text{if } |x| \leq c \\ 0 & \text{if } |x| > c \end{cases} \quad (8)$$

where  $c$  is a parameter of the estimator. Tukey weight was chosen because it has been successfully used in similar contexts, e.g., [1]. Each individual least squares minimization is performed using the conjugate gradient method described above. We found out experimentally that two least squares iterations solve the problem to the accuracy of the measurements, and thus the number of iterations was fixed to two. This is mainly because the inter-frame times are short and motions small, and therefore the initial point of the optimization is already very close to the optimum.

## 4 Experimental Comparison

Experiments are now presented to inspect the two new proposed methods. In particular, three characteristics are investigated: i) accuracy, ii) performance, and iii) robustness. The experiments were performed on two recorded sequences to allow repeated tests. Both sequences had a different target object and can be seen in Fig. 3. The lengths of the sequences are 173 (Sequence 1) and 157 seconds (Sequence 2). The sequences were recorded by moving a camera mounted on a

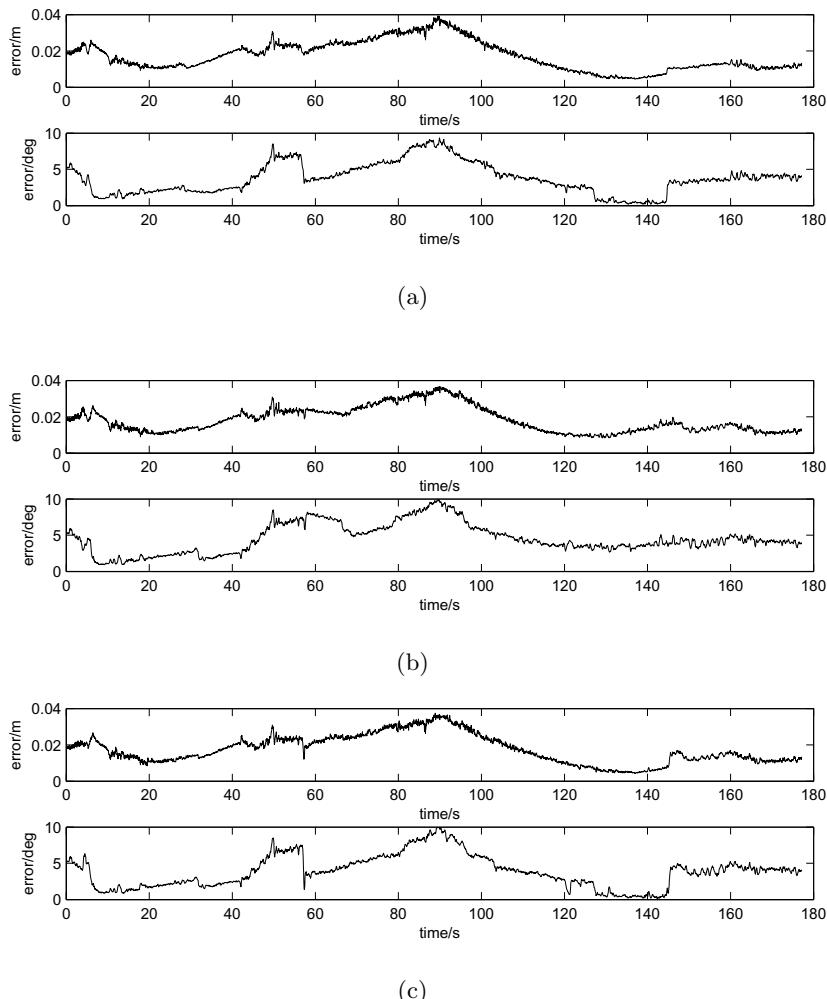


**Fig. 3.** Test sequences: Sequence 1 (top); Sequence 2 (bottom)

robot arm. Ground truth was generated by recording the robot trajectory and determining the hand-eye calibration by the method of Tsai and Lenz [13]. The camera was only coarsely calibrated, the optical axis was assumed to coincide with the center pixel and the manufacturer given focal length was used. A typical number of tracked points was 25 for Sequence 1 and 8 for Sequence 2, but the number changed significantly during both sequences.

#### 4.1 Accuracy

Tracking accuracy was studied for both sequences and the three integration methods described in Sec. 3. Figure 4 shows the error magnitudes for Sequence 1.



**Fig. 4.** Tracking in Sequence 1: (a) direct integration; (b) basic optimization; (c) M-estimator optimization

**Table 1.** Average frame times and frame rates

	Avg. frame time (ms)	Frame rate (Hz)
Direct integration	61.1	16.3
Basic optimization	11.6	86.0
M-estimator opt.	13.1	76.2

All integration methods have similar behavior. Direct integration having slightly lower average error of 1.7 cm in translation and 3.8 degrees in rotation compared to 1.8 cm and 4.5 degrees for basic optimization and 1.8 cm and 3.9 degrees for M-estimator optimization. It can be seen that the basic optimization is slightly inferior to the other two approaches, resulting from the fact that tracked points include some measurement errors, which have unnecessarily large contribution to the trajectory tracked.

For Sequence 2, there were no significant differences, and the average errors were 6.5 cm and 12.2 degrees for both direct integration and basic optimization, and 6.5 cm and 12.0 degrees for M-estimator optimization.

## 4.2 Performance

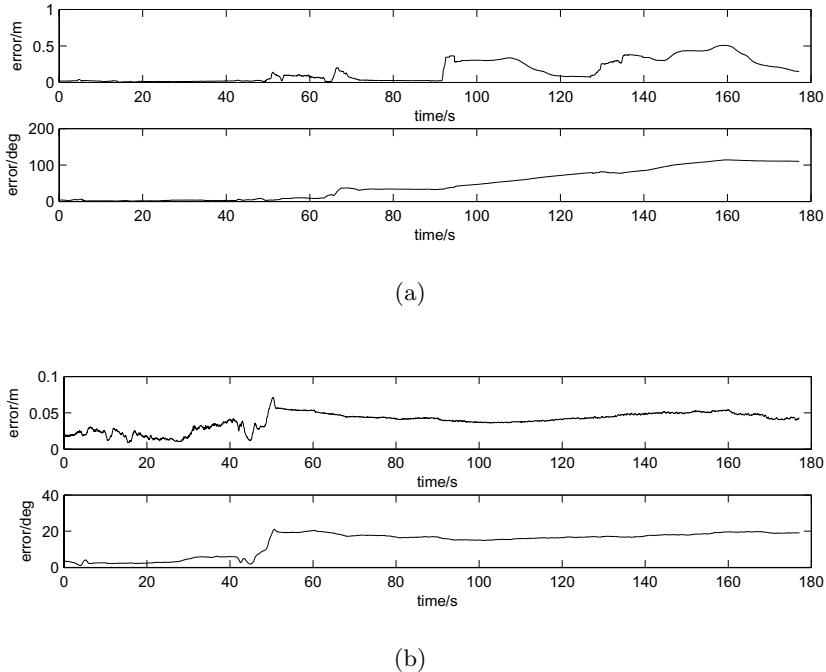
The performance was evaluated by measuring the processor times for the tracking using the three different integration approaches. The total processor time spent during whole Sequence 1 was measured, and then the average frame time and frame rate were calculated. Only Sequence 1 was used because it is longer of the two, in order to minimize the effect of initialization to the computational cost. The tests were run on a 1.7 GHz Pentium M laptop. The results are shown in Table 1.

It is evident from Table 1 that the new methods presented in this paper significantly improve the performance. Basic optimization is slightly faster of the two new methods, but both are clearly suitable for real-time use. The main reason for the improved performance is that the Kalman filter measurements have fewer dimensions. In the direct integration, each tracked point generates two measurements ( $x$ - and  $y$ -coordinates) while in the optimization approaches the model-free measurement is the 6-dimensional pose. For this reason, the Kalman filter update operates on significantly smaller matrices. In addition, the pose measurement is linear to the filter state, and thus no measurement gradient needs to be computed.

The actual runtimes were somewhat larger than the ones above, because the sequences were stored on a hard drive and the disk I/O time is not included in the times given above. For all three methods, the delay of disk I/O was found to be equal, so the results can be compared.

## 4.3 Robustness

The robustness of the two optimization based pose tracking methods was inspected by ignoring totally the model-based cue. The trajectories without model-



**Fig. 5.** Drift in model-free tracking: (a) Basic optimization; (b) M-estimator optimization

based tracking are shown in Fig. 5. Note also the different scaling in the vertical axis in the two graphs. In this case, the tracking is prone to drift. It can be seen in the figure that both methods start to drift after 50 seconds. However, the basic optimization approach soon loses the object position entirely, while the M-estimator approach is capable to keep approximate track of the object position through the whole sequence. The small errors in the translation also indicate that the object position in the image can be tracked. M-estimator approach performs significantly better with 3.9 cm average translation error and 13.8 degree rotation error compared to both direct integration (25 cm/25 degrees) and basic optimization (15 cm/46 degrees). This is explained by the fact that because model-based correction is not made, some tracked points fall outside the object. Therefore the basic optimization approach fails because of the outliers, while the M-estimator approach rejects the outliers.

## 5 Conclusion

Pose tracking systems based on edge models have significant problems in complex scenes due to background, occlusions, and multiple responses. To cope with the problems above, we have presented in this paper two methods of integrating

information from model-free point features in 3D tracking. Both methods use a Kalman filter framework for integrating the features.

Experimental results present an evaluation of the methods with a comparison to the earlier direct integration method. The accuracy of the new methods was comparable to the earlier one. However, computationally the new methods were found to outperform the old one significantly, making video rate operation possible. Finally, the robustness of the methods was evaluated in model-free tracking, which showed that the optimization using M-estimators was clearly more robust against outlier measurements.

Future work will consider tracking more complex geometric shapes, not consisting of planar patches. Also, to improve the tracking against occlusions, we plan to improve the rejection mechanism of the image plane trackers by discarding trackers incompatible with the detected motion.

## References

1. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. *IEEE Trans PAMI* **26** (2004) 1385–1391
2. Vincze, M., Ayromlou, M., Ponweiser, M., Zillich, M.: Edge projected integration of image and model cues for robust model-based object tracking. *Int J of Robotics Research* (2001)
3. Taylor, G., Kleeman, L.: Fusion of multimodal visual cues for model-based object tracking. In: Australasian Conf. on Robotics and Automation, Brisbane, Australia (2003)
4. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. *IEEE Trans. PAMI* **24** (2002) 932–946
5. Wunsch, P., Hirzinger, G.: Real-time visual tracking of 3-d objects with dynamic handling of occlusion. In: IEEE Int. Conf. on Robotics and Automation, ICRA'97, Albuquerque, New Mexico, USA (1997) 2868–2873
6. Kyrki, V., Krägic, D.: Integration of model-based and model-free cues for visual object tracking in 3D. In: Int Conf on Robotics and Automation, ICRA'05. (2005)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.* **60** (2004) 91–110
8. Harris, C.J., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, Manchester, UK (1988)
9. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++. Cambridge University Press (2002)
10. Brent, R.P.: Algorithms for Minimization without Derivatives. Prentice-Hall (1973)
11. Huber, P.J.: Robust estimation of a location parameter. *Annals of Mathematical Statistics* **35** (1964) 73–101
12. Huber, P.J.: Robust Statistics. Wiley (1981)
13. Tsai, R., Lenz, R.K.: A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robotics and Autom.* **5** (1989) 345–358

# Probabilistic Model-Based Background Subtraction

Volker Krüger<sup>1</sup>, Jakob Anderson<sup>2</sup>, and Thomas Prehn<sup>2</sup>

<sup>1</sup> Aalborg Media Lab, Aalborg University,  
Copenhagen, Lautrupvang 15, 2750 Ballerup

<sup>2</sup> Aalborg University Esbjerg,  
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark

**Abstract.** Usually, background subtraction is approached as a pixel-based process, and the output is (a possibly thresholded) image where each pixel reflects, independent from its neighboring pixels, the likelihood of itself belonging to a foreground object. What is neglected for better output is the correlation between pixels. In this paper we introduce a model-based background subtraction approach which facilitates prior knowledge of pixel correlations for clearer and better results. Model knowledge is being learned from good training video data, the data is stored for fast access in a hierarchical manner. Bayesian propagation over time is used for proper model selection and tracking during model-based background subtraction. Bayes propagation is attractive in our application as it allows to deal with uncertainties during tracking. We have tested our approach on suitable outdoor video data.

## 1 Introduction

Companies and scientists work on vision systems that are expected to work in real-world scenarios. Car companies work, e.g., on road sign and pedestrian detection and due to the threat of terrorism, biometric vision systems and surveillance applications are under development.

All these approaches work well in controlled environments, e.g., attempts to recognize humans by their face and gait has proven to be very successful in a lab environment. However, in uncontrolled environments, such as outdoor scenarios, the approaches disgracefully fail, e.g., the gait recognition drops from 99% to merely 30%. This is mainly due to low quality video data, the often small number of pixels on target and visual distractors such as shadows and strong illumination variations.

What is needed are special feature extraction techniques that are robust to outdoor distortions and that can cope with low-quality video data. One of the most common feature extraction techniques in surveillance applications is background subtraction (BGS) [5, 3, 9, 6]. BGS approaches assume a stable camera. They are able to learn a background as well as possible local image variations of it, thus generating a background model even of non-rigid background objects.

During application the model is compared with novel video images and pixels are marked according to the belief that they are fitting the background model. Generally, BGS methods have the following drawbacks:

1. BGS techniques are able to detect “interesting” images areas, i.e., image areas that are sufficiently different from the learned background model. Thus, BGS approaches are able to detect, e.g., a person and the shadow that he/she throws. However, BGS approaches are not able to distinguish between a foreground object and its shadow.
2. Very often, the same objects causes a different output when the scenario changes: E.g. the BGS output for a person walking on green grass or gray concrete may be different.

In this paper we present a Model-based Background Subtracting (MBGS) method that learns not only a background model but also a foreground model. The “classical” background subtraction detects the region of interests while the foreground models are being applied to the classical BGS output to “clean up” possible noise. To reach a maximum of robustness, we apply statistical propagation techniques to the likelihood measures of the BGS.

Having the recent gait recognition attempts in mind, we have applied the following limitations to our discussion (however, the methodology is general enough that we do not see any limit of generality in the chosen setup):

1. We consider only humans as objects and ignore objects that look different from humans.
2. We limit our discussion to silhouettes of humans as they deliver a fairly clothing-independent description of an individual.

The Model-based Background Subtraction System (MBGS System) consists of a learning part to learn possible foreground objects and a MBGS part, where the output of a classical BGS is verified using the previously trained foreground object knowledge.

To learn and represent the foreground knowledge (here silhouettes of humans) is non-trivial due to the absence of a suitable vector space. One possibility is to describe the data in a hierarchical manner, using a suitable metric and a suitable representation of dynamics between the silhouettes. Since we use the silhouettes as density functions where each pixel describes the likelihood of being either foreground or background, we use the Kullback-Leibler distance to compare the silhouettes, k-means clustering is used for clustering similar ones. Similar approaches for hierarchical contour representation can be found in [4, 14].

In the second part, we again consider the contours as densities over spatial coordinates and use normalized correlation to compute the similarity between the silhouette density and the computed one in the input image. Tracking and silhouette selection is being done using Bayesian propagation over time. It can be applied directly since we are dealing with densities and it has the advantage that it considers the uncertainty in the estimation of the tracking parameters and the silhouette selection. The densities in the Bayesian propagation are approximated

using an enhancement of the well-known Condensation method [7]. A similar enhancement of the Condensation method has been applied in video based face recognition[12].

The remainder of this paper is organized as follows: In Sec. 2 we introduce the learning approaches. The actual BGS method is discussed in Sec. 3. We conclude with experimental results in Sec. 4 and final remarks are in Sec. 5.

## 2 Learning and Representation of Foreground Objects

In order to make use of foreground model knowledge, the MBGS system needs to be able to:

- learn a model representation for possibly a number of different and non-rigid objects from video data and
- quickly access the proper model information during application of the MBGS.

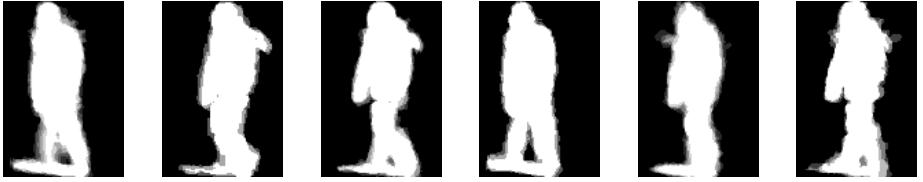
Our main idea is the following: Apply the classical BGS to a scenario that is controlled in a manner that facilitates the learning process. In our case, since we want to learn silhouettes of humans, that only humans are visible in the scene during training and that the background variations are kept as small as possible to minimize distortions. Then, use this video data to learn the proper model knowledge.

After the application of a classical BGS, applying mean-shift tracking [1] allows to extract from the BGS output-data a sequence of small image patches containing, centered, the silhouette. This procedure is the same as the one presented in [10], however, with the difference that here we do not threshold the BGS output but use probabilistic silhouettes (instead of binary ones as in [10]). Thus the silhouettes still contain for each silhouette pixel the belief of being a foreground pixel.

To organize this data we use, similar to [4], a combination of tree structuring and k-means clustering. We use a top down approach: The first level is the root of the hierarchy which contains all the exemplars. Then the second level is constructed by using a the k-means clustering to cluster the exemplars from the root. The third level is constructed by clustering each cluster from the second level, again, using k-means, see Fig. 1 for an example. The k-means clustering uses the Kullback-Leibler divergence measure which measures the similarity between two density functions  $p$  and  $q$ :



**Fig. 1.** An example of our clustering approach: 30 exemplars with  $K=3$  and the algorithm stops after reaching 3 levels



**Fig. 2.** The five images show the cluster centers computed from a training sequence of a single individual

$$KLDist(p, q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} . \quad (1)$$

$KLDist(p, q)$  is non-negative and only zero if  $p$  and  $q$  coincide.

See Fig. 2 for an example clustering of a training sequence of a single individual (as training data, we chose here non-optimal data for better visualization). The tree structure facilitates a fast search of exemplars along the tree vertices, and the cluster centers are either used to apply MBGS on a coarse level or they are used as proto-types, as in [4], to direct the search to a finer level in the hierarchy. Once the tree is constructed, we generate a Markov transition matrix: Assuming that the change over time from one silhouette to a next one can be understood as a first order Markov process, the Markov transition matrix  $M_{ij}^l$  describes the transition probability of silhouette  $s_j$  following after silhouette  $s_i$  at level  $l$  in the hierarchy. During MBGS application particle filtering [8, 11, 2] will be used to find the proper silhouette (see Sec. 3). The propagation of silhouettes over time is non-trivial, as silhouette do not form a vector space. However, what is sufficient is a (not necessarily symmetric) metric space, i.e., given a silhouette  $s_i$ , all silhouettes are needed that are close according to a given (not necessarily symmetric) metric. In the tree structure similar contours are clustered which facilitates the propagation process. The Markov transition matrix  $M_{ij}$  on the other hand describes directly the transition likelihoods between clusters.

### 3 Applying Background Subtraction and Recognizing Foreground Objects

The MBGS system is built as an extension to a pixel based BGS approach. It uses foreground models to define likely correlations between neighbored pixels in the output  $P(\mathbf{x})$  of the BGS application.

Each pixel in the image  $P(\mathbf{x})$  contains a value in the range  $[0, 1]$ , where 1 indicates the highest probability of a pixel being a foreground pixel. A model in the hierarchy can be chosen and deformed according to a 4-D vector

$$\theta = [i, s, x, y], \quad (2)$$

where  $x$  and  $y$  denote the position of the silhouette in the image  $P$ ,  $s$  its scale, and  $i$  is a natural number that refers to a silhouette in the hierarchy.

Presently, the “matching” is done by normalized correlation between a model silhouette density, parameterized according to a deformation vector  $\theta_t$  and the appropriate region of interest in the BGS image  $P_t(\mathbf{x})$ , appropriately normalized.

In order to find at each time-step  $t$  the most likely  $\theta_t$  in the image  $P_t(\mathbf{x})$ , we use Bayesian propagation over time

$$\begin{aligned} p(\theta_t | P_1, P_2, \dots, P_t) &\equiv p_t(\alpha_t, i_t) \\ &= \sum_{i_{t-1}} \int_{\alpha_{t-1}} p(P_t | \alpha_t, i_t) \\ &\quad p(\alpha_t, i_t | \alpha_{t-1}, i_{t-1}) p_{t-1}(\alpha_{t-1}, i_{t-1}) \end{aligned} \quad (3)$$

with  $\alpha_t = [s, x, y]_t$ ;  $P_t$  denotes the probability images while  $p$  denotes density functions. In order to approximate the posteriori density  $p(\theta_t | P_1, P_2, \dots, P_t)$  we use sequential importance sampling (SIS) [2, 7, 11, 13].

Bayesian propagation over time allows us to take into account the uncertainty in the estimated parameters. The uncertainty can be made explicit by computing the entropy of  $p$ .

Monte Carlo methods, like the SIS, use random samples for the approximation of a density function.

Our MBGS system uses separate sample sets for each object in the input image. A new sample set is constructed every time a new object in the video image matches sufficiently well any of the exemplars in the exemplar database.

As the diffusion density  $p(\alpha_t, i_t | \alpha_{t-1}, i_{t-1})$  in Eq. 3 we use the Brownian motion model due to the absence of a better one. For the propagation of the position and scale parameters,  $x$ ,  $y$ , and  $s$ , this is straight forward. For the detection and the propagation of the silhouette we use the following strategy: The likelihood for selecting a silhouette from a certain silhouette cluster in the hierarchy is computed from the Markov transition matrix  $M$  by marginalizing over the silhouettes in that particular cluster. Within a cluster, the new silhouette is then chosen randomly. The reason for this is that since our training data is too little so that the Markov transition matrix  $M$  appeared to be specific to the training videos.

The basic strategy to find the proper silhouette is similar to [12] where the authors find the right identity of a person in a video. In [12], the identity does not change, and the fact that all the particles in the SIS particle filter converge to the same identity is wanted. However, in our case, once all particles have converged to one silhouette, the system would never be able to change anymore to a different silhouette. The diffusion strategy has to assure that enough particles converge to the correct silhouette while at the same time they have the chance to converge to a different one when the shape in the input image changes.

## 4 Experiments

In this section we present qualitative and quantitative results obtained from experiments with our MBGS implementation. The experiments clearly show the

potential of an effective MBGS approach. The purpose of the experiments was to verify that the drawbacks of the classical BGS approach, which were mentioned in section 1, can be remedied with MBGS. More specifically the MBGS system verifies the following:

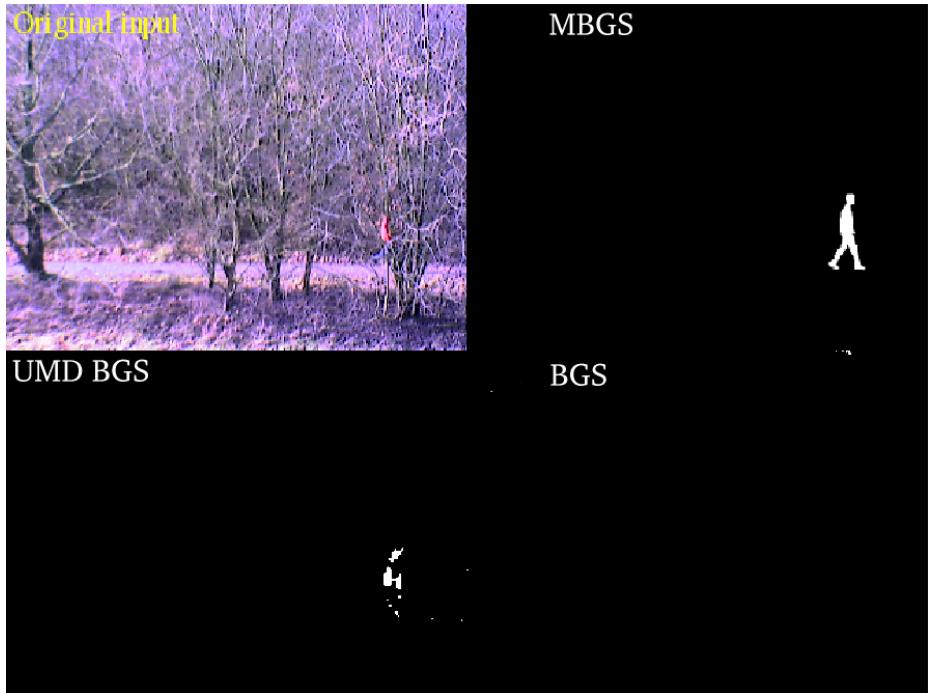
1. Because shadows are not part of the model information provided, these will be classified as background by the implemented MBGS approach. In fact, most non-model object types will be classified as background, and therefore MBGS allows for effective object type filtering.
2. The output presented from the MBGS does not vary, even when the scenario changes significantly. If a model is presented as output, it is always presented intact. The object behind a silhouette is therefore always determinable.

Qualitative verification is done by comparing our MBGS system (top right in the Figs.3-6) with two previously developed pixel-based BGS approaches: One is the non-parametric approach developed at Maryland (UMD BGS, bottom left in the Figs.) [3]. The other BGS, which utilizes a recursive image noise filtering technique, has been developed along with this work.

Figure 3 shows a scenario, with a pedestrian walking behind trees, thereby at times being occluded. The output of two pixel-based approaches is shown in the lower part of the figure. Notice that the shadow cast by the pedestrian is



**Fig. 3.** BGS approach comparison of cases of heavy shadow



**Fig. 4.** BGS approach comparison of a case of heavily occlusion

classified as foreground by these pixel-based BGS approaches. Since the MBGS system operates by inserting a model as foreground, this problem is effectively resolved. Figure 4 shows the same scenario, in a frame where the pedestrian is heavily occluded. The occlusion causes the pedestrian to more or less disappear with pixel based approaches. This happens because the occlusion divides the pedestrian contour into separate smaller parts, which are then removed by the subsequently applied morphological image filters (see for details [3]). The scenario presented in figure 5 shows two pedestrians walking towards each other, thereby crossing behind lamp posts and a statue. When processing this scenario, a combination of image filtering and the background variation, renders the silhouettes of the pixel-based approaches unidentifiable. Also both pixel-based approaches severely distorts the contours of the pedestrians. By only inspecting the pixel-based results, it is hard to tell that the foreground objects are actually pedestrians.

A further example can be seen in Fig. 6.A pedestrian is passed by a car. Still, the MBGS system is able to compute a sensible output due to its model knowledge while the pixel-based BGS techniques fail.

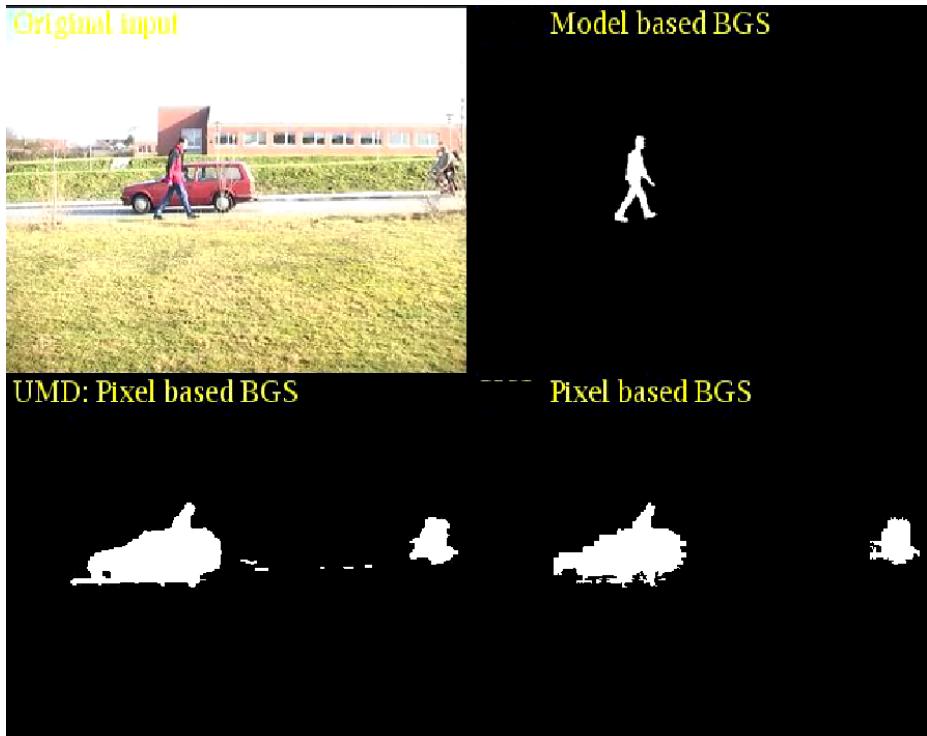
In a quantitative evaluation we have investigated the correctness of the particle method in matching the correct contour. In particular, the experiments verify whether the contour selection strategy of applying a Markov transition



**Fig. 5.** BGS approach comparison in a situation of low contrast

matrix to choose between silhouette clusters, is suitable. When the MBGS is started, the particles are evenly distributed and the system needed usually 20–50 frames to find a sufficiently good approximation of the true density. Then, the selected contour is rather random. After 50 frames, the contour with the maximum likelihood is the correct one in  $\approx 98\%$  of the cases. In  $\approx 20\%$  of the cases the ML contour was incorrect when e.g. a bush was largely occluding the legs. However, recovery time was within 5 frames. In case of partial occlusion of the body through, e.g. small trees, reliability degraded between 1% (slight occlusion) to 10% (considerable occlusion). The contour was incorrect in  $\approx 59\%$  of the cases where the individual was largely occluded, e.g. by a big car or bus. In the videos the individual was in average 70 px. high. Reliability increased considerably with more pixels on target.

The system has been tested on a 2 GHz Pentium under Linux. In videos of size  $320 \times 240$  pixels with only a single person to track, the system runs, with 350 particles, with  $\approx 50$  ms/frame:  $\approx 25$  ms/frame were used by the classical BGS,  $\approx 25$  ms/frame were used by the matching.



**Fig. 6.** This figure shows the output of the MGBS when a pedestrian is passed by a car. Compare the to right image (MBGS) with the pixel-based BGS outputs at the bottom

## 5 Conclusion

The presented model-based background subtraction system combines the classical background subtraction with model knowledge of foreground objects. The application of model knowledge is not applied on a binary BGS image but on the “likelihood image”, i.e. an image where each pixel value represents a confidence of belonging either to the foreground or background. This approach considerably increases robustness as these likelihoods can also be understood as uncertainties which is exploited for the tracking and silhouette selection process. Also, the propagation of densities prevents the need of selecting thresholds (e.g. for binarization of the image  $P$ ) or of maximization. Thresholds are only used for visualization purposes and otherwise for the detection of a new human in the field of view.

In the above application we have chosen silhouettes of humans, but we believe that this choice is without limit of generality since even different object types fit into the tree structure.

The presented experiments were carried out with only a single individual in the database. We have experimented also with different individuals (and thus varying contours), but the output was instable w.f.t. the choice if the individual. This is under further investigation and the use of our approach for gait recognition is future research.

## References

- [1] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, SC, June 13-15, 2000.
- [2] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–209, 2000.
- [3] A. Elgammal and L. Davis. Probabilistic framework for segmenting people under occlusion. In *ICCV, ICCV01*, 2001.
- [4] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Proc. Int. Conf. on Computer Vision*, pages 87–93, Korfu, Greece, 1999.
- [5] I. Haritaoglu, D. Harwood, and L. Davis. W4s: A real-time system for detection and tracking people in 2.5 D. In *Proc. European Conf. on Computer Vision*, Freiburg, Germany, June 1-5, 1998.
- [6] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop*, 1999.
- [7] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 1998.
- [8] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29:5–28, 1998.
- [9] Yuri A. Ivanov, Aaron F. Bobick, and John Liu. Fast lighting independent background subtraction. *Int. J. of Computer Vision*, 37(2):199–207, 2000.
- [10] A. Kale, A. Sundaresan, A.N. Rajagopalan, N. Cuntoor, A.R. Chowdhury, V. Krüger, and R. Chellappa. Identification of humans using gait. *IEEE Trans. Image Processing*, 9:1163–1173, 2004.
- [11] G. Kitagawa. Monta carlo filter and smoother for non-gaussian nonlinear state space models. *J. Computational and Graphical Statistics*, 5:1–25, 1996.
- [12] V. Krueger and S. Zhou. Exemplar-based face recognition from video. In *Proc. European Conf. on Computer Vision*, Copenhagen, Denmark, June 27-31, 2002.
- [13] J.S. Liu and R. Chen. Sequential monte carlo for dynamic systems. *Journal of the American Statistical Association*, 93:1031–1041, 1998.
- [14] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 50–59, Vancouver, Canada, 9-12 July, 2001.

# A Bayesian Approach for Affine Auto-calibration

S.S. Brandt and K. Palander

Helsinki University of Technology,  
Laboratory of Computational Engineering,  
P.O. Box 9203, FI-02015 TKK, Finland

**Abstract.** In this paper, we propose a Bayesian approach for affine auto-calibration. By the Bayesian approach, a posterior distribution for the affine camera parameters can be constructed, where also the prior knowledge can be taken into account. Moreover, due to the linearity of the affine camera model, the structure and translations can be analytically marginalised out from the posterior distribution, if certain prior distributions are assumed. The marginalisation reduces the dimensionality of the problem substantially that makes the MCMC methods better suitable for exploring the posterior of the intrinsic camera parameters. The experiments verify that the proposed approach is a versatile, statistically sound alternative for the existing affine auto-calibration methods.

## 1 Introduction

A central subject in computer vision is the reconstruction of 3D structure of the scene and camera parameters from multiple views that is also known as the structure-from-motion problem. An important related subproblem is the auto-calibration problem, referred to determining the camera parameters without a known calibration object and, starting from [3], it has been widely researched in the computer vision community during the recent years (see [6, 4] for a review). Most of the auto-calibration literature assume the perspective projection camera model, but even auto-calibration methods for wide angle or non-central cameras have been proposed [9, 10].

This paper considers auto-calibration of an affine camera. The affine camera, introduced by [11], is an approximation of the projective camera model which is a generalisation of orthographic, weak perspective and para-perspective camera models and it is a tenable approximation if the viewing distance is large with respect to the depth of the object. The classical papers concerning structure-from-motion using orthographic camera are e.g. [7] about affine reconstruction from two views and [17], where the factorisation approach was introduced. The auto-calibration of an affine camera has been earlier studied in [13]; an overview of the recent contributions on the field is given by [4].

The classical approaches for auto-calibration are based on first computing a (non-unique) projective, or affine in the case of an affine camera, reconstruction

and the corresponding camera projection matrices. The projective, or affine, reconstruction is updated to metric by finding a rectifying transform that is determined by identifying the location of the absolute conic, or its dual, since it is the only invariant conic under rigid motion. Correspondingly, the affine auto-calibration method by Quan [13], is based on the invariance of the camera parameters and it also assumes an affine reconstruction as an input, computed e.g. by factorisation [17] that is updated to metric. Unification of affine and projective auto-calibration approaches has been discussed in [15].

This paper proposes a statistical method for automatic recovery of the affine camera parameters from uncalibrated image sequence. Unlike the auto-calibration algorithm proposed in [13], the proposed method computes the camera parameters directly from point correspondences without an intermediate affine reconstruction. As the solution for the auto-calibration, we shall have a posterior probability distribution for the camera parameters, where also the prior knowledge of the unknown parameters can be taken into account. Moreover, we will show that, because of linearity, the structure and translations can be analytically marginalised, together with the deviation parameter (see [2]), from the posterior distribution, as soon as suitable priors for them are assumed. Due to the marginalisation, the posterior distribution is more suitable to be explored by MCMC methods since dimensionality of the parameter space is notably reduced.

This paper is organised as follows. First the affine camera model and the principles of Bayesian approach are reviewed in Section 2 and 3. Then the Bayesian approach for the auto-calibration is discussed in more detail and our solution for the problem is proposed. Experimental results are reported in Section 5 and conclusions are finally in Section 6.

## 2 Affine Camera Model

The affine camera, introduced in [11], is obtained from the general  $3 \times 4$  projection matrix  $\mathbf{P}$  by setting the last row of to the form  $(0\ 0\ 0\ 1)$ . The affine camera has thus 8 degrees of freedom and the corresponding (homogeneous) projection matrix is

$$\mathbf{P}_A = \begin{pmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

By denoting the top left  $2 \times 3$  submatrix by  $\mathbf{M}_{2 \times 3}$  and the top right 2-vector by  $\mathbf{t}$ , and by using inhomogeneous coordinates, the projection equation is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{M}_{2 \times 3} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}, \quad (1)$$

where  $\mathbf{t}$  is the image of the 3D origin or the image translation.

## 2.1 Parameterisation of the Affine Camera

To obtain a representation for the camera parameters, the inhomogeneous projection matrix  $\mathbf{M}_{2 \times 3}$  can be decomposed into the product  $\mathbf{M}_{2 \times 3} = \mathbf{K}\mathbf{R}$ , where  $\mathbf{K}$  is a  $2 \times 2$  upper-triangular matrix and  $\mathbf{R}$  a  $2 \times 3$  matrix composed of two orthonormal rows, i.e.,

$$\mathbf{M}_{2 \times 3} = \mathbf{K}_{2 \times 2}\mathbf{R}_{2 \times 3} = \begin{pmatrix} \alpha_x & s \\ 0 & \alpha_y \end{pmatrix} \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{pmatrix}. \quad (2)$$

This is known as the RQ decomposition, which can be computed as shown in [6].  $\mathbf{K}_{2 \times 2}$  is the calibration matrix consisting at most three independent camera parameters.  $\mathbf{R}_{2 \times 3}$  contains two rows of the general rotation matrix in the 3D space and has thus 3 independent parameters. The three rotation and the two translation parameters form the set of parameters.

## 2.2 General Model

Another way to parameterise the calibration matrix is

$$\mathbf{K}_{2 \times 2} = k \begin{pmatrix} \xi_a & s_a \\ 0 & 1 \end{pmatrix},$$

where  $k$  is the scaling factor of the camera,  $\xi_a$  is the aspect ratio of the camera and  $s_a$  is the skew of the camera. This is the most general form of the affine camera calibration matrix, of which the special cases, weak perspective, scaled orthographic and orthographic projections, are obtained as follows.

## 2.3 Weak Perspective Projection

By assuming zero skew, we have the weak perspective model

$$\mathbf{K}_{2 \times 2} = k \begin{pmatrix} \xi_a & 0 \\ 0 & 1 \end{pmatrix},$$

where the affine projection matrix has seven degrees of freedom (two intrinsic + five extrinsic parameters).

## 2.4 Scaled Orthographic Projection

The scaled orthographic model is obtained from the weak perspective model by setting equal scales for the  $x$  and  $y$  coordinate axes. The corresponding affine camera has thus six degrees of freedom and the calibration matrix is

$$\mathbf{K}_{2 \times 2} = k \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

## 2.5 Orthographic Projection

By setting the scaling factor  $k$  of the previous calibration matrix equal to unity, the calibration matrix becomes  $2 \times 2$  identity matrix and the orthographic projection model is obtained. This is the most simplified version of the affine camera having five degrees of freedom.

## 3 Bayesian Approach

In general, given a set of observed data samples  $\mathcal{D}$  and a model  $\mathcal{H}$ , suppose there are unknown entities  $\theta$  in the model. The idea of Bayesian approach is to create a probability distribution for all the unknown entities  $\theta$ . The distribution is known as the,

Following [5], suppose with fixed  $\theta$  the data distribution  $p(\mathcal{D}|\theta, \mathcal{H})$ , denoted as the likelihood of the data, is known. Also suppose, that the distribution for  $\theta$  in model  $\mathcal{H}$  is known, often referred to as the,  $p(\theta|\mathcal{H})$ . Writing a product of these two distributions, a joint probability distribution

$$p(\theta, \mathcal{D}|\mathcal{H}) = p(\theta|\mathcal{H})p(\mathcal{D}|\theta, \mathcal{H}) \quad (3)$$

for  $\theta$  and  $\mathcal{D}$  is obtained. Now according to the Bayes' rule, the posterior probability for parameters  $\theta$  in model  $\mathcal{H}$  with the given data samples  $\mathcal{D}$ , is

$$p(\theta|\mathcal{D}, \mathcal{H}) = \frac{p(\theta, \mathcal{D}|\mathcal{H})}{p(\mathcal{D}|\mathcal{H})} = \frac{p(\theta|\mathcal{H})p(\mathcal{D}|\theta, \mathcal{H})}{p(\mathcal{D}|\mathcal{H})}, \quad (4)$$

where  $p(\mathcal{D}|\mathcal{H}) = \int_{\theta} p(\theta|\mathcal{H})p(\mathcal{D}|\theta, \mathcal{H})d\theta$  is a normalising constant and it does not depend on  $\theta$ . Thus, the unnormalised posterior probability distribution for  $\theta$  is

$$p(\theta|\mathcal{D}, \mathcal{H}) \propto p(\theta|\mathcal{H})p(\mathcal{D}|\theta, \mathcal{H}). \quad (5)$$

## 4 Bayesian Affine Auto-Calibration

In this section we thus introduce the auto-calibration problem in the Bayesian framework (Sections 4.1 and 4.2). Our goal is to construct the probability distribution for the intrinsic camera parameters from a set of images. In Section 4.3, we discuss how to compute estimates from the posterior distribution.

### 4.1 Statistical Model for the Data

Now, suppose we have  $N$  uncalibrated images with altogether  $M$  separate points of which we observe projections but all the projections need not to be visible in all the views. Assuming 2D Gaussian noise model for the image point measurements  $\mathbf{m} \equiv \mathcal{D}$ , the joint probability distribution, or the likelihood, for all points  $j$  in

all images  $i$  is written as

$$p(\mathbf{m}|\theta, \mathcal{H}) \propto \prod_{i,j|\delta_{ij}=1} \sigma^{-2} \exp\left\{-\frac{1}{2\sigma^2} (\mathbf{m}_j^i - (\mathbf{M}^i \mathbf{x}_j + \mathbf{t}^i))^T (\mathbf{m}_j^i - (\mathbf{M}^i \mathbf{x}_j + \mathbf{t}^i)) \delta_{ij}\right\}, \quad (6)$$

where  $\delta_{ij} = 1$  if point  $j$  is visible in image  $i$ , otherwise it is zero. The model  $\mathcal{H}$  here refers to the affine projection model (1) equipped with the 2D Gaussian measurement noise model. We here consider the case where the observed data contains distinct point projection coordinates in multiple views and we have established the correspondences between the point projections across the views. Thus, the unknown entities  $\theta$  in the likelihood function are here both the camera parameters (both intrinsic and extrinsic) and the 3D reconstructions of the points, as well as the noise deviation  $\sigma$ .

## 4.2 Posterior Distribution for the Camera Parameters

Let our selected prior distribution for the unknown parameters  $\theta$  be separable such that the posterior distribution is of the form

$$p(\theta|\mathbf{m}, \mathcal{H}) \propto p(\sigma|\mathcal{H})p(\mathbf{K}|\mathcal{H})p(\mathbf{R}|\mathcal{H})p(\mathbf{t}|\mathcal{H})p(\mathbf{x}|\mathcal{H})p(\mathbf{m}|\theta, \mathcal{H}). \quad (7)$$

In pure affine auto-calibration, we are not interested in the structure parameters  $\mathbf{x}$ , the camera extrinsic, and motion parameters  $\mathbf{R}, \mathbf{t}$ , nor the noise deviation  $\sigma$ . In Bayesian framework, our complete ignorance of those parameters may be represented by marginalising them out from the posterior distribution. That is

$$p(\mathbf{K}|\mathbf{m}, \mathcal{H}) = \iiint p(\theta|\mathbf{m}, \mathcal{H}) d\mathbf{x} d\mathbf{R} dt d\sigma, \quad (8)$$

where we have a posterior distribution over the intrinsic camera parameters. The posterior (8) represents a complete solution for the auto-calibration problem. Unfortunately, it is improbable that the integral would have an analytical solution. However, we observe that the affine camera model is, with respect to the structure and translation parameters, hence, the following partial integral can be computed analytically

$$p(\mathbf{K}, \mathbf{R}|\mathbf{m}, \mathcal{H}) = \iiint p(\theta|\mathbf{m}, \mathcal{H}) d\mathbf{x} dt d\sigma, \quad (9)$$

provided that we choose suitable priors for the marginalised variables, and fix the gauge freedom properly. For the deviation  $\sigma$ , we choose the Jeffrey's prior  $p(\sigma|\mathcal{H}) = \frac{1}{\sigma}$ . Furthermore, if we select, for instance, uniform or Gaussian priors for  $\mathbf{x}$  and  $\mathbf{t}$ , the (9) can be generally integrated analytically, excluding the non-degenerate motion or structure cases and incompletely fixed gauge freedom. Note however that the Bayesian approach could generally also handle degenerate cases by defining suitable priors or developing a model selection framework but they are beyond the scope of this paper.

Before we are able to integrate (9), we must fix the gauge freedom, i.e., we must specify in which coordinate system we represent our variables. In structure-from-motion, there is no absolute reference coordinate system, so we must define where and to which orientation we set the origin of our 3D coordinate system and also to which value we should define the global scale. Uncertainties are measured with respect to this selection, which should be taken kept in mind when interpreting the results. For algebraic simplicity, we select “a trivial gauge” by setting the 3D origin to the first point. Likewise, we select the first view as the reference view so that the rotations of the other cameras are measured with respect to the first camera, and, we set the scale of the first camera to unity.

In the case of uniform priors for the structure and translations, and after fixing the gauge as discussed, a tedious algebraic manipulation reveals that

$$p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) \propto p(\mathbf{K} | \mathcal{H}) p(\mathbf{R} | \mathcal{H}) \left( \prod_{j=2}^M (\det \mathbf{C}_j)^{-\frac{1}{2}} \right) (\det \mathbf{C})^{-\frac{1}{2}} \\ \times \left( \sum_{i,k} a_{ik} - \mathbf{b}^T \mathcal{C}^{-1} \mathbf{b} \right)^{\frac{3(M-1)+2(N-\sum_{i,j} \delta_{ij})}{2}}, \quad (10)$$

where

$$\mathbf{C}_j = \sum_{i=1}^N \mathbf{M}^{i^T} \mathbf{M}^i \delta_{ij}, \quad a_{ik} = \sum_{j=1}^M \mathbf{m}_j^{i^T} \left( \delta(i-k) \mathbf{I} - \delta(j-1) \mathbf{M}^i \mathbf{C}_j^{-1} \mathbf{M}^{k^T} \right) \mathbf{m}_j^k \delta_{ij} \delta_{kj}, \\ \mathcal{C} = \begin{pmatrix} \mathbf{C}_{11} & \dots & \mathbf{C}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{N1} & \dots & \mathbf{C}_{NN} \end{pmatrix}, \quad \mathbf{C}_{ik} = \sum_{j=1}^M \left( \delta(i-k) \mathbf{I} - \delta(j-1) \mathbf{M}^i \mathbf{C}_j^{-1} \mathbf{M}^{k^T} \right) \delta_{ij} \delta_{kj}, \\ \mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \end{pmatrix}, \quad \mathbf{b}_k = \sum_{i=1}^N \sum_{j=1}^M \left( \delta(i-k) \mathbf{I} - \delta(j-1) \mathbf{M}^k \mathbf{C}_j^{-1} \mathbf{M}^{i^T} \right) \mathbf{m}_j^i \delta_{ij} \delta_{kj}.$$

The posterior probability for the camera parameters will obtained by numerically evaluating

$$p(\mathbf{K} | \mathbf{m}, \mathcal{H}) = \int p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) d\mathbf{R}, \quad (11)$$

where the integration is performed over the rotation parameters. The computation of estimates from this distribution is described in the following section.

### 4.3 Computing Estimates from the Posterior

The posterior distribution of the intrinsic camera parameters is a complete solution for the affine auto-calibration problem; from it we may compute  $\dots, \dots, \dots$ , i.e., a set of camera parameters that we could regard as the most descriptive candidates. We would also like to know the  $\dots, \dots, \dots$  of our estimates. At this step we have two obvious choices to proceed: we might compute the estimate

that maximises the posterior (the MAP estimate) or the estimate that represents the conditional mean (the CM estimate) of the posterior distribution.

#### 4.4 MAP Estimate

Computation of the maximum a posterior estimate is an optimisation problem. Let  $\theta_{\mathbf{K}}$  represent the set of unknown intrinsic parameters of the affine cameras we have set unknown. Note that some parameters must be set known or invariant across the views, since otherwise the auto-calibration problem is not well defined. Since the rotations can not be marginalised from the posterior, we compute the MAP estimate of the joint posterior (10), i.e., we make the approximation

$$\hat{\theta}_{\mathbf{K}, \text{MAP}} = \arg \max_{\theta_{\mathbf{K}}} p(\mathbf{K} | \mathbf{m}, \mathcal{H}) \approx \arg \max_{\theta_{\mathbf{K}}} \left( \max_{\theta_{\mathbf{R}} | \theta_{\mathbf{K}}} p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) \right), \quad (12)$$

where the maximum can be numerically computed by conventional nonlinear optimisation tools such as Levenberg–Marquardt. The uncertainty can be characterised by the covariance matrix of the estimated parameters. Here we approximate the posterior covariance by first making a Gaussian posterior approximation  $p_{\text{approx}}(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H})$  around the maximum of  $p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H})$  by the Laplace’s method [8]. Then we are able to marginalise the rotation parameters analytically, i.e., we compute

$$\begin{aligned} p(\mathbf{K} | \mathbf{m}, \mathcal{H}) &\approx \int p_{\text{approx}}(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) d\mathbf{R} \\ &\propto \exp \left( -\frac{1}{2} \left( \theta_{\mathbf{K}} - \hat{\theta}_{\mathbf{K}, \text{MAP}} \right)^T \tilde{\mathbf{C}}_{\theta_{\mathbf{K}}}^{-1} \left( \theta_{\mathbf{K}} - \hat{\theta}_{\mathbf{K}, \text{MAP}} \right) \right), \end{aligned} \quad (13)$$

which is a Gaussian approximation over the intrinsic parameters only, hence, we can use the covariance matrix of this Gaussian approximation as our covariance matrix estimate for the intrinsic camera parameters.

#### 4.5 CM Estimate

Another alternative for characterising the posterior, is to compute the conditional mean estimate, conditioned on the data, defined as

$$\hat{\theta}_{\mathbf{K}, \text{CM}} = \mathbb{E}\{\theta_{\mathbf{K}} | \mathbf{m}, \mathcal{H}\} = \iint \theta_{\mathbf{K}} p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) d\theta_{\mathbf{K}} d\theta_{\mathbf{R}}, \quad (14)$$

which is an integration problem. Likewise, to characterise the uncertainty we would like to compute the conditional covariance  $\mathbf{C}_{\theta_{\mathbf{K}}}$ , defined as

$$\text{Cov}\{\theta_{\mathbf{K}} | \mathbf{m}, \mathcal{H}\} = \iint (\theta_{\mathbf{K}} - \hat{\theta}_{\mathbf{K}, \text{CM}}) (\theta_{\mathbf{K}} - \hat{\theta}_{\mathbf{K}, \text{CM}})^T p(\mathbf{K}, \mathbf{R} | \mathbf{m}, \mathcal{H}) d\theta_{\mathbf{K}} d\theta_{\mathbf{R}}. \quad (15)$$

Though the integrals cannot be calculated in closed form, they can be approximately evaluated by Markov chain Monte Carlo methods (see for instance

[16] or [5]). By obtaining a reasonable amount of samples from the distribution, the conditional mean can be approximated as

$$\hat{\theta}_{\mathbf{K}, \text{CM}} \approx \frac{1}{n} \sum_{j=1}^n \theta_{\mathbf{K}}^{(j)}, \quad (16)$$

and the conditional covariance as

$$\mathbf{C}_{\theta_{\mathbf{K}}} \approx \frac{1}{n - \dim(\theta_{\mathbf{K}})} \sum_{j=1}^n \left( \theta_{\mathbf{K}}^{(j)} - \hat{\theta}_{\mathbf{K}, \text{CM}} \right) \left( \theta_{\mathbf{K}}^{(j)} - \hat{\theta}_{\mathbf{K}, \text{CM}} \right)^T, \quad (17)$$

where  $\theta_{\mathbf{K}}^{(1)} \dots \theta_{\mathbf{K}}^{(n)}$  are assumed to be effectively independent samples obtained by MCMC simulation. Here the numerical computation of the CM estimate is rather convenient for the MCMC methods since the dimensionality of the parameter space is equal to  $\dim \theta_{\mathbf{K}} + \dim \theta_{\mathbf{R}}$  which is typically substantially lower than the dimensionality of the original posterior (7).

## 5 Experiments

We experimented our auto-calibration method with two image sequences. The first set was a synthetic house sequence of 9 images, where the simulated images obey the perspective projection model. We identified and manually tracked 81 points from these images, so that the tracked points contained also missing data. Since the perspective effects were relatively strong in the projections, the role of this data set in our affine auto-calibration experiments is only illustrative.

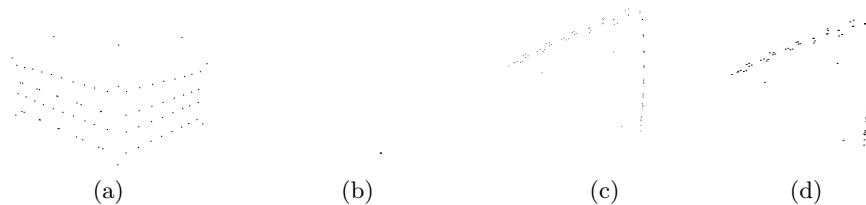
For auto-calibration, we then used the scaled orthographic camera model and computed the MAP and CM estimates for the affine camera parameters from (12) and (16), respectively. For both rotation and scale parameters, we used the uniform prior distribution. The CM estimate was computed using the Metropolis–Hastings algorithm using a Gaussian proposal distribution. Due to gauge fixing, the scale of the first image was set to unity and similarly its rotation matrix was set to the identity matrix, and there were thus 8 scale parameters and  $3 \times 8 = 24$  rotation parameters to be estimated, or, 32 parameters in total. The starting points for the optimisation and sampling were randomly generated.

To illustrate the result, we then recovered the 3D structure and translations,  $\dots$  the found estimates for the camera parameter values; the conditional estimates for structure and translations are computed as proposed in [1]. The MAP and CM estimates showed somewhat similar reconstructions here, as Fig. 1 illustrates. As it can be seen, the perspective effects can not be explained by the affine camera model due to which the angle between the two orthogonal walls is not straight as it was in the original simulated data.

As real data, we experimented the hotel image set, available at the CMU VASC image database<sup>1</sup>, that was also used in [12] and [14] (Fig. 2). In our

---

<sup>1</sup> <http://vasc.ri.cmu.edu/idb/>



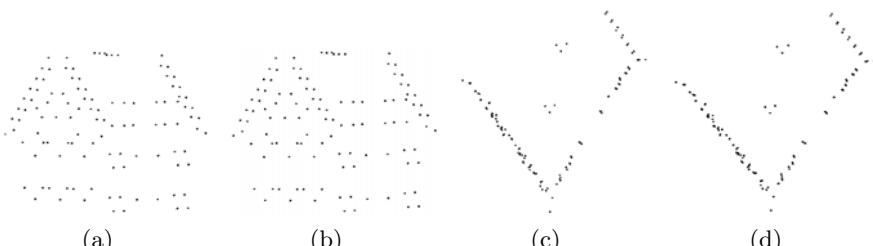
**Fig. 1.** Synthetic house data reconstructed after the recovery of the scaled orthographic camera parameters: (a) MAP estimate, front view; (b) CM estimate, front view; (c) MAP estimate, bottom view. (d) CM estimate, bottom view. The angle between the two wall planes is not straight, because the affine camera model is not able to model the perspective effects of the original data



**Fig. 2.** Two examples from the Hotel model images. The set set is from CMU VASC image database

experiments, we used 10 of the 182 images, where 111 points were identified and tracked manually. All points were visible in all images, i.e., there was no missing data; the initial affine reconstruction in Quan's method [14] was computed using the factorisation algorithm [17]. For this set, we selected the weak perspective projection model, hence, there were 9 scales, 10 aspect ratios, plus 27 rotation parameters to be identified, i.e., 46 parameters in total.

The affine auto-calibration results for the Hotel set are illustrated in Fig. 3. Again, the MAP and CM estimates implied similar reconstruction, so only the



**Fig. 3.** Reconstruction of the Hotel set after determining the camera parameters by auto-calibration. (a) Quan auto-calibration, front view; (b) the MAP estimate, front view; (c) Quan auto-calibration, top view; (d) the MAP estimate, top view

reconstruction corresponding the MAP estimate and the reconstruction corresponding to Quan's method are shown there. The MAP (and CM) estimate seems to be a bit more accurate than the Quan auto-calibration as the corner seen in the top view is more right.

## 6 Summary and Conclusions

This paper proposed a new statistical method auto-calibration from multiple images assuming an affine camera model. Like the earlier methods, the methods is based on the invariance of camera parameters, i.e., under the selected assumption of invariant parameters over the views, the affine camera parameters can be determined. As we used the Bayesian, statistical setting, available prior information of the camera parameters can be directly incorporated in the method. Moreover, the Bayesian approach for auto-calibration is well suitable for an affine camera model, since the uninteresting structure and translation variables, from the auto-calibration point of view, can be analytically marginalised from the posterior distribution under certain prior distribution assumptions. The marginalisation makes the MCMC methods better suitable for exploring the posterior distribution of unknown camera parameters as the large dimensionality of the structure-from-motion problem is substantially reduced. The experimental results validate that the proposed method is statistically sound method for affine auto-calibration, as it minimises statistical error in contrast to the earlier methods that minimise an algebraic error.

## References

1. S. S. Brandt. Conditional solutions for the affine reconstruction of  $N$  views. *Image and Vision Computing*, 2005. In press.
2. G. L. Bretthorst. *Bayesian Spectrum Analysis and Parameter Estimation*, volume 48 of *Lecture Notes in Statistics*. Springer, 1988.
3. O. Faugeras, Q. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proc. ECCV*, pages 321–334, 1992.
4. O. Faugeras and Q.-T. Luong. *Geometry of Multiple Images*. The MIT Press, 2001.
5. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.
6. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
7. J. Koenderink and A. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.
8. D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
9. B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *Proc. CVPR*, volume 1, pages 485–490, 2003.
10. B. Micusik and T. Pajdla. Autocalibration & 3d reconstruction with non-central catadioptric cameras. In *Proc. CVPR*, volume 1, pages 58–65, 2004.

11. J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, Massachusetts, 1992.
12. C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *Proc. ECCV*, pages 97–108, 1994.
13. L. Quan. Self-calibration of an affine camera from multiple views. *Int. J. Comput. Vis.*, 19(1):93–105, 1996.
14. L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *Proc. CVPR*, pages 803–808, 1996.
15. L. Quan and B. Triggs. A unification of autocalibration methods. In *Proc. ACCV*, 2000.
16. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 1999.
17. C. Tomasi and T. Kanade. Shape and motion form image streams under orthography: A factorization approach. *Int. J. Comput. Vis.*, 9(2):137–154, 1992.

# Shape-Based Co-occurrence Matrices for Defect Classification

Rami Rautkorpi and Jukka Iivarinen

Helsinki University of Technology,  
Laboratory of Computer and Information Science,  
P.O. Box 5400, FIN-02015 HUT, Finland  
`{rami.rautkorpi, jukka.iivarinen}@hut.fi`

**Abstract.** This paper discusses two statistical shape descriptors, the Edge Co-occurrence Matrix (ECM) and the Contour Co-occurrence Matrix (CCM), and their use in surface defect classification. Experiments are run on two image databases, one containing metal surface defects and the other paper surface defects. The extraction of Haralick features from the matrices is considered. The descriptors are compared to other shape descriptors from e.g. the MPEG-7 standard. The results show that the ECM and the CCM give superior classification accuracies.

## 1 Introduction

Shape features can be divided into two main categories [1]: syntactical and statistical. Syntactical features use structural descriptions that are suitable for regular shapes, such as those of man-made objects, while statistical features are more suitable for irregular, naturally occurring shapes. In surface defect inspection, the defect types can have very similar overall shapes, so classification based on shape depends on the accurate description of the more subtle features in a shape.

Histogram techniques are a simple and efficient way of extracting statistical information. This paper discusses two histogram-based shape descriptors, the Edge Co-occurrence Matrix (ECM) and the Contour Co-occurrence Matrix (CCM) and presents experimental results of the application of these descriptors to surface defect classification. Also discussed is the extraction of Haralick features from the feature matrices in order to acquire a shorter feature vector for more efficient computation. A comparison of classification performance is made between these descriptors and other shape descriptors that utilize edge and contour information.

## 2 Shape Descriptors

### 2.1 Edge Co-occurrence Matrix

The Edge Co-occurrence Matrix (ECM) contains second order statistics on edge features in an image. It was introduced in [2], where early results from this work

were presented. It is similar to the Gray Level Co-occurrence Matrix (GLCM) [3], but instead of a gray level intensity image, it is derived from an edge image, which describes the locations and directions of edges in the original gray-level image. In this respect it is related to the MPEG-7 descriptor Edge Histogram, which contains first order statistics on edge features.

The ECM is calculated as follows. First edge images are formed from the original image by convolving it with the eight Sobel masks [4]. The eight masks are needed to retain information on the direction of intensity change in an edge. The eight edge images from the convolutions are combined into one edge image by selecting for each pixel the direction of the strongest edge. This final edge image is then thresholded, so that it contains only the strongest edge pixels. The threshold value is defined as a percentage of the strongest edge value present in the image.

The ECM is then formed from the pairs of edge pixels separated by a given displacement. Let  $\mathbf{I}$  be a thresholded edge image and let  $\mathbf{d} = (d_x, d_y)$  be a displacement vector. Then the Edge Co-occurrence Matrix  $\mathbf{H}^{ECM}$  is defined as a matrix, where the  $(i, j)$ th element is the number of edge pixels with direction  $i$  separated from an edge pixel with direction  $j$  by the displacement vector  $\mathbf{d}$ ,

$$H_{ij}^{ECM} = \#\{\mathbf{x} \mid I(\mathbf{x}) = i, I(\mathbf{x} + \mathbf{d}) = j\}, \quad (1)$$

where  $\#$  is the number of elements in the set and  $\mathbf{x} = (x, y)$  runs through the edge image  $\mathbf{I}$ . Since the edges were detected in 8 directions, the size of the ECM is  $8 \times 8$ .

Since only the relative distances and directions of pixel pairs are used, the matrix is translation invariant. However, it is not scale or rotation invariant. The sum of the matrix elements is the number of edge pixel pairs found, and even if the matrix is normalized to unit sum, the feature is scale sensitive due to the length of the displacement. However, normalization makes it possible to interpret the matrix as a probability distribution, from which various features can be calculated.

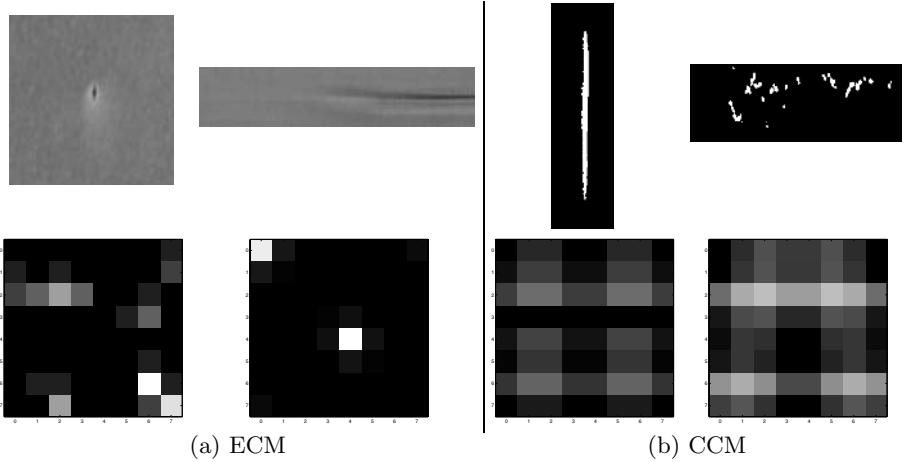
Examples of ECMs calculated from defect images are shown in Figure 1(a).

## 2.2 Contour Co-occurrence Matrix

The Contour Co-occurrence Matrix (CCM), introduced in [5], contains second order statistics on the directionality of the contours of objects in an image. It resembles the ECM and the GLCM, but instead of a two-dimensional image, the co-occurrence information is calculated from the Freeman chain code [6] of the contour of an object. In this regard, it is related to the Chain Code Histogram (CCH) [7] that can be regarded as the CCM's first-order counterpart.

The first step in calculating the CCM of an object is to generate the chain code of its contour. The starting point of the contour is not stored, so the resulting feature descriptor is translation invariant.

The co-occurrence matrix is then formed from the pairs of links separated by a given displacement. Let  $A$  be a chain of length  $n$  and let  $d$  be a displacement, i.e.



**Fig. 1.** Example images of surface defects and the ECMs and CCMs calculated from them

the difference between two chain link indices (not the distance, i.e. the absolute value of the difference). Then the edge co-occurrence matrix  $\mathbf{H}^{CCM}$  is defined as a matrix, where the  $(i, j)$ th element is the number of instances of a link with value  $i$  separated from a link with value  $j$  by the displacement  $d$ ,

$$H_{ij}^{CCM} = \#\{k \mid a_k = i, a_{k+d \pmod n} = j\}, \quad (2)$$

where  $\#$  is the number of elements in the set and  $k$  runs through the values  $0, 1, \dots, n - 1$ . Because the chain is derived from a closed contour, the index  $k$  and displacement  $d$  are summed modulo  $n$ , so that the displacement wraps around the chain's arbitrary starting point. Since the chain code is octal, the size of the CCM is  $8 \times 8$ .

Two basic variations of the CCM may be considered, based on whether the displacement  $d$  is constant over all examined contours (let this be called the CCM1), or dependent on the length of the chain, i.e.  $d = cn$ , where  $c$  is a real number in the range  $[0, 1[$  (let this be called the CCM2). If the sum of the CCM's elements is normalized to unity, the matrix represents the joint probability of the link values  $i$  and  $j$  occurring at link indexes with the difference  $d$ . Thus normalized, the CCM2 becomes scale invariant.

Examples of CCMs calculated from defect images are presented in Figure 1(b).

### 2.3 Other Shape Descriptors

Other shape descriptors considered in this paper are taken from the MPEG-7 standard, formally named “Multimedia Content Description Interface” [8]. They are well standardized descriptors that are used in searching, identifying, filtering and browsing images or video in various applications. The features are:

- Edge Histogram (EH) calculates the amount of vertical, horizontal, 45 degree, 135 degree and non-directional edges in 16 sub-images of the picture, resulting in a total of 80 histogram bins.
- Contour-based Shape (CBS) consists of a set of peak coordinates derived from a Curvature Scale Space (CSS) representation of a contour, and the eccentricities and circularities of the contour and its convex prototype, which is created by repeatedly low-pass filtering the contour.
- Region-based Shape (RBS) utilizes a set of 35 Angular Radial Transform (ART) coefficients that are calculated within a disk centered at the center of the image's Y channel.

In addition to the MPEG-7 shape features we also tested three other shape descriptors that we have used previously for defect image classification and retrieval:

- Simple Edge Histogram (SEH) is similar to its MPEG-7 counterpart, but instead of dividing an image into several sub-images, it is calculated like the ECM, for the whole image.
- Simple Shape Descriptor (SSD) [9] consists of several simple descriptors calculated from an object's contour. The descriptors are convexity, principal axis ratio, compactness, circular variance, elliptic variance, and angle.
- Chain Code Histogram (CCH) [7] is an 8-dimensional histogram calculated from the Freeman chain code of a contour. It is the first-order equivalent of the CCM.

## 3 Experiments

### 3.1 Experimental Setup

Experiments were carried out with two image databases containing defect images, one from a metal web inspection system and the other from a paper web inspection system. All images are grayscale images, supplied with binary mask images containing segmentation information. The contours of the objects were extracted from the segmentation masks using inner boundary tracing with 8-connectivity. The images have different kinds of defects and their sizes vary according to the size of a defect. Classification of defects is based on the cause and type of a defect, and different classes can therefore contain images that are visually dissimilar in many aspects. The paper defect database has 1204 images. They are preclassified into 14 different classes with between 63 and 103 images in all of the classes but one which has only 27 images. The metal defect database has 2004 images. They are preclassified into 14 different classes, with each class containing from 101 up to 165 images. The databases were provided by ABB Oy. More information on these databases can be found e.g. in [10, 11].

All tests of the classification performance of the feature were made with K-Nearest Neighbor leave-one-out cross-validation (KNN-LOOCV), using the Euclidean distance. Each feature vector is in turn selected to be classified, while

the remaining vectors form the training set. The test vector is classified by voting among the  $K$  nearest vectors in the training set, using the ground truth information supplied with the images. All tests here use the value  $K = 5$ . The classification success rate in a given class is the percentage of images belonging to the class that have been classified correctly. The average or overall success rate is the mean of the success rates of the classes weighted by the number of images in the classes.

### 3.2 Preliminary Experiments

Some initial tests were made to determine good values for the parameters of the features. These include the displacements for both features and the edge detection threshold for the ECM. A comparison was made between the two versions of the CCM.

The optimum edge detection threshold for these datasets was found to be approximately 15%, meaning that any detected edge pixels with a strength less than 15% of the strongest edge present in the image were discarded. This value is highly dependent on the nature of the problem and the image generation process, and is not necessarily the optimum value for any other dataset.

The first comparison is between ECMs using a “full circle” arrangement of eight displacement vectors with a distance of one, an asymmetrical “half circle” version of the same with four vectors, a symmetrical arrangement with vectors  $(1, 1)$  and  $(-1, -1)$ , and an asymmetrical version with the vector  $(1, 1)$ . The ECMs were normalized to unit sum. The average classification rates were 59% for the vector  $(1, 1)$ , 52% for the combination  $(1, 1)$  and  $(-1, -1)$ , 59% for the full circle and 51% for the half circle.

The most important thing to notice is that the asymmetrical ECMs give significantly better results than the symmetrical ones. Also significant is that reducing the number of displacement vectors from four to one has very little effect on classification performance. This is important since the time required to calculate the ECMs is directly dependent on the number of displacement vectors.

Several different combinations of edge pixel pair displacements were experimented with. Increasing the displacement distance reduced classification success rates, which is to be expected, because the further apart two pixels are, the less likely they are to have any meaningful relationship to each other, and also the less likely it is that they are both edge pixels, since a significant portion of the image contains no edges. It was found that restricting the displacements to only two, vectors  $(1, 1)$  and  $(2, 2)$ , and concatenating the resulting vectors gave nearly as good results as any other reasonable combination of multiple displacements, with the advantage of being relatively fast to compute and resulting in a relatively short feature vector.

With the CCM the smallest displacements are not likely to give optimal results. In the ECM, the greater the distance, the less likely it is for an edge pixel to be paired with another edge pixel, and the less likely it is for the occurrence of such a pair to represent a significant relationship between those pixels. In the CCM all chain links represent points on the contour of an object, and all link

pairs have a definite relationship, depending on the displacement between them and the shape and size of the contour. At different distances, different features of the contours are represented in the CCM, and the displacements that give the best results need to be determined individually for each set of data.

Based on experimental results, the displacements 10 and 20 were chosen for the CCM1 in the metal database, and the displacements 20 and 40 in the paper database. The resulting matrices were concatenated to form the feature descriptor. For the CCM2, the relative displacements 0.10, 0.20, 0.30, and 0.40 were chosen, and the matrices were summed together to form the feature vector. For more details, see [5].

Classification results using the descriptors CCM1 and CCM2 as developed above are presented in Table 1. Although the CCM1 performed better, it also required more care in selecting the displacements. If optimizing the selection of displacements is not possible, e.g. the database grows during use, and the initial set is not representative of the actual pool of data being used, then the CCM2 is probably more reliable, due to the use of relative displacements. In this case, using the CCM1 with optimized displacements gives a slight advantage. In the remaining experiments only the CCM1 will be used, and will be referred to simply as the CCM.

**Table 1.** Comparison between the CCM1 and the CCM2

	Classification success rates (%)			
	CCM1 unnorm.	CCM1 norm.	CCM2 unnorm.	CCM2 norm.
Metal	53	49	51	47
Paper	56	58	55	54

### 3.3 Haralick Features

Using multiple displacement vectors and forming the feature vector by concatenating matrices can result in a long feature vector. Although the performance of the feature can be improved by increasing the information in the feature vector, computations are more efficient with shorter vectors, which may be crucial in real-time applications. One way to make shorter vectors while retaining the information necessary for good performance is to extract from the matrix a set of features which are commonly used with the GLCM. These include

, and , which are the original Haralick features [3], and and from the additional features introduced by Connors and Harlow [12]. While the GLCM is symmetric, the ECM and the CCM are not, which means that one of the original Haralick

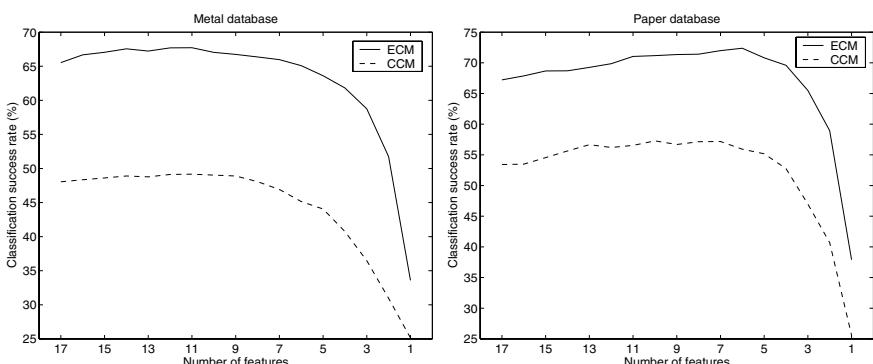
features, ..., needs to be calculated for both axes of the matrix. These features are called simply ... and ... In order to calculate these features, the matrix must be normalized so that it can be interpreted as a probability distribution.

A matrix obviously cannot be described completely with this set of features, but performance comparable to that of the original matrix can be achieved with a suitable selection from the features. The extracted feature set may even have a better classification success rate than the original matrix, but since the objective is to reduce the feature vector length, it is more interesting to see what is the smallest set that still performs at an acceptable level compared with the original matrix.

The set of all 17 features was used as a starting point, and a greedy algorithm was used to eliminate features from the set. On each iteration, the classification was performed with all such subsets of the current feature set where one feature had been removed. The subset that gave the highest overall success rate was used as the set for the next iteration. Since the values of the features have very different ranges, the feature vector was whitened using PCA, resulting in a vector where each component has zero mean and unit variance. Figure 2 shows graphs for the classification rate at each iteration.

The lowest number of features sufficient for achieving the same approximate level of performance as the original matrix was four, with the ECM in the paper database. In the metal database the same number of features performed at the level of the normalized matrix, which is only a little weaker than the unnormalized matrix, so four was selected as the size of the Haralick feature sets. The sets are listed in Table 2.

With the CCM, the performance of the feature set is not as good in comparison to the original matrix as it is with the ECM. This is especially evident in the metal database, where the performance level of the unnormalized matrix was not reached at any point, and the level of the normalized matrix required nine features.



**Fig. 2.** Classification success rates as a function of the number of features during the feature elimination

**Table 2.** The sets of features used in classification

	Metal	Paper
ECM	Sum of squares 1	Sum of squares 1
	Sum of squares 2	Sum of squares 2
	Information measures of correlation 2	Information measures of correlation 1
	Sum average	Sum entropy
CCM	Difference entropy	Inverse difference moment
	Information measures of correlation 1	Entropy
	Cluster shade	Information measures of correlation 2
	Cluster prominence	Cluster prominence

### 3.4 Feature Comparisons

The ECM and CCM were compared with other shape descriptors with similar approaches. The unnormalized version of the ECM, the normalized version, and the Haralick features calculated from the normalized ECM were compared with the Edge Histogram (EH) and the Simple Edge Histogram (SEH). The unnormalized CCM, the normalized version and the extracted Haralick features were compared with the Contour-based Shape (CBS), the Region-based Shape (RBS), the Chain Code Histogram (CCH), and the Simple Shape Descriptor (SSD). The results are shown in Tables 3 and 4.

The results show the superiority of the ECM over the EH. In the metal database, the unnormalized ECM scores 18 percentage units higher than the EH. In the paper database, the difference is 9 percentage units. However, when compared with the SEH, the differences are 23 percentage units for metal and 29 percentage units for paper, so the difference between the first order SEH and second order ECM appears consistent between the databases. The considerable advantage the EH has over the SEH in the paper database may be related to the fact that paper defects have more clearly defined edges than metal defects. The stronger contrast between edge regions and non-edge regions could result in less correlation between the feature vector components and increased discriminatory power for the EH in the paper database.

The unnormalized ECM is better than the normalized ECM in both databases, and in part this may be caused by its greater sensitivity to scale. However, in some classes the normalization decreases the success rate significantly, but extracting Haralick features from the normalized matrix increases the success rate again, even exceeding the success rate of the unnormalized ECM. This suggests that the decrease in success rate associated with normalization is not simply a result of a loss of information.

In the CCM comparisons the best results were obtained with the CCM, and the second best with the SSD. In the metal database the unnormalized CCM scored 4 percentage units higher than the normalized CCM, and 11 percentage units higher than the SSD. However, in the paper database the normalized CCM

**Table 3.** KNN classification results in the metal and paper databases

	Classification success rates (%)				
	ECM unnorm	ECM norm.	ECM Haralick	EH	SEH
Metal	67	63	62	49	44
Paper	69	64	70	60	40

**Table 4.** KNN classification results in the metal and paper databases

	Classification success rates (%)						
	CCM unnorm.	CCM norm.	CCM Haralick	SSD	CCH	CBS	RBS
Metal	53	49	42	42	36	31	20
Paper	56	58	52	52	40	49	46

scored 2 percentage units higher than the unnormalized CCM and only 6 percentage units higher than the SSD. The lowest scorers are the CCH, the CBS and the RBS. Their rankings are inconsistent between the databases, the RBS being the worst in the metal database and the CCH being the worst in the paper database. The advantage of the CCM over the first order CCH is consistent between the databases, 17 percentage units for metal and 18 percentage units for paper.

## 4 Conclusions

Statistical shape features can be used to describe irregular objects such as surface defects. The Edge Co-occurrence Matrix (ECM) and Contour Co-occurrence Matrix (CCM) perform better in surface defect classification tasks than their first order counterparts and other shape descriptors, including the Region-based Shape (RBS) and Contour-based Shape (CBS) descriptors from the MPEG-7 standard. The ECM performs better than the CCM, which may be explained by the fact that the defect images do not have well-defined boundaries.

The length of the feature vectors can be decreased by extracting Haralick features from the co-occurrence matrices. No set that would be conveniently small and would also give good results for each descriptor–database-combination could be found. The effort of computing the features and finding a good set should be considered if Haralick features are used.

**Acknowledgments.** The financial supports of the Technology Development Centre of Finland (TEKES's grant 40102/04) and our industrial partner ABB Oy (J. Rauhamaa) are gratefully acknowledged.

## References

1. Marshall, S.: Review of shape coding techniques. *Image and Vision Computing* **7** (1989) 281–294
2. Rautkorpi, R., Iivarinen, J.: A novel shape feature for image classification and retrieval. In: Proceedings of the International Conference on Image Analysis and Recognition, Porto, Portugal (2004)
3. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3** (1973) 610–621
4. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis and Machine Vision*. Chapman & Hall Computing, London (1993)
5. Rautkorpi, R.: Shape features in the classification and retrieval of surface defect images. Master's thesis, Helsinki University of Technology (2005)
6. Freeman, H.: Computer processing of line-drawing images. *Computing Surveys* **6** (1974) 57–97
7. Iivarinen, J., Visa, A.: Shape recognition of irregular objects. In Casasent, D.P., ed.: *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*. Proc. SPIE 2904 (1996) 25–32
8. Manjunath, B.S., Salembier, P., Sikora, T., eds.: *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons Ltd. (2002)
9. Iivarinen, J., Visa, A.: An adaptive texture and shape based defect classification. In: Proceedings of the 14th International Conference on Pattern Recognition. Volume I., Brisbane, Australia (1998) 117–122
10. Pakkanen, J., Ilvesmki, A., Iivarinen, J.: Defect image classification and retrieval with MPEG-7 descriptors. In Bigun, J., Gustavsson, T., eds.: *Proceedings of the 13th Scandinavian Conference on Image Analysis*. LNCS 2749, Gteborg, Sweden, Springer-Verlag (2003) 349–355
11. Iivarinen, J., Rautkorpi, R., Pakkanen, J., Rauhamaa, J.: Content-based retrieval of surface defect images with PicSOM. *International Journal of Fuzzy Systems* **6** (2004) 160–167
12. Connors, R., Harlow, C.: Toward a structural textural analyser based on statistical methods. *Computer Graphics and Image Processing* **12** (1980) 224–256

# Complex Correlation Statistic for Dense Stereoscopic Matching

Jan Čech and Radim Šára

Center for Machine Perception  
Czech Technical University  
Prague, Czech Republic

{cechj, sara}@cmp.felk.cvut.cz, <http://cmp.felk.cvut.cz>

**Abstract.** A traditional solution of area-based stereo uses some kind of windowed pixel intensity correlation. This approach suffers from discretization artifacts which corrupt the correlation value. We introduce a new correlation statistic, which is completely invariant to image sampling, moreover it naturally provides a position of the correlation maximum between pixels. Hereby we can obtain sub-pixel disparity directly from sampling invariant and highly discriminable measurements without any postprocessing of the discrete disparity map. The key idea behind is to represent the image point neighbourhood in a different way, as a response to a bank of Gabor filters. The images are convolved with the filter bank and the complex correlation statistic (CCS) is evaluated from the responses without iterations.

## 1 Introduction

In stereo, we have to recognize corresponding points, i.e. image points which are the projection of the same spatial point, according to how much the image point neighbourhoods are similar, computing some kind of image correlation statistic.

A stereo algorithm usually consists of two essential modules: the measurement evaluation and the matching. The first process computes some kind of similarity (correlation) statistics between all potential correspondences. The second process takes these measurements and establishes matches according to some principle or optimizing some criterion. Both stages are important and dramatically influence the matching results, which could be seen in stereo evaluation works [10, 4]. This paper is exclusively devoted to the similarity measurement stage introducing a new correlation statistic which can be used by various matching algorithms. We introduce a Complex Correlation Statistic (CCS) which is invariant to image sampling and allows sub-pixel match localization.

Birchfield and Tomasi [1] noticed that it is important the correlation statistic be invariant to image discretization and proposed a sampling-invariant pixel dissimilarity. It is a simple extension of Sum of Absolute Differences (SAD) based on a linear interpolation. It works quite well but it tends to fail where there are very high frequencies in the images. The aggregation of this pixel

dissimilarity over a window has become popular in area-based stereo, e.g. [16]. Szeliski and Scharstein [14] continued with this effort and recommended several other matching scores based on interpolated image signals, e.g. to interpolate the images to a higher resolution, compute the usual statistics, aggregate and subsample to the original resolution. The drawback of this approach is that the increase in resolution is finite, which limits the discretization invariance property.

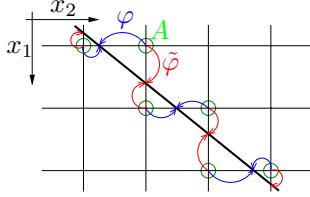
There are several possibilities to achieve sub-pixel matching. Again, the simplest way is to work with interpolated high resolution images. We could have the sub-pixel precision up to the level of interpolation and also the statistic less sensitive to image sampling. But, computational expenses increase dramatically. A possible solution might be to interpolate in the space of correlation statistic (disparity space), like fitting a parabola between three values in the correlation table to find where the extreme of the statistic is. These methods were studied by Shimizu and Okutomi [11, 12], where they formulate which interpolant is suitable for which statistics, and proposed a method to compensate a systematic error using that method. In theory, a sub-pixel disparity map could be also obtained from any probabilistic labeling framework, e.g. [13, 15], where the final disparity is determined as a mean value from all integer disparity labels. However, it has not been analyzed properly whether such an estimate is accurate.

Sub-pixel matching can also be achieved using phase-based techniques. They are based on the Fourier shifting theorem: a shift in the spatial domain corresponds to a phase shift in the frequency domain. Weng [17] introduced the windowed Fourier phase as a matching primitive in a hierarchical disparity estimation procedure. Sanger [6] used Gabor filters and the final disparity is calculated as a weighted average of the disparities obtained from various filter bands. Fleet et al. [3, 2] also used Gabor filters. They introduced a stability criterion of the phase, which is based on the discrepancy between the tuning frequency of the Gabor filter and the local frequency of the response. Unstable estimates are not used in the subsequent iterative algorithm. Xiong and Shaffer [18, 19] proposed to use hypergeometric filters beside the Gabors. They use high-order Taylor expansion of the response which allows affine invariant estimation, but it requires solving a set of equations by an iterative procedure for each match.

Our approach is very close to phase-based techniques in the sense we model the image using Gabor filters and exploit the shifting theorem. But we do not estimate the disparity map directly, we embed the estimation of the sub-pixel disparity in the correlation statistic which is evaluated in a closed form without any complicated optimizations.

## 2 Complex Correlation Statistic

**Definition.** Let us have rectified images  $I_L(x, y)$  and  $I_R(x, y)$ , in which epipolar lines coincide with image rows  $y$  in both images. Image similarities for all potential pixel correspondences  $(x_1, y) \in I_L, (x_2, y) \in I_R$ , for current scanline  $y$  form so called correlation table  $c(x_1, x_2)$ . We define the Complex Correlation

**Fig. 1.** Correlation table of the CCS**Fig. 2.** Computational block

Statistic to be a complex number

$$\text{CCS}(x_1, x_2) = Ae^{j\varphi}, \quad (1)$$

where the magnitude  $A$  is the similarity value which is invariant to image sampling, and the phase  $\varphi$  shows the correct position of the match between pixels, see Fig. 1. The thick black line represents the truth matching, i.e. an image of a surface in the disparity space. Green circles mark cells of the correlation table at locations, where magnitude  $A$  should be the highest (ideally 1). Blue arrows represent the angle  $\varphi$  pointing towards the correct match position in the horizontal direction. Red arrows are pointing towards the correct match position in the vertical direction. This is the angle  $\tilde{\varphi}$  of the complementary correlation  $\tilde{\text{CCS}}(x_1, x_2) = Ae^{j\tilde{\varphi}}$ . Magnitudes of CCS and  $\tilde{\text{CCS}}$  are the same, the only difference is in phases. These statistics are evaluated in each cell of the correlation table. A flowchart of this procedure is sketched in Fig. 2. The inputs are intensity values of the neighbourhood  $\mathcal{N}$  of the left  $I_L$  and right  $I_R$  image at position  $x_1$ ,  $x_2$ , respectively on the  $y$ th row. Swapping the inputs  $f$ ,  $g$  of this block causes swapping the CCS to  $\tilde{\text{CCS}}$ .

In the following sections, we will describe what is inside the CCS-block. First, we give the formula and then we explain it.

**Procedure for computing the CCS.** Both images are convolved with a bank of several Gabor filters tuned to different frequencies to equally sample the frequency spectra, and with the corresponding  $x$ -partial derivative filter bank. The CCS is computed from the responses in a closed form. The bank of filters is

$$c_i(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j(u_{0i}x+v_{0i}y)}, \\ c_{xi}(x, y) = \frac{\partial c_i(x, y)}{\partial x} = \left(-\frac{x}{\sigma^2} + ju_{0i}\right) c_i(x, y), \quad (2)$$

where  $i = 1, \dots, N$  is the index of a filter tuned to a frequency  $(u_{0i}, v_{0i})$  with a constant scale  $\sigma$ , and the convolutions with input images  $f(x, y)$ ,  $g(x, y)$  are

$$G_{fi}(x, y) = f * c_i, \quad G_{gi}(x, y) = g * c_i, \quad G_{xfi}(x, y) = f * c_{xi}. \quad (3)$$

We estimate the local frequency component, i.e. a partial derivative of the phase of the Gabor response

$$u_{fi}(x, y) = \frac{\Im(G_{xfi})\Re(G_{fi}) - \Re(G_{xfi})\Im(G_{fi})}{|G_{fi}|^2}. \quad (4)$$

Now, the CCS will be evaluated per scanline  $y$ . First, for all the cells of the correlation table and for all the filters  $i = 1, \dots, N$ , we estimate the local sub-pixel disparity

$$d_i(x_1, x_2) = \frac{\arg(\overline{G_{f_i}(x_1, y)}G_{g_i}(x_2, y))}{u_{f_i}(x_1, y)}. \quad (5)$$

We denote  $a_{f_i}(x, y) = |G_{f_i}|$ ,  $a_{g_i}(x, y) = |G_{g_i}|$  and finally, we aggregate the data

$$\text{CCS}(x_1, x_2) = \frac{2 \sum_{i=1}^N a_{f_i}(x_1, y)a_{g_i}(x_2, y)e^{jd_i(x_1, x_2)}}{\sum_{i=1}^N a_{f_i}(x_1, y)^2 + \sum_{i=1}^N a_{g_i}(x_2, y)^2}. \quad (6)$$

In this subsection we have described completely the procedure for computing the correlation table of the Complex Correlation Statistic. In the next section we will explain it more in details and show why it works.

**Derivation of the formula for the CCS.** Locally, we have two signals  $f(x, y)$  and  $g(x, y)$  related by

$$g(x, y) = f(x + d(x, y), y), \quad (7)$$

where the (local) disparity is assumed to be small and linearly varying with  $x, y$

$$d(x, y) = d_0 + d_1x + d_2y. \quad (8)$$

First, we will show the  $d(x, y)$  can be estimated from the responses of the Gabor filters. Let us assume that our signals are real signals consisting of a single frequency  $(u, v)$  only

$$\begin{aligned} f(x, y) &= a \cos(ux + vy + \varphi), \\ g(x, y) &= a \cos(u(x + d_0 + d_1x + d_2y) + vy + \varphi). \end{aligned} \quad (9)$$

Convolving these signals with the Gabor filter tuned to the frequency  $(u_0, v_0)$ , we get

$$\begin{aligned} G_f(x, y) &= f * c \approx 1/2 ae^{-\sigma^2((u_0-u)^2+(v_0-v)^2)} e^{j(ux+vy+\varphi)}, \\ G_g(x, y) &= g * c \approx 1/2 ae^{-\sigma^2((u_0-(u+d_1))^2+(v_0-v)^2+2d_2(v-v_0))} \\ &\quad \cdot e^{j(u(d_0+d_1x+d_2y)+ux+vy+\varphi)}. \end{aligned} \quad (10)$$

This holds well where the signal frequency  $(u, v)$  and the tuning frequency of the filter  $(u_0, v_0)$  are close to each other. Then we can neglect the contribution of the symmetric frequencies, which arises from the convolution integrals. The disparity is determined from the argument of the Gabor responses:

$$\arg(\overline{G_f}G_g) = -(ux+vy+\varphi) + (u(d_0+d_1x+d_2y)+ux+vy+\varphi) = u d(x, y), \quad (11)$$

which is the formula (5), because  $u = u_f$  is the local frequency, as can be seen by taking partial derivatives of the arguments of (10):

$$\begin{aligned} u_f &= \frac{\partial \arg(G_f)}{\partial x} = u, & v_f &= \frac{\partial \arg(G_f)}{\partial y} = v, \\ u_g &= \frac{\partial \arg(G_g)}{\partial x} = u + d_1u, & v_g &= \frac{\partial \arg(G_g)}{\partial x} = v + d_2u. \end{aligned} \quad (12)$$

We can estimate the disparity gradient  $(d_1, d_2)$  from the local frequencies as  $d_1 = (u_g - u_f)/u_f$ ,  $d_2 = (v_g - v_f)/u_f$ . If we use  $u_g$  instead of  $u_f$  in the denominators here and in (5), we get the complementary disparity  $\tilde{d}$ , discussed in Sec. 2, the definition.

Numerically, we can estimate the local frequency, i.e. the partial derivative of the phase of the response, from the Gabor and Gabor partial derivatives filters. We denote  $R = \Re(G_f)$ ,  $I = \Im(G_f)$ ,  $R_x = \Re(G_{xf})$ ,  $I_x = \Im(G_{xf})$ . Then

$$u_f = \frac{\partial \arg(G_f)}{\partial x} = \frac{\partial}{\partial x} \arctan\left(\frac{I}{R}\right) = \frac{1}{R^2 + I^2} \left( \frac{\partial I}{\partial x} R - \frac{\partial R}{\partial x} I \right) = \frac{I_x R - R_x I}{R^2 + I^2}, \quad (13)$$

which is the formula (4). Other local frequencies can be derived analogously.

Computation of the CCS according to (6) is inspired by the Moravec's normalized cross-correlation [5], which works well for windowed data. We use a similar formula regardless of the fact we work with complex Gabor responses. It is not exactly the same, because there is the necessary frequency normalization (5). After the normalization, the meaning is as follows: Under the correspondence, the local disparity estimates  $d_i$  from all the filters should agree and the magnitudes  $a_{f_i}$  and  $a_{g_i}$  for each filter should also be the same. The formula (6) measures how much this is true.

For example, let us have two images where one is shifted from the other with a constant disparity  $\delta$  which is smaller than one pixel,  $f(x, y)$  and  $g(x, y) = f(x + \delta, y)$ . No noise is assumed. Then all the local disparities in (5) are  $d_i = \delta$  and from (10), we can see that  $a_{f_i} = a_{g_i} = a_i$ . Substituting this into (6) we get

$$\text{CCS}(f(x), f(x + \delta)) = \frac{2 \sum_{i=1}^N a_i a_i e^{j\delta}}{\sum_{i=1}^N a_i^2 + \sum_{i=1}^N a_i^2} = e^{j\delta}. \quad (14)$$

The phase  $\arg(\text{CCS}) = \delta$  and the magnitude  $|\text{CCS}| = 1$ . Clearly, when either the local disparities  $d_i$  do not agree or the local magnitudes differ, the  $|\text{CCS}| < 1$ .

Notice, when the scene is not fronto-parallel, i.e. the disparity is not constant  $d_1 \neq 0$ ,  $d_2 \neq 0$ , then the magnitudes of the Gabor responses  $a_{f_i}$ ,  $a_{g_i}$  differ in (10). It means the  $|\text{CCS}| < 1$ , but according to experiments we made, it does not cause serious problems for reasonable slants, see Sec 3.

This derivation is valid only for a mono-frequency signal. But it holds well for general signals containing all frequencies. This is due to high frequency localization of the Gabor filters, proportional to the scale  $\sigma$ . The other reason it works well is that formula (6) aggregates the data from various filter bands and averages out weak response estimates and symmetrically invalid estimates. Therefore, we do not need to use any stability criterion as [3, 18].

**Usage of the CCS and technical notes.** For each scanline we get the correlation table of CCS. This is a finite set of the sub-pixel correspondence hypotheses. The task of the matching algorithm is to select a subset. It can be simplified so that, we submit a table of magnitudes  $|\text{CCS}|$  to a common discrete algorithm, which establishes the integer matches. Then the sub-pixel disparity is obtained by adding the phase of the CCS to them.

There are high correlation values  $|CCS|$  in the vicinity of the truth (sub-pixel) matches, which are due to the fact that each CCS phase aims at the same point, as in Fig. 1. The maximum  $|CCS|$  is not sharp as a consequence. It might be a problem for some algorithms. So we rearrange the table of magnitudes. The resolution of the table in the direction of the disparity correction is increased twice by adding  $1/2$  pixel cells. The correlations including the phase are binned in these new cells. Then the magnitude in each bin is determined from the correlation which has the smallest phase. Other magnitudes are set to zero.

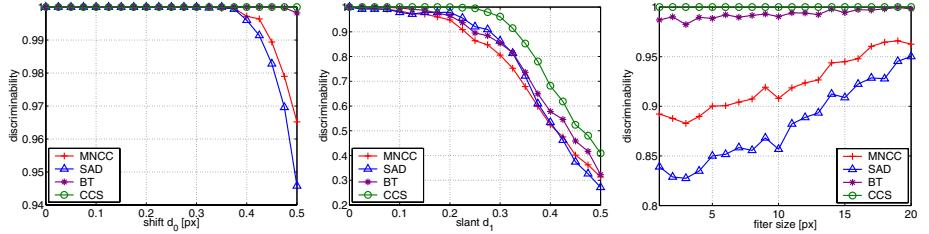
Our filter bank usually contains 50 filters, 10 in the horizontal, 5 in the vertical frequency direction. Gabor filter in the frequency domain is a Gaussian centered at the tuning frequency with a standard deviation proportional to the scale  $1/\sigma$ . So, the tuning frequencies are selected uniformly from  $\pm[0.2\pi, 0.8\pi] \times [0.2\pi, 0.8\pi]$ . Too high and too low frequencies are excluded because of aliasing and because of the approximations in (10). The scale of the filters is selected from the range  $\sigma \in [2, 5]$ , depending on the scene complexity. This is a parameter.

### 3 Experiments

We will demonstrate the important properties of CCS: the magnitude is discriminable and invariant to image sampling and the sub-pixel disparity estimation of its phase is accurate. We made several experiments on both synthetic and real data. We will quantitatively evaluate the performance of CCS and compare to other correlation statistics. Qualitative evaluation as disparity maps of real outdoor scenes will be presented.

**Synthetic Data.** We generated stereo-image pairs assuming the cameras observe a textured plane, i.e. the ground-truth disparity is a linear function as in (8). The images were generated in a high resolution  $1000 \times 1000$ , warped according to the required disparity and finally integrated (filtered) and subsampled into a working resolution  $50 \times 50$ . In the following experiments, the Gabor scale parameter of CCS was set  $\sigma = 2$  and the size of the window of other statistics to  $5 \times 5$  pixels.

The discriminability is a property which intuitively says that correlation statistic assigns high values to the true corresponding pairs while keeping low values of all other potential matches. We define the discriminability as a probability that the ground-truth match has all X-zone competitors [8] of lower correlation, see Fig. 4, and it is estimated from  $discriminability = \text{card}\{(i, j) \in \mathcal{G} : \forall(k, l) \in \mathcal{X}(i, j), c(k, l) < c(i, j)\} / \text{card } \mathcal{G}$ , where  $(i, j)$  is a cell in the correlation table,  $\mathcal{G}$  (red circles) is the set of the ground-truth correspondences and the  $\mathcal{X}(i, j)$  (green crosses) is the forbidden zone for  $(i, j)$ , and  $c(i, j)$  is the correlation value. The estimation was averaged from 100 random trials over texture generation for each stereopair. This definition of discriminability is insensitive to the scale of the statistic  $c$ . We only need the similarity property: higher similarity implies higher value of  $c$ .



**Fig. 3.** Discriminability experiment results

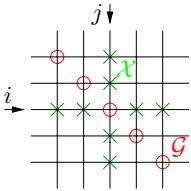
We will compare the discriminability of CCS with other (window) statistics: the Sum of Absolute Differences (SAD), the Moravec's Normalised Cross Correlation (MNCC) [5] and the sum of Birchfield-Tomasi sampling insensitive pixel dissimilarities [1] over a window. The SAD and BT is redefined to be a difference of the original statistic from 1 to have the similarity property.

We measured the discriminability for a constant sub-pixel disparity:  $d(x, y) = d_0, d_0 \in [0, 0.5]$  px (fronto-parallel scene), and for a slanted plane:  $d(x, y) = d_1 x, d_1 \in [0, 0.5]$ . We do not present results for the other slant  $d_2$ , since it was very similar to the behaviour under varying  $d_1$ .

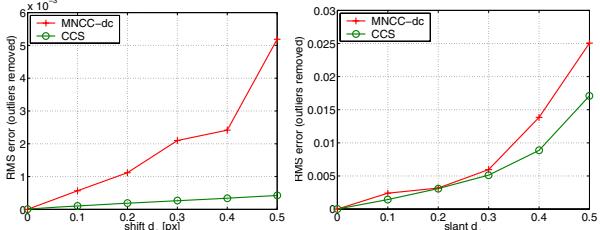
The results for the fronto-parallel case are shown in Fig. 3 (left): The worst case occurs where the true disparity  $d_0$  is 0.5 pixel. The SAD and MNCC tends to fail towards this point, BT has small problems at this point too, although it should be invariant to image discretization, but due to interpolation used in BT, it gets worse as more high frequencies are present, see Fig. 3 (right). We did not observe any failure of CCS in any of the 100 trials, which corroborates its invariance to image sampling. Unlike the others, the CCS has no problems with high frequencies, see Fig. 3 (right). These plots show the discriminability where  $d(x, y) = 0.5$  px versus the size of the moving average integration filter used for image subsampling. The size=1 means, there are all frequencies up to the Nyquist limit and they are attenuated with increasing filter size. The results for the slanted plane are shown in Fig. 3 (center): None of the tested statistics is invariant to the slant. The discriminability decreases with higher slants  $d_1$  as expected, but, surprisingly, the CCS is the best for all measurements.

By matching accuracy we mean that the estimated disparity is close to the ground-truth disparity. We measure the root mean square (RMS) error of this difference, but matches whose error in disparity is higher than 1 pixel are considered outliers and excluded. These outliers are mismatches, a consequence of the low discriminability. The ratio of outliers was 0.04 for the worst case.

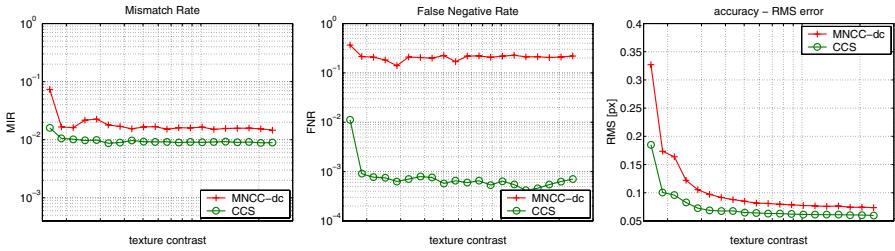
We ran the same integer matching, Confidently Stable Matching (CSM) [8], first with CCS magnitudes, and second with MNCC. We compared the accuracy of the sub-pixel disparity obtained directly from the CCS phase (CCS) with the disparity correction (DC) method on the MNCC matching (MNCC-dc). This method postprocess the integer disparity map by fitting affine window optimizing a ML criterion [7]. Note that RMS results are primarily determined



**Fig. 4.** To the definition of discriminability



**Fig. 5.** Accuracy experiment results



**Fig. 6.** Laboratory test scene: matching error evaluation plots

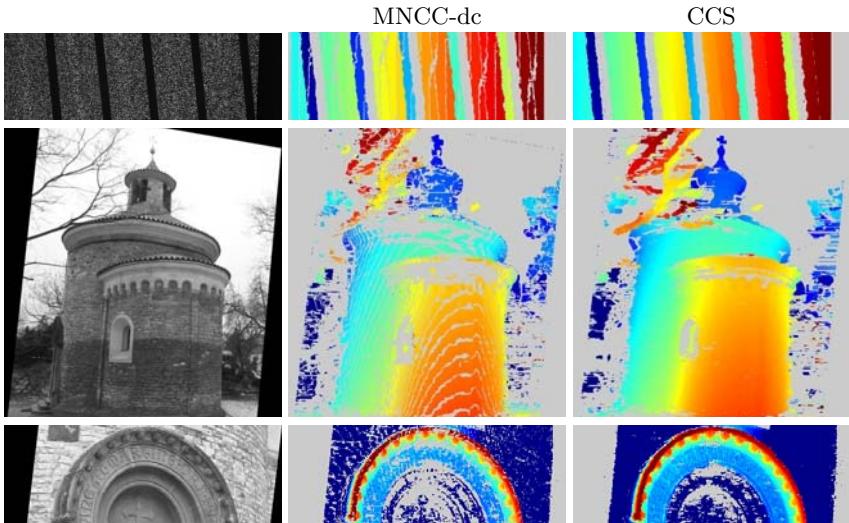
by the DC step, hence we do not need to test other statistics for the pixel resolution matching (like BT, SAD followed by DC, etc.)

We measured the accuracy for the fronto-parallel plane, see Fig. 5 (left), and for the slanted plane Fig. 5 (right) as in the discriminability experiments. The CCS is clearly the best for the fronto-parallel case. Surprisingly, the CCS remains better for slanted scenes, despite it does not explicitly model this distortion unlike the DC method.

**Real data.** Experiments on images captured by real cameras follow.

The algorithm performance was tested on the CMP dataset [4]. The scene consists of thin stripes in front of a planar background, see Fig. 7 (first row). The algorithms are evaluated on an image series of varying texture contrast. We observed the Mismatch Ratio MIR (matches which differ more than one pixel from the ground-truth), the False Negative Ratio FNR (the sparsity of matching) and the accuracy as the RMS error, Fig. 6. We can see the performance of CCS is better for all texture contrasts in all observed quality measurements. It has about twice less mismatches, the FNR is lower by the order of 2 magnitudes. The accuracy is also better: from twice (for weak textures of low contrast, at left) to 1.5 times lower RMS error for high contrasts. The disparity maps for the best texture contrast are shown in fig. 7 (first row).

We will show two results from an outdoor scene, which were captured by a standard hand-held digital camera. The second row of Fig. 7 is a shot from the St. Martin scene with non-planar surfaces, with some half occluded regions and thin objects. The CCS gives denser results than MNCC-dc. The



**Fig. 7.** Real scenes: left images and disparity maps (color-coded, gray unassigned)

occlusion boundaries are not estimated worse than with a standard correlation. Gabor scale in CCS was set to  $\sigma = 4$  and the window size of MNCC was set to  $9 \times 9$  pixels. Next example, Fig. 7 (last row), is a scene with an inscription above the door. Importantly, the letter-to-background difference in disparity is less than one pixel. The inscription is clearly visible in the CCS disparity map, while not in the MNCC-dc map, which is very noised. Gabor scale in CCS was set to  $\sigma = 3$  and the window size of MNCC to  $7 \times 7$  pixels.

## 4 Discussion and Conclusions

We have proposed a new Complex Correlation Statistic which is insensitive to image sampling and provides sub-pixel matching accuracy. We showed on both synthetic and real data it achieves a high discriminability and accuracy of the sub-pixel disparity estimation. In theory, using the CCS, we obtain sampled continuous solution from a discrete algorithm.

The model of the CCS assumes locally fronto-parallel scene (constant disparity) and continuous surface. We showed analytically which distortion occurs when the surface is slanted (locally linear disparity). Practical experiments demonstrated that for this case the distortion is not that strong and we have consistently higher discriminability than standard windowed statistics including windowed Birchfield-Tomasi's dissimilarity. The accuracy is higher even compared to the algorithm which models the distortion of the matching window. On a real data, we showed that it works for curved surfaces as well.

When the surface is not continuous, i.e. when there are occlusion boundaries and thin objects, the algorithm exhibits errors as all other window-based al-

gorithms do: These are the overreach artifacts near occlusion boundaries when the texture of the background is weak [9]. This might be quite a strong effect, although the effective window size due to Gaussian envelope is small, non zero entries in the window may reach far. The scale of the Gabor filter  $\sigma$  should be as small as possible, but large enough to ensure the discriminability and numerical stability of the CCS estimation. Perhaps, the solution might be to design a filter bank which consists of filters with various scales  $\sigma$ . Researchers often choose the  $\sigma$  inversely proportional to the tuning frequency, see e.g. [3, 18], but we have not find a rigorous arguments to do so. Designing an optimal filter bank is a subject of our current research.

The computational complexity of this algorithm is higher than the complexity of standard windowed intensity statistics, because we cannot use the boxing algorithm, which exploits precomputed partial sums. But there are no iterations and optimizations within, we have a closed-form formula for the CCS. So, using a special (but simple) parallel hardware, we can have a real time implementation.

**Acknowledgements.** This work was supported by The Czech Academy of Sciences under project 1ET101210406.

## References

1. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *PAMI*, 20(4):401–6, 1998.
2. D. J. Fleet and A. D. Jepson. Stability of phase information. *PAMI*, 15(12):1253–68, 1993.
3. D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.
4. J. Kostková, J. Čech, and R. Šára. Dense stereomatching algorithm performance for view prediction and structure reconstruction. In *Proc. SCIA*, pp. 101–7, 2003.
5. H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th Int. Joint Conf. Artificial Intell.*, pp. 584–94, 1977.
6. T. D. Sanger. Stereo disparity computation using gabor filters. *Biol. Cybern.*, 58(6):405–18, 1988.
7. R. Šára. Sub-pixel disparity correction. Internal working paper 98/01, Center for Machine Perception, Czech Technical University, Prague, 1998.
8. R. Šára. Finding the largest unambiguous component of stereo matching. In *Proc. ECCV*, pp. 900–14, 2002.
9. R. Šára and R. Bajcsy. On occluding contour artifacts in stereo vision. In *Proc. CVPR*, pp. 852–57, 1997.
10. D. Scharstein, R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.
11. M. Shimizu and M. Okutomi. Precise sub-pixel estimation on area-based matching. In *Proc. ICCV*, pp. 90–7, 2001.
12. M. Shimizu and M. Okutomi. Significance and attributes of subpixel estimation on area-based matching. *System and Computers in Japan*, 34(12):1791–1800, 2003.
13. J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *Proc. ECCV*, pp. 510–24, 2002.

14. R. Szeliski and D. Scharstein. Sampling the disparity space image. *PAMI*, 25(3):419–25, 2004.
15. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proc. ICCV*, pp. 900–6, 2003.
16. O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proc. CVPR*, pp. 556–61, 2003.
17. J. Weng. A theory of image matching. In *Proc. ICCV*, pp. 200–9, 1990.
18. Y. Xiong and S. A. Shafer. Variable window gabor filters and their use in focus and correspondence. Technical Report CMU-RI-TR-94-06, 1994.
19. Y. Xiong and S. A. Shafer. Hypergeometric filters for optical flow and affine matching. *IJCV*, 24(2):163–77, 1997.

# Reconstruction from Planar Motion Image Sequences with Applications for Autonomous Vehicles

H. Stewénius, M. Oskarsson, and K. Åström

Centre For Mathematical Sciences,

Lund University, Lund, Sweden

{stewe, magnuso, kalle}@maths.lth.se

**Abstract.** Vision is useful for the autonomous navigation of vehicles. In this paper the case of a vehicle equipped with multiple cameras with *non-overlapping* views is considered. The geometry and algebra of such a moving platform of cameras are considered. In particular we formulate and solve structure and motion problems for a few novel cases. There are two interesting minimal cases; three points in two platform positions and two points in three platform positions. We also investigate initial solutions for the case when image lines are used as features. In the paper is also discussed how classical algorithms such as intersection, resection and bundle adjustment can be extended to this new situation. The theory has been tested on synthetic and real data with promising results.

## 1 Introduction

One example of successful use of computer vision for autonomous vehicle navigation is that of laser guided navigation. Here the sensor can be seen as that of a one-dimensional retina, which has all-around sight in a plane, (typically) a horizontal plane. During the last two decades many geometrical problems and system design problems have been investigated and solved for this system. Algorithms using a pre-calculated map and good initial position was demonstrated in [8]. In 1991 a semi-automatic method for map making was constructed [1] and in 1996 a fully automatic system for map making was made, [2]. Theory for structure and motion recovery is detailed in [3].

During the last decade there has also been many attempts at making fully automatic structure and motion systems for ordinary camera systems. A good overview of the techniques available for structure and motion recovery can be found in [7]. Much is known about minimal cases, feature detection, tracking and structure and motion recovery. Many automatic systems rely on small image motions in order to solve the correspondence problem, i.e. association of features across views. In combination with most cameras' small fields of view, this limits the way the camera can be moved in order to make good 3D reconstruction. The problem is significantly more stable with a large field of view [9]. This has spurred research in so called omnidirectional cameras.

In this paper we investigate an alternative approach to vision-based structure and motion system. We consider a vehicle equipped with cameras. This gives a large combined field of view with simple and cheap cameras. Another advantage is that there are no moving parts contrary to laser scanner sensors. If the cameras can be placed so that the focal points coincide, then the geometrical problems are identical to that of a single camera. Also if the cameras are positioned so that they have a large field of view in common, i.e. a stereo setup, there are known techniques for calculating structure and motion. In this paper we consider cameras where some of these constraints have to be satisfied.

Consider cameras that are fixed to a vehicle and consider images taken by these cameras at different times, where the vehicle has moved. Assume that a number of corresponding points (possibly in different images) are measured. Then there are a number of problems that are interesting to look at.

1. **Calibration.** Assume that each camera sees enough points to calculate its own relative camera motion. Since the relative motion of all cameras is the same, how many cameras and views are needed to solve for the common motion. When is it possible to solve for the cameras' relative positions?
2. **Structure and motion.** Assume that the cameras' positions relative to the vehicle are known, but not the vehicle's motion, represented by a transformation matrix  $\mathbf{T}_i$ . Given a number of corresponding points, how should one calculate the world points  $\mathbf{U}_j$  and the transformations  $\mathbf{T}_i$  from image data?
3. **Intersection.** Assume that the cameras' positions relative to the vehicle are known, and that also the vehicle's motion is known. Assume that a number of corresponding points are measured, how should one calculate the world points  $\mathbf{U}_j$  from image data?
4. **Resection.** Assume that the cameras' positions relative to the vehicle are known, but not the vehicle's motion, represented by a transformation matrix  $\mathbf{T}_i$ . Assume that a number of corresponding points (possibly in different images) are measured, how should one calculate the transformations  $\mathbf{T}_i$  from image data and known world points  $\mathbf{U}_j$ ?

In this paper we assume that the calibration of the cameras relative to the vehicle has been done. In the experiments in section 6 the calibration was done manually. One could also consider autocalibration approaches similar to the approach in [5], where the problem of aligning video sequences was addressed, or the calibration problems in robotics [13] which are similar in nature to the calibration of cameras relative to a vehicle.

Multiple camera platforms have been studied in [10], where a discrete constraint linking two camera motions and image points, similar to the fundamental constraint for two views of a configuration of points, [7] was formulated. However, there is no analysis of that constraint. Similarly in [4] multi-camera motions are considered but the analysis is concentrated to that of motion flow. In this paper we study and solve some of the minimal cases for multi-camera platforms. Such solutions are of paramount importance as initial estimates to bootstrap automatic structure and motion recovery systems.

## 2 Geometry of Multi-camera Platforms

The standard pinhole camera model is used,

$$\lambda \mathbf{u} = \mathbf{P} \mathbf{U}, \quad (1)$$

where the camera matrix  $\mathbf{P}$  is a  $3 \times 4$  matrix. A scene point  $\mathbf{U}$  is in  $\mathcal{P}^3$  and a measured image point  $\mathbf{u}$  is in  $\mathcal{P}^2$ . As the platform moves the cameras move together. This is modeled as a transformation  $\mathbf{T}_i$  between the first position and position  $i$ . In the original coordinate system the camera matrix for camera  $k$  at position  $i$  is  $\mathbf{P}_k \mathbf{T}_i$ .

It is assumed here that the camera views do not necessarily have common points. In other words, a point is typically seen by only one camera. On the other hand it can be assumed that in a couple of neighboring frames a point can be seen in the same camera. Assume here that point  $j$  is visible in camera  $k$ . The measurement equation for the  $n$  points at  $m$  positions is then

$$\lambda_{ij} \mathbf{u}_{ij} = \mathbf{P}_k \mathbf{T}_i \mathbf{U}_j, \quad j = 1, \dots, n, i = 1, \dots, m. \quad (2)$$

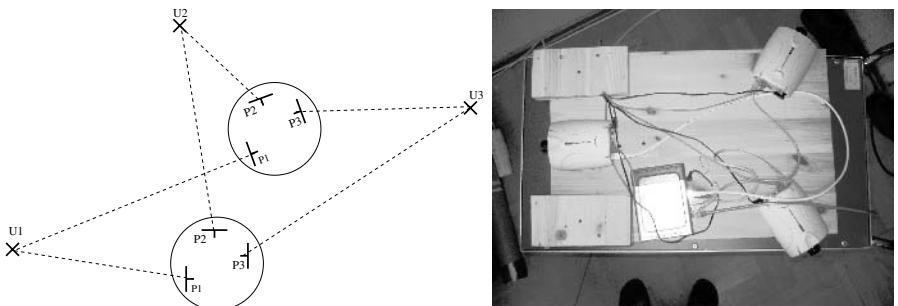
Given  $n$  image points from  $m$  different platform positions  $\mathbf{u}_{ij}$ , and the camera matrices  $\mathbf{P}_k$  the **structure and motion problem** is to find reconstructed points,  $\mathbf{U}_j$ , and platform transformations,  $\mathbf{T}_i$ :

$$\mathbf{U}_j = \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix} \text{ and } \mathbf{T}_i = \begin{bmatrix} a_i & b_i & 0 & c_i \\ -b_i & a_i & 0 & d_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ with } a_i^2 + b_i^2 = 1, \quad (3)$$

such that  $\lambda_{ij} \mathbf{u}_{ij} = \mathbf{P}_k \mathbf{T}_i \mathbf{U}_j, \quad \forall i = 1, \dots, m, j = 1, \dots, n$  for some  $\lambda_{ij}$ .

The special form of  $\mathbf{T}_i$  is due to the planar motion.

Counting the number of unknowns and constraints for generalized cameras restricted to planar motion we have the following, given  $n$  3D-points and  $m$



**Fig. 1.** Three calibrated cameras with constant and known relative positions taking images of three points at two platform positions

cameras. Each point has three degrees of freedom and each camera has three degrees of freedom. Each point in each camera gives two constraints. In addition we have the freedom to specify a Euclidean coordinate system. We have fixed a plane, which leaves three degrees of freedom. So in order to solve a system,

$$2mn \geq 3m + 3n - 3. \quad (4)$$

There are two interesting cases, where equality holds,

**Theorem 1.**

**Theorem 2.**

The two cases will be discussed in sections 2.1 and 2.2 respectively.

## 2.1 Solving for Two Sets of Three Rays

All observations are assumed to be given as Plücker vectors  $(q, q')$ , see e.g. [11]. Assuming that the first camera is at the origin, that is,  $(a_0, b_0, c_0, d_0) = (1, 0, 0, 0)$  there are now only four motion variables left  $(a_1, b_1, c_1, d_1)$ . For convenience we will not use the indices on these variables.

Given a motion  $\mathbf{T}$  and Plücker coordinates for the observations these can be inserted into the generalized epipolar constraint [10]

$$q_1^T R {q'_2}^T + q_1^T R[t] \times q_2 + {q'_1}^T R q_2 = 0 \quad (5)$$

where  $[t]_\times$  is the cross-product matrix formed from  $t$  such that  $[t]_\times v = t \times v$ .

Inserting our three observed point pairs in equation (5) gives 3 equations each of degree 2. We also have that  $a^2 + b^2 = 1$ . This gives a total of 4 equations in 4 unknowns.

The three first equations are in the monomials  $(ac, ad, bc, bd, a, b, c, d, 1)$ . Using linear elimination on these equations gives a way to express  $d$  as a linear combination of  $(a, b, c, 1)$  and  $d$  can thus be eliminated without increasing the degree. This leads to two equations in  $a, b$  and  $c$  named  $f_1$  and  $f_2$ . A third polynomial comes from the rotation constraint,  $f_3 = a^2 + b^2 - 1$ .

The nine polynomials  $f_1, f_2, f_3, af_1, af_2, bf_1, bf_2, bf_3, cf_3$  can be represented as

$$M [a^2b, a^2c, ab^2, abc, a^2, ab, ac, b^2, bc, b^3, b^2c, a, b, c, 1]^T \quad (6)$$

where  $M$  is a  $9 \times 15$  matrix. Performing a Gauss-Jordan elimination on  $M$  gives a GrevLex Gröbner basis. For details on Gröbner bases and their use in elimination theory see e.g. [6].

Given the Gröbner basis the action matrix  $m_a$  for multiplication with  $a$  can be extracted. The matrix  $m_a$  is  $4 \times 4$  and the left eigenvectors give the solutions to  $(a, b, c, 1)$ . One of the solutions corresponds to not moving the vehicle, which leaves three solution of which two may be complex. The remaining motion variable  $d$  is computed using back-substitution.

## 2.2 Solving for Three Sets of Two Rays

For each time  $i = 1, 2, 3$  and point  $j = 1, 2$  we have

$$\pi_{ij} \mathbf{T}_i \mathbf{U}_j = 0 \text{ and } \pi'_{ij} \mathbf{T}_i \mathbf{U}_j = 0, \quad (7)$$

where  $\pi_{ij}$  and  $\pi'_{ij}$  are planes in the camera coordinate system defining the observed ray, calculated from the image point,  $\mathbf{T}_i$  the position of the vehicle and  $\mathbf{U}_j$  the scene point. The coordinate system is fixed by

$$\mathbf{U}_1 = [0 \ 0 \ z_1 \ 1]^T, \mathbf{U}_2 = [0 \ y_2 \ z_2 \ 1]^T. \quad (8)$$

We also know that

$$a_i^2 + b_i^2 = 1, i = 1, 2, 3. \quad (9)$$

Equation (7) can now be written

$$[A + B(\{(a_i, b_i)\})] X = 0 \text{ where } X = [z_1, z_2, c_1, d_1, c_2, d_2, c_3, d_3, 1, y_2]^T \quad (10)$$

and  $A$  is a matrix with coefficients computed from the observations. The vector  $B(\{(a_i, b_i)\})$  is linear in  $\{(a_i, b_i)\}$  and has coefficients computed from the observations. For this  $12 \times 10$  matrix to have non-trivial solutions all  $10 \times 10$  sub-determinants must be zero. All these sub-determinants are linear and homogeneous in  $\{(a_i, b_i)\}$ . The system can now be written as

$$\begin{cases} M [a_1 \ b_1 \ a_2 \ b_2 \ a_3 \ b_3]^T, \\ a_i^2 + b_i^2 = 1, i = 1, 2, 3. \end{cases} \quad (11)$$

The matrix  $M$  is linear in  $(a_1, a_2, a_3, b_1, b_2, b_3)$  and has rank 3. We can thus compute  $(a_1, a_2, a_3)$  as a linear function of  $(b_1, b_2, b_3)$ . This is inserted into equation (9), giving three equations of order two in  $(b_1, b_2, b_3)$ ,

$$Q [b_1^2, b_2^2, b_3^2, b_1 b_2, b_1 b_3, b_2 b_3, 1]^T = 0 \quad (12)$$

where  $Q$  is a  $3 \times 7$  matrix. In order to compute the solutions we compute the multiples of these three polynomials by  $1, x^2, xy, xz, y^2, yz, z^2$  and arrange the coefficients into a matrix. By performing Gauss-Jordan elimination on this matrix we get the elements of the Gröbner basis needed for computing the action matrix for multiplication by  $b_3^2$  on polynomials containing only monomials of even order. Solving the eigenvalue problem gives the three solutions to  $(b_3^2, b_2 b_3, b_1 b_3)$ . From this the values of  $b_3$  are computed and by division,  $b_1$  and  $b_2$  as well.

When the  $\{b_i\}$  have been computed, the  $\{a_i\}$  can be computed using back-substitution in equation (11) and the remaining unknowns can then be computed using back-substitution in equation (10).

## 2.3 Three Positions and Lines

When viewing a line  $\mathbf{l}$  in an image the corresponding scene line is constrained to lie on a scene plane  $\pi = \mathbf{P}^T \mathbf{l}$ . Two positions give no constraints on the platform

motion, but with three positions there is a constraint that three planes intersect in a line. This constraint can be written

$$\text{rank} [\mathbf{T}_1^T \boldsymbol{\pi}_1 \mathbf{T}_2^T \boldsymbol{\pi}_2 \mathbf{T}_3^T \boldsymbol{\pi}_3] = 2. \quad (13)$$

This implies that all four sub-determinants of size  $3 \times 3$  are zero. The sub-determinant that involves rows 1, 2 and 3 can be interpreted as the constraint that three lines (the intersection with the planes and the plane at infinity) intersect in a point (the direction of the space line). This constraint does not involve translation components of  $\mathbf{T}$ , since the plane at infinity is unaffected by translation. Introduce the following parametrisation for the three transformation matrices,

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix} a_2 & e_2 - b_2 & 0 & c_2 \\ b_2 - e_2 & a_2 & 0 & d_2 \\ 0 & 0 & e_2 & 0 \\ 0 & 0 & 0 & e_2 \end{bmatrix}, \mathbf{T}_3 = \begin{bmatrix} a_3 & e_3 - b_3 & 0 & c_3 \\ b_3 - e_3 & a_3 & 0 & d_3 \\ 0 & 0 & e_3 & 0 \\ 0 & 0 & 0 & e_3 \end{bmatrix}.$$

Here we have used homogenized versions of the transformations matrices  $\mathbf{T}_2$  and  $\mathbf{T}_3$ . There is a constraint that the first  $2 \times 2$  block is a rotation matrix, i.e.

$$a^2 + (b - e)^2 - e^2 = a^2 + b^2 - 2be = 0. \quad (14)$$

By dehomogenizing using  $b = 1$ , we see that  $e = (1 + a^2)/2$ . Using

$$b_2 = 1, b_3 = 1, e_2 = (1 + a_2^2)/2, e_3 = (1 + a_3^2)/2, \quad (15)$$

the plane at infinity constraint is of the form

$$(k_{22}a_2^2 + k_{12}a_2 + k_{02})a_3^2 + (k_{21}a_2^2 + k_{11}a_2 + k_{01})a_3 + (k_{20}a_2^2 + k_{10}a_2 + k_{00}) = 0. \quad (16)$$

Thus there are two unknowns on the rotation parameters  $a_2$  and  $a_3$ . Using two different lines, two constraints are obtained. The resultant of these two polynomials with respect to  $a_3$  is an eight degree polynomial. Two of the roots to this polynomial are  $\pm i$ . The remaining six roots can be complex or real. For each solution on  $a_2$ , the variable  $a_3$  can be obtained from equation (16). Then  $(b_2, b_3, e_2, e_3)$  follow from (15).

Once the rotation is known it can be corrected for. The rank constraint (13) is then linear in the translation parameters  $(c_2, c_3, d_2, d_3)$ . Using four lines it is then possible to obtain these motion parameters uniquely in general.

Observe that by counting equations and unknowns, three lines would be a minimal case. However, since the constraint at infinity only involves the rotation parameters, these are found already with two lines. The third line gives one additional constraint on the rotation parameters, which then becomes over-determined, and a third constraint on the four translation parameters. The fourth line gives yet one additional constraint on the rotation parameters and the fourth constraint on the four translation parameters.

There are a couple of interesting degenerate cases. If the camera centers lie in the same horizontal plane, then lines in that plane give no constraints. Vertical

lines give no constraints on rotation. A platform viewing a set of vertical lines is equivalent to a platform motion with 1D retina cameras. For this situation the minimal case is six lines in three views. There are in general 39 solutions to this situation.

### 3 Intersection

Generalization of the intersection algorithm [7] to this new situation is straightforward. When both calibration  $\mathbf{P}_1, \dots, \mathbf{P}_K$  and platform motion  $\mathbf{T}_1, \dots, \mathbf{T}_m$  is known it is straightforward to calculate scene points coordinates  $\mathbf{U}$  from image measurements by intersecting view-lines. A linear initial solutions is obtained by solving

$$\begin{bmatrix} \mathbf{u}_1 \times \mathbf{P}_k \mathbf{T}_1 \\ \vdots \\ \mathbf{u}_n \times \mathbf{P}_k \mathbf{T}_n \end{bmatrix} \mathbf{U} = 0 \quad (17)$$

in a least squares sense. This solution can then be refined by non-linear optimization, cf. section 5.

### 4 Resection

Generalization of the resection algorithm [7] is slightly more complex since the view lines do not go through a common points (the focal point) as in the ordinary camera resection algorithm.

Here we introduce a direct method for finding an initial estimate to the resection problem based on a linearized reprojection error. The idea is to solve

$$\mathbf{u}_j \times \mathbf{P}_k \mathbf{T} \mathbf{U}_j = 0, \quad j = 1, \dots, n, \quad (18)$$

which is linear in  $\mathbf{T}$ . In our case there are non-linear constraints on  $\mathbf{T}$ , so we use the theory for constrained optimization to find the optimal solution under the constraints.

Parameterize the platform motion as in (3), using parameters  $x = (a, b, c, d)$ . The constraints  $\mathbf{u}_j \times \mathbf{P}_k \mathbf{T} \mathbf{U}_j = 0$  is linear in these parameters. The linear constraints can be rewritten  $f = Mx = 0$ . However there is a non-linear constraint  $g = a^2 + b^2 - 1 = 0$ . Initial solution to the resection problem can be found by solving

$$\min_{xg=0} \sum_j |\mathbf{u}_j \times \mathbf{P}_k \mathbf{T}(x) \mathbf{U}_j|^2. \quad (19)$$

Introduce the Lagrangian  $L(x, \lambda) = |Mx|^2 + \lambda g$ . The solution to (19) can be found by  $\nabla L = 0$ . Here

$$\nabla_x L = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} + \lambda \begin{bmatrix} 2a \\ 2b \\ 0 \\ 0 \end{bmatrix} = 0. \quad (20)$$

Here one may solve for  $(c, d)$  as

$$\begin{bmatrix} c \\ d \end{bmatrix} = -D^{-1}C \begin{bmatrix} a \\ b \end{bmatrix}. \quad (21)$$

Inserting this in the above equation gives

$$(A - BD^{-1}C + 2\lambda I) \begin{bmatrix} a \\ b \end{bmatrix} = 0. \quad (22)$$

Here there is a non-trivial solution to  $(a, b)$  if and only if  $-2\lambda$  is one of the two eigenvalues of  $A - BD^{-1}C$ . For these two solutions  $(a, b)$  is determined up to scale. The scale can be fixed by  $a^2 + b^2 = 1$ . Finally  $(c, d)$  can be found from (21). Of the two solutions, only one is a true solution to (19). This gives a reasonably good initial estimate on the resection parameters. This estimate is then improved by minimizing reprojection errors. Experience shows that only a few iterations are needed here.

## 5 Bundle Adjustment

Note that the discussion in the previous sections has focused on finding initial estimates of structure and motion. In practice it is necessary to refine these estimates using non-linear optimization or bundle adjustment, cf. [12, 7]. The generalization of bundle adjustment to platform motions is straightforward. One wants to optimize platform motions  $\mathbf{T}_i$  and scene points  $\mathbf{U}_j$  so that reprojection error is minimized. The fact that the platform has a very large field of view makes bundle adjustment much better conditioned than what is common.

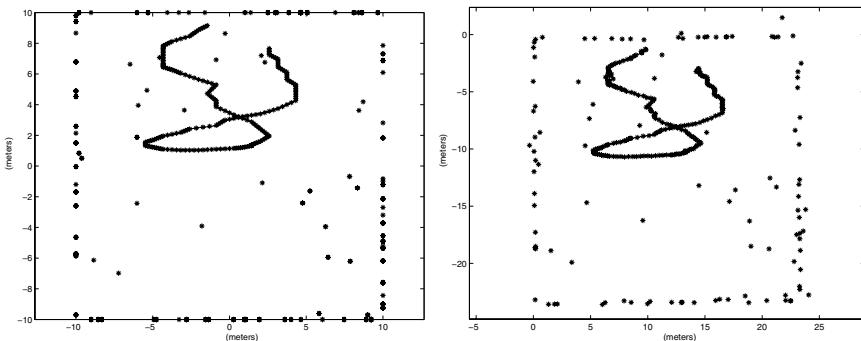
## 6 Experimental Verification

Experiments on both simulated and real data were conducted.

A virtual room was constructed as a cube and points randomly placed on the surfaces of the cube. The synthetic vehicle was moved along the  $z$ -plane taking “pictures” of this room.

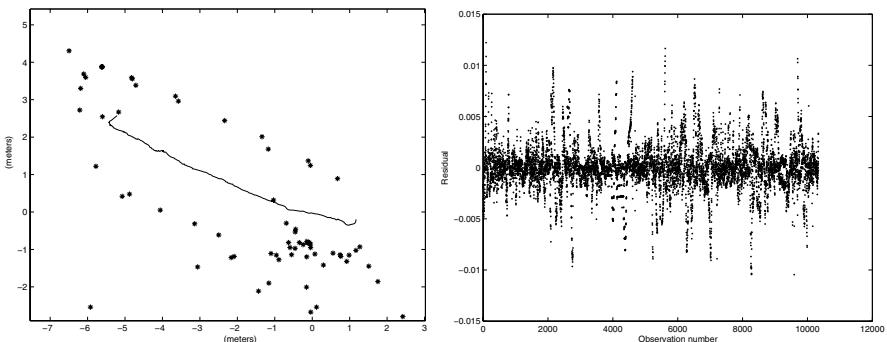
The experiment was carried out at different levels of noise added to the image point measurements. The first experiment was to add errors of magnitude up to 1/100 for a calibrated camera, that is with a known field of view of 0.6 and assuming 300 pixels image width corresponding to an error of 5 pixels. The result is shown in figure 2, with the true room to the left, and the reconstruction to the right.

In an experiment with real data a system with three digital video cameras was assembled and moved along a corridor with markers on the walls. Tracking was done by following these markers and some additional manual tracking.



**Fig. 2.** True room to the left path compared to reconstruction on the right

A reconstruction is shown in figure 3. The structure point which we seem to be passing through is a structure point which has a very high variance, as its measured view lines are almost collinear with the locations from which it is observed. The residuals in reprojection for this reconstruction are shown in figure 3. The size of the residuals are in the order of the error in a calibrated camera.



**Fig. 3.** Reconstruction and residuals for real data

## 7 Conclusions

In this paper we study the visual geometry of a moving platform with multiple cameras (typically pointed outwards) in general positions and orientations. For planar motion of such platforms the case of two motions and at least three points is solved, the case of three motions and two points is solved, as is the case of three motions and a number of lines.

In the experimental validation it is demonstrated how these algorithms, combined with novel algorithms for resection, intersection and bundle adjustment, are used in automatic structure and motion recovery using such platforms. The validation is done both for synthetic and real data.

Future work includes generalizations of the above ideas to that of full (non-planar) camera motion as well as testing and developing fully automatic systems for map-making.

## References

1. K. Åström. Automatic mapmaking. In *1st IFAC, International Workshop on Intelligent Autonomous Vehicles*, 1993. Selected papers.
2. K. Åström. *Invariancy Methods for Points, Curves and Surfaces in Computational Vision*. PhD thesis, Dept of Mathematics, Lund University, Sweden, 1996.
3. K. Åström and M. Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. *Journal of Mathematical Imaging and Vision*, 12(2):121–135, 2000.
4. P. Baker, C. Fernmuller, and Y. Aloimonos. A spherical eye from multiple cameras (makes better models of the world). In *Proc. Conf. Computer Vision and Pattern Recognition, Hawaii, USA*, 2001.
5. Y. Caspi and M. Irani. Alignment of Non-Overlapping sequences. In *Proc. 8th Int. Conf. on Computer Vision, Vancouver, Canada*, pages 76–83, 2001.
6. D. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.
7. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
8. K. Hyppä. Optical navigation system using passive identical beacons. In Louis O. Hertzberger and Frans C. A. Groen, editors, *Intelligent Autonomous Systems, An International Conference, Amsterdam, The Netherlands, 8-11 December 1986*, pages 737–741. North-Holland, 1987.
9. M. Oskarsson and K. Åström. Accurate and automatic surveying of beacon positions for a laser guided vehicle. In *Proc. European Consortium for Mathematics in Industry, Gothenburg, Sweden*, 1998.
10. R. Pless. Using many cameras as one. In *Proc. Conf. Computer Vision and Pattern Recognition, Madison, USA*, 2003.
11. J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Clarendon Press, Oxford, 1952.
12. C. C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, 1980.
13. H. Zhuang, Z. Roth, and R. Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformations of the form  $ax=by$ . *IEEE Trans. on Robotics and Automation*, 10(4):549–554, 1994.

# Stereo Tracking Error Analysis by Comparison with an Electromagnetic Tracking Device

Matjaž Divjak and Damjan Zazula

System Software Laboratory,  
Faculty of Electrical Engineering and Computer Science,  
University of Maribor,  
Smetanova 17, Maribor, Slovenia  
[{matjaz.divjak, zazula}@uni-mb.si](mailto:{matjaz.divjak, zazula}@uni-mb.si)  
<http://storm.uni-mb.si/staff.html>

**Abstract.** To analyze the performance of a vision-based tracking algorithm a good reference information is needed. Magnetic trackers are often used for this purpose, but the inevitable transformation of coordinate systems can result in notable alignment errors. This paper presents an approach for estimating the accuracy of various transformation models as well as individual model parameters. Performance is evaluated numerically and then tested on real data. Results show that the method can be successfully used to analyze the tracking error of free-moving objects.

## 1 Introduction

Performance characterization of vision-based motion tracking systems is becoming an increasingly important task, especially due to a large number of existing tracking algorithms and the speed with which new ones are being presented. This issue is demonstrated by the success of the PETS Workshops [1], the PEIPA's Performance Characterization in Computer Vision project [2], as well as numerous scientific papers [3, 4]. Still, the number of papers that provide comparison with reliable ground-truth data is surprisingly low.

Authors usually manually inspect the video and mark the object positions by hand. This approach is labour intensive, unreliable and difficult to compare. To make it easier, a number of semi-automatic tools are available [5, 6]. While such approximations to ground-truth can be very useful in certain applications, they cannot be regarded as a reliable reference because their absolute accuracy is unknown.

A more reliable approach uses a second motion tracking system with better accuracy as a reference. Magnetic tracking devices and 3D laser scanners seem the most popular choices for this purpose. Magnetic trackers feature high accuracy and speed of measurements, they are not occluded by human body and they have been in use for more than 30 years. Unfortunately, their precision is greatly affected by the presence of ferromagnetic materials and the distance from the transmitter [7, 8]. Therefore, one must take appropriate means to reduce the measurement errors before the actual experiments. On the other hand, laser scanners provide dense and incredibly accurate range measurements which can be easily used as a good reference. Main drawbacks

are the scanning process which takes a fair amount of time (making it inappropriate for moving objects) and the operational area of contemporary devices which is quite limited. Despite that, the authors in [9] claim to be the first to provide a dense ground-truth data registered with respect to the camera.

Our tracking performance analysis is based on comparison of motion trajectories provided by the visual tracking system and the reference tracker. Each trajectory is expressed in its own coordinate system and in order to compare them, they need to be aligned accordingly. This transformation is crucial for a reliable comparison. In this paper we present a method for estimating the accuracy of various transformation approaches as well as individual model parameters. Performance of each model is numerically evaluated and then tested on real-world data. A stereocamera and a magnetic tracker are used to gather motion information.

The remainder of the paper is organized as follows: Section 2 describes the coordinate system transformation problem more formally and introduces analytical transformation models, together with a metric for comparing their performance. Section 3 starts with a comparison of numerical model properties and continues with the presentation of real-world experimental results. In Section 4 the results are compared and discussed, while Section 5 concludes the paper.

## 2 Coordinate System Transformation Models

In order to estimate the performance of a vision-based tracker by comparing it with the electromagnetic tracking device, the target's position must be measured by both systems simultaneously. First, the magnetic sensor is firmly attached to the target object. Each time an image of the scene is captured by the camera, the sensor's position is read and stored into a file, forming a motion trajectory of the object, as detected by the tracking device (a reference trajectory). Afterwards, the video is processed by a tracking algorithm to reconstruct the second trajectory. Both trajectories are expressed in their own coordinate systems (CS). In order to compare them, they need to be transformed into a common CS. Without loss of generality we selected the coordinate system of magnetic tracker ( $CS^M$ ) as the common one.

Obviously, the transformation itself plays a crucial role in performance characterization. If it contains errors, the two trajectories are not aligned properly and the resulting divergence does not convey the differences correctly. The following questions arise: How do we estimate which transformation is better and which is worse? Can we compare the results of two different transformations? Which parameter affects the transformation error the most?

The most frequently used method for comparing the two trajectories is aligning them by some optimization process. This approach completely ignores the possible bias errors and gives little information on how well the tracking algorithm follows the actual movement of the object. For example, if the algorithm would consistently provide exaggerated depth estimations, the “aligned” trajectories could still be a close match.

Another solution to this problem is aligning the two coordinate systems physically by carefully positioning the camera and the magnetic tracker. Although this might seem a fast and simple procedure, the alignment is never perfect and results in consid-

erable errors in the transformation. A quick calculation shows that an orientation error of 1° results in position error of 3.5 cm at a distance of 2 meters from the camera.

Let's assume we have a point in 3D space that needs to be expressed in both coordinate systems. In camera's coordinate system ( $CS^C$ ) we denote it by  $\mathbf{p}^C = (p_1^C, p_2^C, p_3^C, 1)^T$  and in  $CS^M$  by  $\mathbf{p}^M = (p_1^M, p_2^M, p_3^M, 1)^T$ , respectively (using homogenous coordinates). Since both vectors  $\mathbf{p}^M$  and  $\mathbf{p}^C$  represent the same point in space, we can write  $\mathbf{p}^M = \mathbf{A}\mathbf{p}^C$ , where transformation matrix  $\mathbf{A}$  contains the information about translation and rotation of  $CS^C$  with regards to  $CS^M$ . Vector  $\mathbf{O}^C = (o_1, o_2, o_3)^T$  describes the position of camera's origin, while base vectors  $\mathbf{i}^C = (i_1, i_2, i_3)^T$ ,  $\mathbf{j}^C = (j_1, j_2, j_3)^T$  and  $\mathbf{k}^C = (k_1, k_2, k_3)^T$  describe its orientation. Using homogenous coordinates, matrix  $\mathbf{A}$  yields the following structure:

$$\mathbf{A} = \begin{bmatrix} i_1 & j_1 & k_1 & o_1 \\ i_2 & j_2 & k_2 & o_2 \\ i_3 & j_3 & k_3 & o_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Vectors  $\mathbf{i}^C$ ,  $\mathbf{j}^C$ ,  $\mathbf{k}^C$  and  $\mathbf{O}^C$  that define matrix  $\mathbf{A}$  depend on a set of parameters  $\Theta = \{\Theta_l\}$ ,  $l = 1, \dots, N$ . Their exact number depends on the actual transformation model selected. In general, each element  $a_{u,v}$  of matrix  $\mathbf{A}$ ,  $a_{u,v} \in \mathbf{A}$ ,  $\forall u, v \in [1, 2, 3, 4]$ , can be described as a function of those parameters:

$$a_{u,v} = f_{u,v}(\Theta_1, \Theta_2, \dots, \Theta_N). \quad (2)$$

Of course, the camera position information is usually not available, but we could measure it by placing one of the magnetic sensors on the camera and reading its data. This approach has several shortcomings:

- The origin of  $CS^C$  is usually located inside the camera body and is impossible to be measured directly.
- The camera housing is usually metallic and therefore distorts the sensor's electromagnetic field.
- While inaccurate measurements of camera position have a relatively small effect on overall accuracy, the errors in camera orientation can cause significant changes in results.

To address the abovementioned problems and to provide a way for numeric comparison we present three different models for transformation of  $CS^C$  into  $CS^M$ . In all models the magnetic tracker is only used to measure the position of special control points which are used to calculate the orientation of the camera. For a unique solution at least three control points are needed. The following choices will be examined:

- All three control points are measured away from the camera (model A).
- Two points are measured on the camera and one away from it (model B).
- One point is measured on the camera and the other two away from it (model C).

## 2.1 Model A

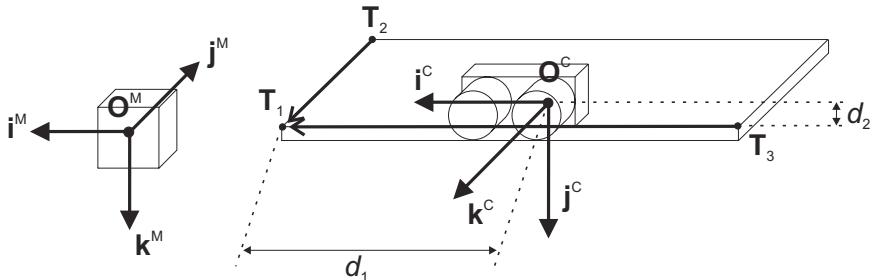
To ensure that camera body does not interfere with measurements, all three control points are measured at a certain distance from it. The camera is fixed to a flat wooden board and aligned as accurately as possible with the board's sides (Fig. 1). Three corners of the board are selected and their coordinates are measured by magnetic sensor to obtain three control points  $\mathbf{T}_1$ ,  $\mathbf{T}_2$  and  $\mathbf{T}_3$ . Since it is assumed that camera's coordinate axes are completely aligned with the board, the base vector  $\mathbf{i}^c$  can be expressed by  $\overline{\mathbf{T}_3\mathbf{T}_1}$ , the base vector  $\mathbf{k}^c$  by  $\overline{\mathbf{T}_2\mathbf{T}_1}$  and the base vector  $\mathbf{j}^c$  is determined by the cross product (Fig. 1):

$$\mathbf{i}^c = \frac{\overline{\mathbf{T}_3\mathbf{T}_1}}{\|\overline{\mathbf{T}_3\mathbf{T}_1}\|}, \mathbf{k}^c = \frac{\overline{\mathbf{T}_2\mathbf{T}_1}}{\|\overline{\mathbf{T}_2\mathbf{T}_1}\|}, \mathbf{j}^c = \mathbf{k}^c \times \mathbf{i}^c. \quad (3)$$

The position of coordinate origin  $\mathbf{O}^c$  in  $CS^M$  is expressed by manually measuring relative distances  $d_1$  and  $d_2$  between control point  $\mathbf{T}_1$  and  $\mathbf{O}^c$  (Fig. 1):

$$\mathbf{O}^c = \mathbf{T}_1 - d_1 \mathbf{i}^c - d_2 \mathbf{j}^c. \quad (4)$$

This way the transformation model A can be completely described by 11 parameters  $\Theta_A = \{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, d_1, d_2\}$ , where  $\mathbf{T}_1 = (x_1, y_1, z_1)$ ,  $\mathbf{T}_2 = (x_2, y_2, z_2)$  and  $\mathbf{T}_3 = (x_3, y_3, z_3)$ .



**Fig. 1.** Magnetic tracker (left) and the stereocamera setup (right) for model A

## 2.2 Model B

The second transformation model neglects the fact that the camera housing disturbs the measurements, because those disturbances are hoped to be very small compared to the errors caused by false orientation data. First, the position of control point  $\mathbf{T}_1$  in front of the left camera lens (Fig. 2) is measured. Next, the magnetic sensor is attached to an arbitrary flat background (control point  $\mathbf{T}_2$  in Fig. 2) and the camera is aligned in such a way that the sensor is visible exactly in the centre of the left stereo image. This step insures that point  $\mathbf{T}_2$  lies on the camera's optical axis. Unfortunately, this alignment is never perfect and a mean error of  $1/2$  pixel should be expected. As a result,  $\mathbf{T}_2$  is displaced from the optical axis by  $\Delta\mathbf{T}_2$ ,  $\mathbf{T}'_2 = \mathbf{T}_2 + \Delta\mathbf{T}_2$ .

Mean displacement  $\Delta\mathbf{T}_2$  can be estimated by considering the view angle of the camera and the orientation of the background with regards to  $CS^C$  (details of the derivation are omitted due to space constraints):

$$\Delta\mathbf{T}_2 = (r, r, r, 1)^T, r = \frac{\|\mathbf{T}_2 - \mathbf{T}_1\|}{2\sqrt{3}f} \sqrt{\left(\frac{9.6}{v_H \cos \alpha}\right)^2 + \left(\frac{7.6}{v_V \cos \beta}\right)^2}. \quad (5)$$

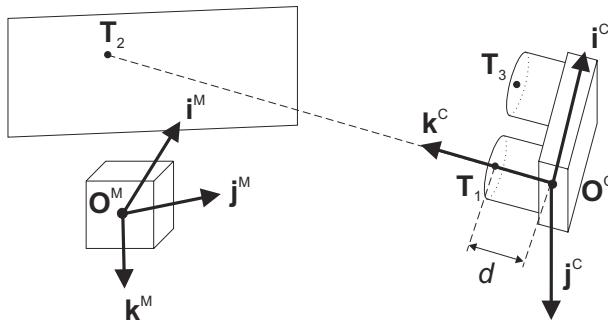
Here,  $f$  denotes the focal length of the lens,  $v_H$  and  $v_V$  are horizontal and vertical dimensions of camera image and  $\alpha$  and  $\beta$  describe the orientation of the background. Since base vector  $\mathbf{k}^C$  has the same direction as the camera's optical axis, we calculate it from  $\mathbf{T}_1$  and  $\mathbf{T}_2'$ . Similarly, base vector  $\mathbf{i}^C$  is obtained by measuring the coordinates of control point  $\mathbf{T}_3$ , positioned in front of the right camera lens (Fig. 2):

$$\mathbf{k}^C = \frac{\overline{\mathbf{T}_1\mathbf{T}_2'}}{\|\overline{\mathbf{T}_1\mathbf{T}_2'}\|}, \mathbf{i}^C = \frac{\overline{\mathbf{T}_1\mathbf{T}_3}}{\|\overline{\mathbf{T}_1\mathbf{T}_3}\|}. \quad (6)$$

Base vector  $\mathbf{j}^C$  is calculated from (3), again. The origin of  $CS^C$  also lies on the camera's optical axis and is determined by displacing the point  $\mathbf{T}_1$  by  $d$  (Fig. 2):

$$\mathbf{O}^C = \mathbf{T}_1 - d\mathbf{k}^C. \quad (7)$$

The transformation model B is therefore described by 12 parameters:  $\Theta_B = \{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, d, \alpha, \beta\}$ .



**Fig. 2.** Stereocamera and magnetic tracker setup for model B

The setup for model C is the same, except that  $\mathbf{T}_3$  is measured anywhere on the background.

### 2.3 Model C

This model is similar to model B, except that control point  $\mathbf{T}_3$  is also measured on the background. The procedure for obtaining base vectors  $\mathbf{k}^C$  and  $\mathbf{j}^C$  is therefore the same. Control point  $\mathbf{T}_3$  should be visible in the left stereo image and should also be dis-

placed by  $\Delta \mathbf{T}_3$  to compensate for the misorientation of the background with regards to  $\text{CS}^C$ ,  $\mathbf{T}'_3 = \mathbf{T}_3 + \Delta \mathbf{T}_3$ .

Vector  $\overline{\mathbf{T}_2 \mathbf{T}'_3}$  is coplanar with the base vector  $\mathbf{i}^C$ , but needs to be rotated around the optical axis to align it completely. The angle of rotation  $\gamma = \arctg(m_H/m_V)$  is determined by horizontal ( $m_H$ ) and vertical ( $m_V$ ) displacement of  $\mathbf{T}_3$  from the centre of the left stereo image. The rotated point  $\mathbf{T}_3''$  is used to calculate the base vector  $\mathbf{i}^C$ :

$$\mathbf{i}^C = \frac{\overline{\mathbf{T}_2 \mathbf{T}_3''}}{\|\overline{\mathbf{T}_2 \mathbf{T}_3''}\|}. \quad (8)$$

The origin of  $\text{CS}^C$  is determined by displacement  $d$ , as in (7), again. The third transformation model is therefore described by 14 parameters:  $\Theta_C = \{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, d, \alpha, \beta, m_H, m_V\}$ .

## 2.4 Comparison of Models

In order to estimate the error of transformation from  $\text{CS}^C$  to  $\text{CS}^M$ , the sensitivity of transformation matrix  $\mathbf{A}$  to the parameter set  $\Theta$  must be determined:

$$\frac{\partial \mathbf{A}}{\partial \Theta_l} = \frac{\partial a_{u,v}}{\partial \Theta_l} = \frac{\partial f_{u,v}(\Theta_1, \Theta_2, \dots, \Theta_N)}{\partial \Theta_l}, \text{ for } \forall u, v \in [1, 2, 3, 4], \forall l \in [1, \dots, N]. \quad (9)$$

Unfortunately, the resulting mathematical expressions are too complex for direct comparison. Instead, the derivatives are estimated numerically using parameter values from our experiments (Section 3). This gives an estimate on the largest contributor to the transformation error. In general, the mutual interaction of parameter errors is unknown, but the overall upper error bound of each model can be estimated.

The magnitude of error amplification for a certain parameter can be expressed by

$$\left\| \frac{\partial \mathbf{p}^M}{\partial \Theta_l} \right\| = \left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \mathbf{p}^C \right\| \leq \left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \right\| \cdot \left\| \mathbf{p}^C \right\|, \quad (10)$$

$$S_l = \frac{\left\| \frac{\partial \mathbf{p}^M}{\partial \Theta_l} \right\|}{\left\| \mathbf{p}^M \right\|} \leq \frac{\left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \right\| \cdot \left\| \mathbf{p}^C \right\|}{\left\| \mathbf{p}^M \right\|}, \text{ for } \forall l = 1, \dots, N. \quad (11)$$

Expression (11) describes the relative sensitivity  $S_l$  of point  $\mathbf{p}^M$  with regards to parameter  $\Theta_l$ . By inserting  $\mathbf{p}^C = \mathbf{A}^{-1} \mathbf{p}^M$  into (11), an expression for calculating relative sensitivity of separate parameters is obtained:

$$S_l = \frac{\left\| \frac{\partial \mathbf{p}^M}{\partial \Theta_l} \right\|}{\left\| \mathbf{p}^M \right\|} \leq \frac{\left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \right\| \cdot \left\| \mathbf{A}^{-1} \mathbf{p}^M \right\|}{\left\| \mathbf{p}^M \right\|} \leq \frac{\left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \right\| \cdot \left\| \mathbf{A}^{-1} \right\| \cdot \left\| \mathbf{p}^M \right\|}{\left\| \mathbf{p}^M \right\|} = \left\| \frac{\partial \mathbf{A}}{\partial \Theta_l} \right\| \cdot \left\| \mathbf{A}^{-1} \right\|, \text{ for } \forall l = 1, \dots, N. \quad (12)$$

Finally, the upper relative sensitivity limit of the whole model ( $S^{\text{MAX}}$ ) equals the sum of separate sensitivities:

$$S^{\text{MAX}} = \frac{\left\| \frac{\partial \mathbf{p}^M}{\partial \Theta} \right\|}{\left\| \mathbf{p}^M \right\|} \leq \left( \left\| \frac{\partial \mathbf{A}}{\partial \Theta_1} \right\| + \left\| \frac{\partial \mathbf{A}}{\partial \Theta_2} \right\| + \dots + \left\| \frac{\partial \mathbf{A}}{\partial \Theta_N} \right\| \right) \cdot \left\| \mathbf{A}^{-1} \right\|. \quad (13)$$

### 3 Experiments

Our approach was tested on real-world data using the Videre Design's STH-MD1-C [10] stereo head and Polhemus' 3Space Fastrak [11] magnetic tracker. Fastrak's static resolution is 0.8 mm RMS for position and 0.15° RMS for orientation when the sensor is within 75 cm from magnetic transmitter. One of Fastrak's sensors was attached to the back of the test subject's right palm. The test subject moved his palm along a predefined, physically limited path so the movement remained largely the same during all of the experiments. Three different video sequences were captured, each consisting of 120 – 200 colour image pairs with  $320 \times 240$  pixels.

To ensure that all transformation models were compared on the same data, the positions of all control points and other model parameters were measured before conducting the experiments. Video data was processed by our algorithm for detection of human hands and faces. 3D centroids of detected regions in each image were tracked with a predictor-corrector based algorithm [12].

Vision-based trajectories were transformed into  $\text{CS}^M$  using transformation matrices of all three presented models. The parameters used to construct the matrices are shown in Table 1 and Table 2. Tables 3, 4 and 5 show the comparison results of various parameter norms. The upper sensitivity limit ( $S^{\text{MAX}}$ ) of each model is presented in Table 6. Final performance of the presented models was estimated by calculating the RMS difference between the coordinates of the two trajectories (Table 7). In Fig. 3 an example of trajectories transformed into  $\text{CS}^M$  is depicted.

**Table 1.** The measured parameter values for models A, B and C

Parameter	$d_1$	$d_2$	$d$	$\alpha$	$\beta$	$m_H$	$m_V$
Value	490 mm	14 mm	48 mm	10°	2°	85 pixels	9 pixels

**Table 2.** An example of the measured coordinates of control points for models A, B and C

Parameter	$x_1$	$y_1$	$z_1$	$x_2$	$y_2$	$z_2$	$x_3$	$y_3$	$z_3$
Model A (mm)	201.4	241.9	91.3	211.2	522.2	100.2	-68.1	266.0	124.5
Model B (mm)	-83.7	204.8	88.9	-62.4	-99.4	-41.7	-97.4	198.8	84.8
Model C (mm)	-83.7	204.8	88.9	-62.4	-99.4	-41.7	107.4	-17.7	-10.6

**Table 3.** Parameter norms for model A

<b>Parameter</b>	$x_1$	$y_1$	$z_1$	$x_2$	$y_2$	$z_2$
$S_l$	398.5	175.2	174.8	1.4	0.7	19.9
<b>Cond. no.</b>	11145.6	7744.0	2716.1	1.9	9.8	14.0
<b>Parameter</b>	$x_3$	$y_3$	$z_3$	$d_1$	$d_2$	
$S_l$	10.6	224.3	224.3	398.6	398.6	
<b>Cond. no.</b>	620.5	14590.5	490.5	1	1	

**Table 4.** Parameter norms for model B

<b>Parameter</b>	$x_1$	$y_1$	$z_1$	$x_2$	$y_2$	$z_2$
$S_l$	410.4	390.7	410.2	20.4	3.4	20.3
<b>Cond. no.</b>	25408.7	4817.2	5326.1	1269.5	54.7	48.1
<b>Parameter</b>	$x_3$	$y_3$	$z_3$	$d$	$\alpha$	$\beta$
$S_l$	0.4	6.9	4.5	390.1	5.2	1.1
<b>Cond. no.</b>	2.3	4.5	1	1	67.4	67.4

**Table 5.** Parameter norms for model C

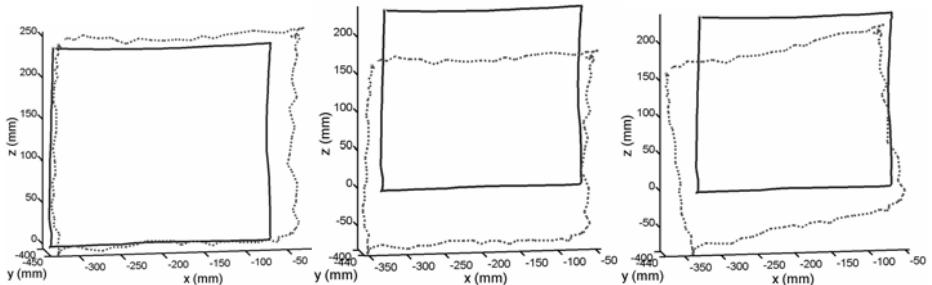
<b>Parameter</b>	$x_1$	$y_1$	$z_1$	$x_2$	$y_2$
$S_l$	408.8	389.1	408.5	20.3	3.4
<b>Cond. no.</b>	1163619.	136461.	134923.	558360.	25446.2
<b>Parameter</b>	$z_2$	$x_3$	$y_3$	$z_3$	$d$
$S_l$	20.2	0.1	0.1	0.8	388.5
<b>Cond. no.</b>	11006.3	1	1	1	1
<b>Parameter</b>	$\alpha$	$\beta$	$m_H$	$m_V$	
$S_l$	5.2	1.1	0.5	4.6	
<b>Cond. no.</b>	771576.9	771576.9	1	1	

**Table 6.** The upper sensitivity limit ( $S^{\text{MAX}}$ ) of models A, B and C

<b>Model</b>	A	B	C
$S^{\text{MAX}}$	2026.9	1661.1	1651.2

**Table 7.** The RMS difference between the coordinates of video-based and magnetic-based trajectories. Mean values for three experiments are shown

<b>Model</b>	<b>x RMS difference (mm)</b>	<b>y RMS difference (mm)</b>	<b>z RMS difference (mm)</b>	<b>Total RMS difference (mm)</b>
A	$16.1 \pm 6.1$	$5.8 \pm 1.2$	$10.7 \pm 1.7$	$20.5 \pm 4.9$
B	$14.8 \pm 2.9$	$12.7 \pm 0.7$	$72.5 \pm 6.4$	$75.1 \pm 6.8$
C	$34.7 \pm 3.2$	$12.7 \pm 0.4$	$55.1 \pm 6.7$	$66.4 \pm 6.9$



**Fig. 3.** Transformation of vision-based trajectories into  $CS^M$ . Magnetic tracker data is depicted by solid lines, the stereocamera data is depicted by dotted lines (model A, model B, model C)

## 4 Discussion

With careful selection of parameters each presented model can be shown to give the best transformation results. However, if a fixed set of parameters is given this limits the tests to one specific setup of stereocamera and Fastrak. All numerical results and conclusions are thus valid for this selected setup only. The methodology is however universally applicable.

A quick glance at Tables 3, 4 and 5 reveals which model parameters are the most sensitive and thus contribute the majority of the transformation error. For model A this is  $T_1$ ,  $T_3$ ,  $d_1$  and  $d_2$ , with  $x_1$ ,  $d_1$  and  $d_2$  being the most sensitive. For models B and C parameters  $T_1$  and  $d$  are the most sensitive. Also the model C has a large number of badly conditioned parameters. Comparison of the estimated  $S^{MAX}$  values shows that models B and C have similar sensitivity, but model A is the most sensitive and, therefore, the most error-prone (Table 6).

However, the results of actual trajectory comparison show a different picture. Transformation with model A resulted in the smallest RMS difference of all three models, while model B is having the worst results (Table 7, Fig. 3). How is this possible?

We have to consider that if a certain parameter has large sensitivity and small actual value, its effect on the transformation can be smaller than from a parameter with small sensitivity and large actual value. The main advantage of model A is that all three control points are measured at a certain distance from the camera (approx. 50 cm in our experiments). If a certain measurement error is made, its effect on the camera orientation is by far smaller than if the same error is made only a few cm away from the camera. The selected parameter set clearly makes models B and C more error-prone than model A. For example, even though parameters  $\alpha$  and  $\beta$  have negligible sensitivity, it is clear that an orientation error of several degrees would have devastating effects on the transformation.

Fastrak's reference measurements also contain certain amount of error (0.8 mm), but since it is much smaller in comparison to video errors, it can be neglected. However, if the magnetic sensor is used outside the perimeter of the highest accuracy, those errors should be taken into consideration and compensated accordingly.

## 5 Conclusion

By analyzing the worst-case sensitivity of various transformation models a limited comparison of those models is possible. The most influential parameters can easily be identified, but the actual parameter values used also have a significant effect on the final transformation error. By careful selection of parameters, any model can be shown to perform the best. Therefore, such comparisons are only reasonable if the parameters are fixed to a certain setup of camera and magnetic tracker.

In our experiments the transformation model A was able to align the vision-based and magnetic-based trajectories reasonably well. The remaining total coordinate difference of 20 mm (RMS) is believed to be caused mostly by our tracking algorithm. Further classification of this error into the errors induced by the transformation and our tracking algorithm would provide more accurate discrepancy estimates, making it possible to use the tracker as an absolute ground-truth reference. However, this task remains for the future work.

## References

1. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). IEEE Winter Vision Multi-Meeting. Breckenridge, Colorado (2005) <http://pets2005.visualsurveillance.org/>
2. Performance Characterization in Computer Vision. PEIPA - Pilot European Image Processing Archive. <http://peipa.essex.ac.uk/index.html>
3. Christensen, H.I., Förstner W.: Performance characteristics of vision algorithms. Machine Vision and Applications, Vol. 9, No. 5-6. Springer-Verlag (1997), 215-218
4. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding, Vol. 73, No. 1. Academic Press (1999) 82-98
5. Doermann, D., Mihalcik, D.: Tools and Techniques for Video Performance Evaluation. Int. Conf. on Pattern Recognition ICPR 2000. Barcelona (2000) 4167 – 4170
6. Black, J., Ellis, T., Rosin, P.: A Novel Method for Video Tracking Performance Evaluation. The Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. Nice (2003) 125-132
7. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. IEEE Trans. Pattern Analysis and Machine Intel., Vol. 22, No. 4, (2000) 322-336
8. Yao, Z., Li, H.: Is A Magnetic Sensor Capable of Evaluating A Vision-Based Face Tracking System? Conf. on Computer Vision and Pattern Recognition Workshop, Vol. 5. Washington (2004)
9. Mulligan, J., Isler, V., Daniilidis, K.: Performance Evaluation of Stereo for Tele-presence. Int. Conf. on Computer Vision ICCV 2001, Vol. 2. Vancouver (2001)
10. Videre Design, STH-MD1/-C Stereo Head. User's Manual (2001)
11. Polhemus Inc., 3Space Fastrak User's Manual (1998)
12. Divjak, M.: 3D Motion Tracking Using a Stereocamera. Master Thesis. Faculty of Electrical Engineering and Computer Science, University of Maribor. Maribor (2003)

# Building Detection from Mobile Imagery Using Informative SIFT Descriptors\*

Gerald Fritz, Christin Seifert, Manish Kumar, and Lucas Paletta

JOANNEUM RESEARCH Forschungsgesellschaft mbH,  
Institute of Digital Image Processing,  
Wastiangasse 6, A-8010 Graz, Austria  
[lucas.paletta@joanneum.at](mailto:lucas.paletta@joanneum.at)

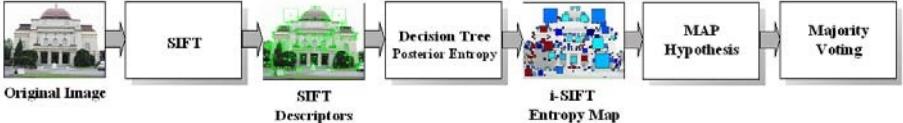
**Abstract.** We propose reliable outdoor object detection on mobile phone imagery from off-the-shelf devices. With the goal to provide both robust object detection and reduction of computational complexity for situated interpretation of urban imagery, we propose to apply the 'Informative Descriptor Approach' on SIFT features (i-SIFT descriptors). We learn an attentive matching of i-SIFT keypoints, resulting in a significant improvement of state-of-the-art SIFT descriptor based keypoint matching. In the off-line learning stage, firstly, standard SIFT responses are evaluated using an information theoretic quality criterion with respect to object semantics, rejecting features with insufficient conditional entropy measure, producing both sparse and discriminative object representations. Secondly, we learn a decision tree from the training data set that maps SIFT descriptors to entropy values. The key advantages of informative SIFT (i-SIFT) to standard SIFT encoding are argued from observations on performance complexity, and demonstrated in a typical outdoor mobile vision experiment on the MPG-20 reference database.

## 1 Introduction

Research on visual object detection has recently focused on the development of local interest operators [7, 9, 11, 6, 5] and the integration of local information into robust object recognition [1, 6]. Recognition from local information serves several purposes, such as, improved tolerance to occlusion effects, or to provide initial evidence on object hypotheses in terms of providing starting points in cascaded object detection. Recently, [11] investigated informative image fragments for object representation and recognition, and [3] applied information theoretic analysis to determine saliency measures in multi-view object recognition. While these approaches performed fundamental analyses on the appearance patterns, the natural extension of improving advanced local detectors by investigating

---

\* This work is supported by the European Commission funded projects MACS under grant number FP6-004381 and MOBVIS under grant number FP6-511051, and by the FWF Austrian Joint research Project Cognitive Vision under sub-projects S9103-N04 and S9104-N04.



**Fig. 1.** Concept for automated building recognition. First, standard SIFT descriptors are extracted within the test image. The proposed informative SIFT (i-SIFT) approach determines the entropy in the descriptor and performs decision making (MAP hypothesizing) only on attended descriptors. Majority voting is then used to integrate local votes into a global classification

about the information content they provide with respect to object discrimination remained open. The Informative Feature Approach is particularly suited for computer vision on emerging technologies, such as, mobile devices, requiring careful outline of algorithms to cope with limited resources, crucial constraints on response times, and complexity in the visual input from real world conditions.

The key contribution of the presented work is (i) to demonstrate a reliable methodology for the application of object detection in mobile phone imagery, and (ii) illustrating that the Informative Feature Approach [3] can perfectly be extended to complex features, such as the SIFT interest point detector [6], to render recognition more efficient. First, we provide a thorough analysis on the discriminative power of complex SIFT features, using local density estimations to determine conditional entropy measures, that makes the actual local information content explicit for further processing. Second, we build up an efficient i-SIFT based representation, using an information theoretic saliency measure to construct a sparse SIFT descriptor based object model. Rapid SIFT based object detection is then exclusively applied to test patterns with associated low entropy, applying an attention filter with a decision tree encoded entropy criterion. We demonstrate that i-SIFT (i) provides better foreground-background discrimination, (ii) significantly reduces the descriptor dimensionality, (iii) decreases the size of object representation by one order of magnitude, and (iv) performs matching exclusively on attended descriptors, rejecting the majority of irrelevant descriptors.

The experiments were performed on raw mobile phone imagery on urban tourist sights under varying environment conditions (changes in scale, viewpoint, and illumination, severe degrees of partial occlusion). We demonstrate in this challenging outdoor object detection task the superiority in using informative SIFT (i-SIFT) features to standard SIFT, by increased reliability in foreground/background separation, and the significant speedup of the algorithm by one order of magnitude, requiring a fraction ( $\approx 20\%$ ) of features of lower dimensionality (30%) for representation.

## 2 Informative Local Descriptors

The ~~.....~~ requires to extract relevant features in a pre-processing stage to recognition, by optimizing feature selection with respect

to the information content in the context of a specific task, e.g., object recognition. In the following sections, we motivate the determination of informative descriptors from information theory (Sec. 2.1), and describe the application to local SIFT descriptors (Sec. 2.2).

## 2.1 Local Information Content

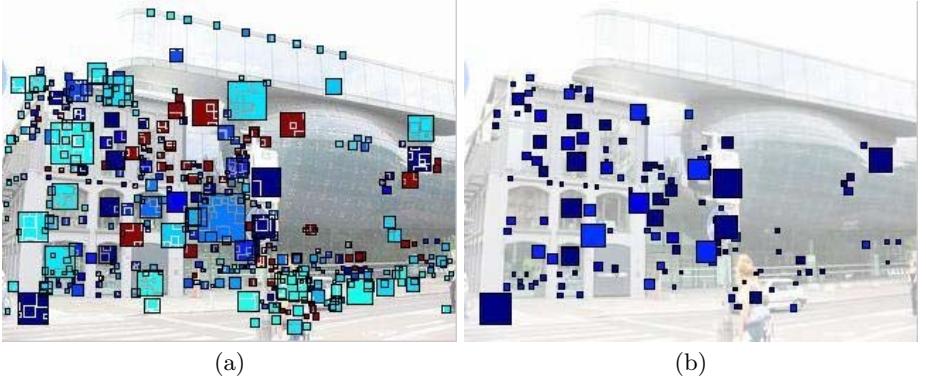
Saliency of interest points has been attributed due to various interpretations, such as, from sample densities within local neighborhoods [4], or according to the class specific general frequency of occurrence in recognition [11]. The . . . . . determines the information content from a posterior distribution with respect to given task specific hypotheses. In contrast to costly . . . optimization, we expect that it is sufficiently accurate to estimate a . . . information content, by computing it from the posterior distribution within a sample test point's local neighborhood in feature space.

We are primarily interested to get the . . . . . of any sample local descriptor  $\mathbf{f}_i$  in feature space  $\mathcal{F}$ ,  $\mathbf{f}_i \in \mathcal{R}^{|\mathcal{F}|}$ , with respect to the task of object recognition, where  $o_i$  denotes an object hypothesis from a given object set  $\Omega$ . For this, we need to estimate the entropy  $H(O|\mathbf{f}_i)$  of the posterior distribution  $P(o_k|\mathbf{f}_i)$ ,  $k = 1 \dots \Omega$ ,  $\Omega$  is the number of instantiations of the object class variable  $O$ . The Shannon conditional entropy denotes  $H(O|\mathbf{f}_i) \equiv -\sum_k P(o_k|\mathbf{f}_i) \log P(o_k|\mathbf{f}_i)$ . We approximate the posteriors at  $\mathbf{f}_i$  using only samples  $\mathbf{g}_j$  inside a Parzen window of a local neighborhood  $\epsilon$ ,  $\|\mathbf{f}_i - \mathbf{g}_j\| \leq \epsilon$ ,  $j = 1 \dots J$ . We weight the contributions of specific samples  $\mathbf{f}_{j,k}$  - labeled by object  $o_k$  - that should increase the posterior estimate  $P(o_k|\mathbf{f}_i)$  by a Gaussian kernel function value  $\mathcal{N}(\mu, \sigma)$  in order to favor samples with smaller distance to observation  $\mathbf{f}_i$ , with  $\mu = \mathbf{f}_i$  and  $\sigma = \epsilon/2$ . The estimate about the Shannon conditional entropy  $\hat{H}(O|\mathbf{f}_i)$  provides then a measure of ambiguity in terms of characterizing the information content with respect to object identification within a single local observation  $\mathbf{f}_i$ . Fig. 2 depicts . . . . . in an entropy-coded representation of local SIFT features  $\mathbf{f}_i$ .

From discriminative descriptors we proceed to . . . . . object . . . . ., providing increasingly sparse representations with increasing recognition accuracy, in terms of storing only . . . . . descriptor information that is . . . . . purposes, i.e., those  $\mathbf{f}_i$  with  $\hat{H}(O|\mathbf{f}_i) \leq \Theta$ . A specific choice on the threshold  $\Theta$  consequently determines both storage requirements and recognition accuracy (Sec. 4). To speed up the matching we use efficient memory indexing of nearest neighbor candidates described by the adaptive tree method [2].

## 2.2 Informative SIFT Descriptors

We apply the . . . . . on SIFT based descriptors that are among the best local descriptors with respect to matching distinctiveness, invariance to blur, image rotation, and illumination changes [8]. However, critical bottlenecks in SIFT based recognition are identified as performing extensive



**Fig. 2.** Informative descriptors for saliency

SIFT keypoint matching with high computational complexity due to the nearest neighbor indexing, and the lack of any representation of uncertainty to enable approximate reasoning. We apply informative feature selection to the SIFT descriptor with the aim to significantly decrease the computational load using attentive matching, while attaining improved detection accuracy, and providing a probabilistic framework for individual SIFT descriptors.

Descriptors of the Scale Invariant Feature Transform (SIFT [6]) are invariant to image scale and rotation, in addition, they show robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. [6] further augments the SIFT descriptor into a distinctive feature approach, by proposing specific matching and descriptor representation methods techniques for object recognition. While the informative SIFT (i-SIFT) approach will specifically improve the matching and representation procedures regarding the complete SIFT approach, any consecutive recognition methodology mentioned in [6] might be applied as well to i-SIFT, such as, using the Hough transform to identify clusters belonging to a single object, etc.

The application of the i-SIFT approach tackles three key aspects of SIFT estimation: (i) reducing the high dimensionality (128 features) of the SIFT keypoint descriptor, (ii) thinning out the number of training keypoints using posterior entropy thresholding (Sec. 2.1), in order to obtain an informative and sparse object representation, and (iii) providing an entropy sensitive matching method to reject non-informative outliers, described in more detail as follows,

1. Reducing the dimensionality (128 features) of the SIFT descriptor is crucial to keep nearest neighbor indexing computationally feasible. Possible solutions are K-d and Best-Bin-First search ([6]) that practically perform by  $\mathcal{O}(ND)$ , with  $N$  training prototypes composed of  $D$  features.

To discard statistically irrelevant feature dimensions, we applied Principal Component Analysis (PCA) on the SIFT descriptors. This is in contrast to the PCA-SIFT method [5], where PCA is applied to the normalized gradient pattern, but that also becomes more errorprone under illumination changes [8].

2. According to the Informative Descriptor Approach (Sec. 2.1) we exclusively select local SIFT descriptors for object representation. The degree of reduction in the number of training descriptors is determined by threshold  $\Theta$  for accepting sufficiently informative descriptors. In the experiments (Sec. 4) this approximately reduces the representation size by one order of magnitude. avoids a general cut-off determined by the posterior entropy measure but attributes to each objects its partition  $1/|\Omega|N_{sel}$  of a predetermined total number  $N_{sel}$  of to-be selected descriptors. This prevents the method from associating too few SIFT descriptors to a corresponding object representation.
3. in nearest neighbor indexing is then necessary as a means to reject outliers in analyzing test images. Any test descriptor  $\mathbf{f}_*$  will be rejected from matching if it comes not close enough to any training descriptor  $\mathbf{f}_i$ , i.e., if  $\forall \mathbf{f}_i : |\mathbf{f}_i - \mathbf{f}_*| < \epsilon$ , and  $\epsilon$  was determined so as to optimize posterior distributions with respect to overall recognition accuracy (Sec. 2.1).

### 3 Attentive Object Detection

This section outlines a framework for object detection that enables performance comparison between SIFT and i-SIFT based local descriptors. i-SIFT based object detection can achieve a significant speedup from attentive filtering for the rejection of less promising candidate descriptors. This attentive mapping of low computational complexity is described in terms of a decision tree which learns its tree structure from examples, requiring very few attribute comparisons to decide upon acceptance or rejection of a SIFT descriptor under investigation.

To enable direct performance comparison between SIFT and i-SIFT based object recognition, we determine an adapted majority voting procedure to decide upon a preferred object hypothesis. Detection tasks require the rejection of images whenever they do not contain any objects of interest. For this we consider to estimate the entropy in the posterior distribution - obtained from a normalized histogram of the object votes - and reject images with posterior entropies above a predefined threshold. The proposed recognition process is characterized by an entropy driven selection of image regions for classification, and a voting operation, as follows,

1. **Mapping** of local patterns into descriptor subspace.
2. **Probabilistic interpretation** to determine local information content and associated entropy measure.

3. **Rejection** of descriptors contributing to ambiguous information.
4. **Nearest neighbor analysis** of selected imagettes within  $\epsilon$ -environment .
5. **Majority voting** for object identifications over a region of interest.

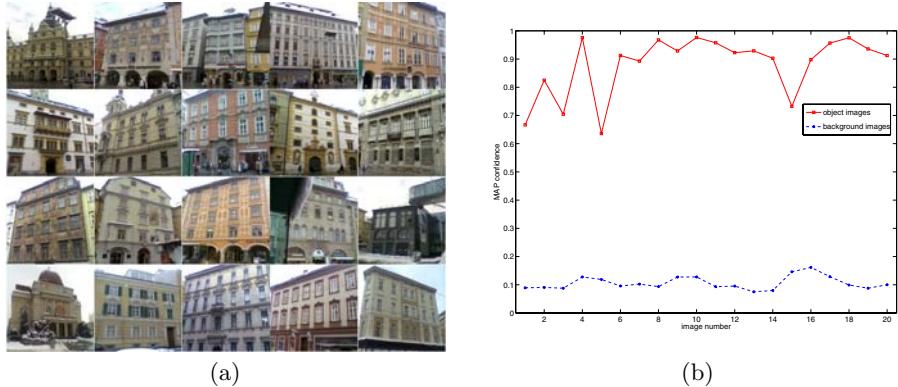
Each pattern from a test image that is mapped to SIFT descriptor features is analyzed for its conditional entropy with respect to the identification of objects  $o_i \in O$ . An entropy threshold  $\Theta$  for rejecting ambiguous test descriptors in eigenspace is most easily identical with the corresponding threshold applied to get a sparse model of reference points. Object recognition on a collection of (matched and therefore labelled) SIFT descriptors is then performed on finding the object identity by majority voting on the complete set of class labels attained from individual descriptor interpretations.

For a rapid estimation of local entropy quantities, the descriptor encoding is fed into the decision tree which maps SIFT descriptors  $\mathbf{f}_i$  into entropy estimates  $\hat{H}$ ,  $\mathbf{f}_i \mapsto \hat{H}(\Omega|\mathbf{f}_i)$ . The C4.5 algorithm [10] builds a decision tree using the standard top-down induction of decision trees approach, recursively partitioning the data into smaller subsets, based on the value of an attribute. At each step in the construction of the decision tree, C4.5 selects the attribute that maximizes the information gain ratio. The induced decision tree is pruned using pessimistic error estimation [10]. The extraction of informative SIFTS (i-SIFTS) in the image is performed in two stages. First, the decision tree based entropy estimator provides a rapid estimate of local information content of a SIFT key under investigation. Only descriptors  $\mathbf{f}_i$  with an associated entropy below a predefined threshold  $\hat{H}(O|\mathbf{f}_i) < \Theta$  are considered for recognition. Only these selected discriminative descriptors are then processed by nearest neighbor analysis, with respect to the object models, and interpreted via majority voting.

There are several issues in using i-SIFT attentive matching that significantly ease the resulting computational load, showing improvements along several dimensions. Firstly, information theoretic selection of candidates for object representation experimentally . . . . . of the object representation of up to . . . . . , thus supporting sparse representations on devices with limited resources, such as, mobile vision enhanced devices. Secondly, the reduction of dimensionality in the SIFT descriptor representation may in addition . . . . .  $\leq 30\%$ . Finally, the attentive decision tree based mapping is applied to reject SIFT descriptors for further analysis, thereby . . . . .  $\leq 20\%$  SIFT descriptors for further analysis.

## 4 Experiments

Targeting emerging technology applications using computer vision on mobile devices, we perform the performance tests on mobile phone imagery captured



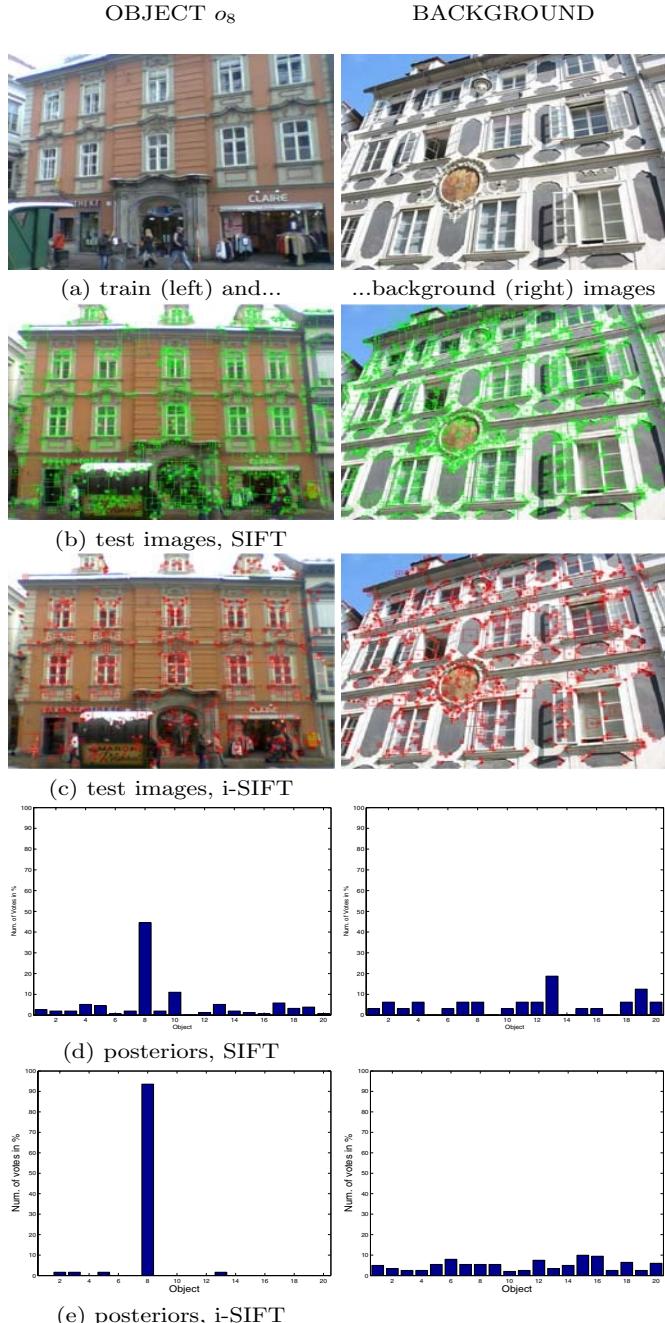
**Fig. 3.** The MPG-20 database, consisting of mobile phone images from 20 buildings (numbered  $o_1$ – $o_{20}$  from top-left to bottom-right) in the city of Graz (displayed images were used for training, see Sec. 4). (b) i-SIFT outperforming SIFT supported MAP confidence based discrimination between object and background

about tourist sights in the urban environment of the city of Graz, Austria, i.e., from the MPG-20 database (see below, Fig. 3). In order to evaluate the improvements gained from the 'Informative Descriptor Approach', we compare the performance between the standard SIFT key matching and the i-SIFT attentive matching.

The MPG-20 database<sup>1</sup> includes images from 20 objects, i.e., facades of buildings from the city of Graz, Austria. Most of these images contain a tourist sight, together with 'background' information from surrounding buildings, pedestrians, etc. The images were captured from an off-the-shelf camera phone (Nokia 6230) of resolution  $640 \times 480$ , containing severe changes in 3D viewpoint, partial occlusions, scale changes by varying distances for exposure, and various illumination changes due to different weather situations and changes in daytime. For each object, we then selected 2 images taken by a viewpoint change of  $\approx \pm 30^\circ$  of a similar distance to the object for training to determine the i-SIFT based object representation. 2 additional views - two different front views of distinct distance and therefore significant scale change - were taken for test purposes, giving 40 test images in total. Further images (noted in separation) were obtained from 'background', such as, other buildings, landscape, etc., and from objects under severe illumination conditions (in the evening, Christmas lighting, etc.).

The training images were bit-masked by hand, such that SIFT descriptors on background information (cars, surrounding

<sup>1</sup> The MPG-20 (Mobile Phone imagery Graz) database can be downloaded at the URL <http://dib.joanneum.at/cape MPG-20>.



**Fig. 4.** Sample object detection results for object  $o_8$  (left) and background (right), (a) depicting train images, (b) SIFT descriptor locations on test images, (c) selected i-SIFT descriptors, (d) posterior distribution on object hypotheses from SIFT and (e) i-SIFT descriptors, demonstrating more distinctive results for i-SIFT based interpretation

**Table 1.** Performance comparison between *standard SIFT* keypoint matching [6] and *i-SIFT* attentive matching on MPG-20 mobile imagery

Recognition Method	MAP accuracy MPG-20 [%]	PT [%]	PF [%]	obj $\bar{H}$	bgd $\bar{H}$	obj avg. MAP	bgd avg. MAP
SIFT	95.0	82.5	0.1	3.0	3.4	43.9	18.7
i-SIFT	97.5	100.0	0.0	0.5	4.1	88.0	10.6

buildings, pedestrians) were discarded. In total 28873 SIFT descriptors were determined for the 40 training images, 722 on average. The 40 (non-masked) test images generated a similar number of SIFT descriptors per image. For each of these descriptors the distances to the closest nearest neighbor ( $d_1$ ) and the 2nd closest neighbor ( $d_2$ ) was calculated. If the distance ratio ( $\frac{d_1}{d_2}$ ) was greater than 0.8 the sift descriptor remained unlabeled (as described in [6], otherwise the label of the closest nearest neighbor was assigned. Thus, on average  $\approx 30\%$  of the SIFT features were retained for voting. Object recognition is then performed using majority voting. The average entropy in the posterior of the normalized voting histograms was  $H_{avg} \approx 3.0$ . A threshold of 25% in the MAP hypothesis confidence was used as decision criterion to discriminate between object ( $> 25\%$ ) and background ( $\leq 25\%$ ) images (for both SIFT and i-SIFT).

For the training of the i-SIFT selection, the SIFT descriptor was projected to an eigenspace of dimension 40, thereby decreasing the original descriptor input dimensionality (128 features) by a factor of three. The size  $\epsilon$  of the Parzen window for local posterior estimates was chosen 0.4, and 175 SIFT keys per object were retained for object representation. The threshold on the entropy criterion for attentive matching was defined by  $\Theta = 1.0$ . In total, the number of attended SIFT descriptors was 3500, i.e.,  $\approx 12.1\%$  of the total number that had to be processed by standard SIFT matching. The recognition accuracy according to MAP (Maximum A Posteriori) classification was 100%, the average entropy in the posterior distribution was  $H_{avg} \approx 0.5$ , in very analogy to the value achieved by standard SIFT matching (see above).

Table 1 illustrates the results of the MPG-20 experiments, and the results of a comparison between standard SIFT keypoint matching and i-SIFT attentive matching.

## 5 Summary and Conclusions

The presented work proposed a methodology for reliable urban object detection from off-the-shelf mobile phone imagery. We applied the

significantly improving the efficiency in object detection, both with respect to memory resources and to speedup the recognition process. The paper also introduces using the informative SIFT (i-SIFT) descriptors, applying an information theoretic criterion for the selection of discrim-

inative SIFT descriptors. Matching with the i-SIFT descriptor (i) significantly reduces the dimensionality of the descriptor encoding, (ii) provides sparse object representations that reduce storage by one order of magnitude with respect to standard SIFT, and (iii) enables attentive matching by requiring 4–8 times less SIFT features per image to be identified by more costly nearest neighbor search.

This innovative local descriptor is most appropriate for sensitive operation under limited resources, such as, in mobile devices. We evaluated the performance of the i-SIFT descriptor on the public available MPG-20 database, including images from 20 building objects and ‘non-object background’ pictures from the city of Graz. The i-SIFT did not only compare well with the high recognition accuracy when using standard keypoint matching, but also provided discriminative posterior distributions, robust background detection, and - as surplus - significant speedup in processing times.

## References

1. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2003.
2. J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.
3. G. Fritz, L. Paletta, and H. Bischof. Object recognition using local information content. In *Proc. International Conference on Pattern Recognition, ICPR 2004*, volume II, pages 15–18. Cambridge, UK, 2004.
4. D. Hall, B. Leibe, and B. Schiele. Saliency of interest points under scale changes. In *Proc. British Machine Vision Conference, BMVC 2002*, Cardiff, UK, 2002.
5. Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proc. Computer Vision and Pattern Recognition, CVPR 2004*, volume 2, pages 506–513, Washington, DC, 2004.
6. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
7. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conference on Computer Vision*, pages 128–142, 2002.
8. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. <http://www.robots.ox.ac.uk/~vgg/research/affine/>, 2004.
9. S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. British Machine Vision Conference*, pages 113–122, 2002.
10. J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
11. M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *Proc. International Conference on Computer Vision, ICCV 2003*, pages 281–288. Nice, France, 2003.

# Perception-Action Based Object Detection from Local Descriptor Combination and Reinforcement Learning\*

Lucas Paletta, Gerald Fritz, and Christin Seifert

JOANNEUM RESEARCH Forschungsgesellschaft mbH,  
Institute of Digital Image Processing,  
Wastiengasse 6, A-8010 Graz, Austria  
[lucas.paletta@joanneum.at](mailto:lucas.paletta@joanneum.at)

**Abstract.** This work proposes to learn visual encodings of attention patterns that enables sequential attention for object detection in real world environments. The system embeds a saccadic decision procedure in a cascaded process where visual evidence is probed at informative image locations. It is based on the extraction of information theoretic saliency by determining informative local image descriptors that provide selected foci of interest. The local information in terms of code book vector responses and the geometric information in the shift of attention contribute to recognition states of a Markov decision process. A Q-learner performs then performs search on useful actions towards salient locations, developing a strategy of action sequences directed in state space towards the optimization of information maximization. The method is evaluated in outdoor object recognition and demonstrates efficient performance.

## 1 Introduction

Recent research in neuroscience [2] and experimental psychology [5] has confirmed evidence that decision behavior plays a dominant role in human selective attention in object and scene recognition . E.g., there is psychophysical evidence that human observers represent visual scenes not by extensive reconstructions but merely by purposive encodings via attention patterns [9] of few relevant scene features, leading to the assumption of transsaccadic object memories. Current biologically motivated computational models on sequential attention identify shift invariant descriptions across saccade sequences, and reflect the encoding of scenes and relevant objects from saccade sequences in the framework of neural network modeling [9] and probabilistic decision processes [1].

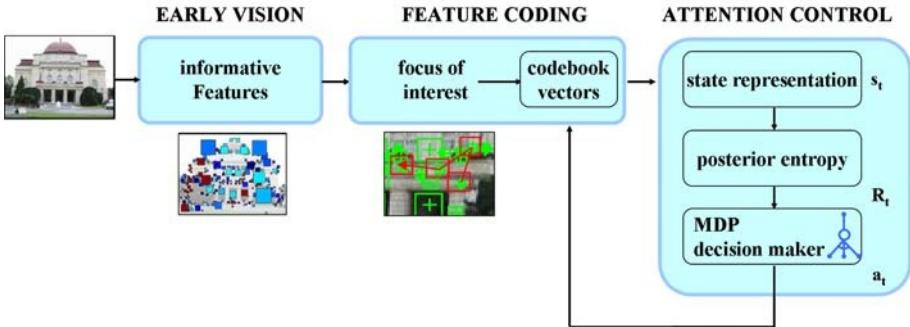
---

\* This work is supported by the European Commission funded projects MACS under grant number FP6-004381 and MOBVIS under grant number FP6-511051, and by the FWF Austrian Joint research Project Cognitive Vision under sub-projects S9103-N04 and S9104-N04.

In computer vision, recent research has been focusing on the integration of information received from single local descriptor responses into a more global analysis with respect to object recognition [7]). State-of-the-art solutions, such as, (i) identifying the MAP hypothesis from probabilistic histograms [3], (ii) integrating responses in a statistical dependency matrix [12], and (iii) collecting evidence for object and view hypotheses in parametric Hough space [7], provide convincing performance under assumptions, such as, statistical independence of the local responses, excluding segmentation problems by assuming single object hypotheses in the image, or assuming regions with uniformly labelled operator responses. An integration strategy closing methodological gaps when above assumptions are violated should therefore (i) cope with statistical dependency between local features of an object, (ii) enable to segment multiple targets in the image and (iii) provide convincing evidence for the existence of object regions merely on the geometry than on the relative frequency of labelled local responses.

The original contribution of this work is to provide a scalable framework for cascaded sequential attention in real-world environments. Firstly, it proposes to integrate local information only at locations that are relevant with respect to an information theoretic saliency measure. Secondly, it enables to apply efficient strategies to group informative local descriptors using a decision maker. The decision making agent used Q-learning to associate actions to cumulative reward with respect to a task goal, i.e., object recognition. Objects are represented in a framework of perception-action, providing a transsaccadic memory that stores useful grouping strategies of a kind of behavior. In object recognition terms, this method enables to match not only between local feature responses, but also taking the geometrical relations between the specific features into account, thereby defining their more global visual configuration. The proposed method is outlined in a perception-action framework, providing a sensorimotor decision maker that selects appropriate saccadic actions to focus on target descriptor locations. The advantage of this framework is to become able to start interpretation from a single local descriptor and, by continuously and iteratively integrating local descriptor responses 'on the fly', being capable to evaluate the complete geometric configuration from a set of few features.

The saccadic decision procedure is embedded in a cascaded recognition process (Fig. 1) where visual evidence is probed exclusively at salient image locations. In a first processing stage, salient image locations are determined from an entropy based cost function on object discrimination. Local information in terms of code book vector responses determine the recognition state in the Markov Decision Process (MDP). In the training stage, the reinforcement learner performs trial and error search on useful actions towards salient locations within a neighborhood, receiving reward from entropy decreases. In the test stage, the decision maker demonstrates feature grouping by matching between the encountered and the trained saccadic sensorimotor patterns. The method is evaluated in experiments on object recognition using the reference COIL-20 (indoor imagery) and



**Fig. 1.** Concept of the proposed perception-action system for object recognition. The Early Vision module extracts informative SIFT descriptors from the input image and associates codebook vectors. Sequential attention operates on the geometry between these vectors and statistically reinforces promising feature-action configurations

the TSG-20 object (outdoor imagery) database, proving the method being computationally feasible and providing rapid convergence in the discrimination of objects.

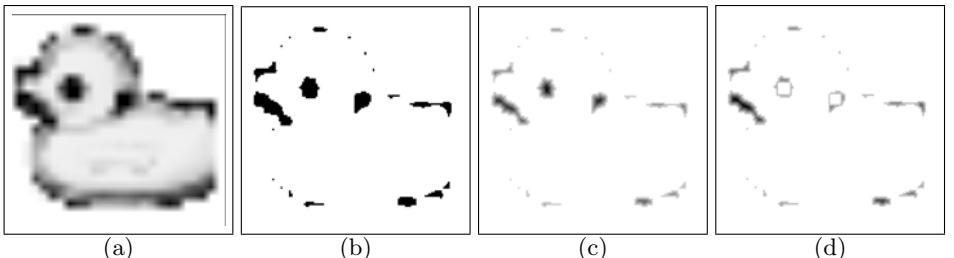
## 2 Informative Foci of Interest for Object Detection

In the proposed method, attention on informative local image patterns is shifted between the largest local maxima derived from a local feature saliency map (Fig. 3). Informative features are selected using an information theoretic saliency measure on local descriptor patterns as described in detail. The following sections describe the informative feature method from [3] and relate the resulting saliency map to the sequential attention approach.

We determine the information content from a posterior distribution with respect to given task specific hypotheses. In contrast to costly optimization, we expect that it is sufficiently accurate to estimate information content, by computing it from the posterior distribution within a sample test point's local neighborhood in feature space [3]. The object recognition task is applied to sample local descriptors  $\mathbf{f}_i$  in feature space  $\mathcal{F}$ ,  $\mathbf{f}_i \in \mathcal{R}^{|\mathcal{F}|}$ , where  $o_i$  denotes an object hypothesis from a given object set  $\Omega$ . We need to estimate the entropy  $H(O|\mathbf{f}_i)$  of the posteriors  $P(o_k|\mathbf{f}_i)$ ,  $k = 1 \dots \Omega$ ,  $\Omega$  is the number of instantiations of the object class variable  $O$ . Shannon conditional entropy denotes  $H(O|\mathbf{f}_i) \equiv -\sum_k P(o_k|\mathbf{f}_i) \log P(o_k|\mathbf{f}_i)$ . We approximate the posteriors at  $\mathbf{f}_i$  using only samples  $\mathbf{g}_j$  inside a Parzen window of a local neighborhood  $\epsilon$ ,  $\|\mathbf{f}_i - \mathbf{f}_j\| \leq \epsilon$ ,  $j = 1 \dots J$ . We weight the contributions of specific samples  $\mathbf{f}_{j,k}$  - labeled by object  $o_k$  - that should increase the posterior estimate  $P(o_k|\mathbf{f}_i)$  by a Gaussian kernel function value  $\mathcal{N}(\mu, \sigma)$  in order to favor samples with smaller distance to observation  $\mathbf{f}_i$ , with  $\mu = \mathbf{f}_i$  and  $\sigma = \epsilon/2$ .

The estimate about the conditional entropy  $\hat{H}(O|\mathbf{f}_i)$  provides then a measure of ambiguity in terms of characterizing the information content with respect to object identification within a single local observation  $\mathbf{f}_i$ . We receive sparse instead of extensive object representations, in case we store only descriptor information that is purposes, i.e.,  $\mathbf{f}_i$  with  $\hat{H}(O|\mathbf{f}_i) \leq \Theta$ . A specific choice on the threshold  $\Theta$  consequently determines both storage requirements and recognition accuracy. For efficient memory indexing of nearest neighbor candidates we use the adaptive tree method. The local patterns are projected into eigenspace, a Parzen window approach is used to estimate the local posterior distribution  $P(o_k|\mathbf{g}_i)$ , given eigencoefficient vector  $\mathbf{g}_i$  and object hypothesis  $o_k$ . The information content in the pattern is computed from the Shannon entropy in the posterior. These features support attention on most salient, i.e., informative image regions for further investigation [4].

Attention on informative local image patterns is shifted between largest local maxima derived by the information theoretic saliency measure. Saccadic actions originate from a randomly selected maximum and target towards one of n-best ranked maxima – represented by a focus of interest (FOI) – in the saliency map. At each local maximum, the extracted local pattern is associated to a codebook vector of nearest distance in feature space. Fig. 2 depicts the principal stages in selecting the FOIs. From the saliency map (a), one computes a binary mask (b) that represents the most informative regions with respect to the conditional entropy, by selecting each pixels contribution to the mask from whether its entropy value  $H$  is smaller than a predefined entropy threshold  $H_\Theta$ , i.e.,  $H < H_\Theta$ . (c) applying a distance transform on the binary regions of interest results mostly in the accurate localization of the entropy minimum. The maximum of the local distance transform value is selected as FOI. Minimum entropy values and maximum transform values are combined to give a location of interest for the first FOI, applying a 'Winner-



**Fig. 2.** Extraction of FOI (focus of interest) from an information theoretic saliency measure map. (a) Saliency map from the entropy in the local appearances (9 × 9 pixel window). (b) Binary mask from a thresholded entropy map representing most informative regions ( $H_\Theta = 0.2, H < H_\Theta$  white pixels). (c) Distance transform on most informative regions. (d) Inhibition of return for the first 2 FOIs (black regions in informative areas) for maximum saliency extraction from WTA (winner-takes-all) computation [6]

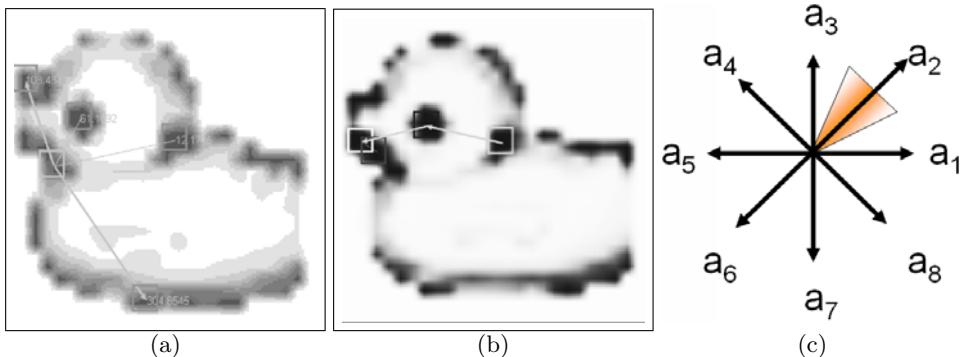
takes-it-all' (WTA) principle [6]. (d) Masking out the selected maximum of the first FOI, one can apply the same WTA rule, selecting the maximum saliency. This masking is known as 'inhibition of return' in the psychology of visual attention [10].

### 3 Sensory-Motor Patterns of Sequential Attention

Sequential attention shifts the focus of attention in the ranked order of maximum saliency, providing an integration of the visual information in the sampled focused attention windows. In the proposed method, saccadic actions operate on  $n$  best-ranked maxima (e.g.,  $n=5$  in Fig. 3a) of the information theoretic saliency map. At each local maximum, the extracted local pattern  $\mathbf{g}_i$  is associated to a codebook vector  $\Gamma_j$  of nearest distance  $d = \arg \min_j \|\mathbf{g}_i - \Gamma_j\|$  in feature space. The codebook vectors were estimated from k-means clustering of a training sample set  $G = \mathbf{g}_1, \dots, \mathbf{g}_N$  of size  $N$  ( $k = 20$  in the experiments). The focused local information patterns (in Fig. 3b: the appearance patterns) are therefore associated and thereby represented by prototype vectors, gaining discrimination mainly from the geometric relations between descriptor encodings (i.e, the label of the associated codebook vector) to discriminate saccadic attention patterns. Saccadic actions originate from a randomly selected local maximum of saliency and target towards one of the remaining ( $n-1$ ) best-ranked maxima via a saccadic action  $a \in A$  (Fig. 3a). The individual action and its corresponding angle  $\alpha(x, y, a)$  is then categorized into one out of  $|A| = 8$  principal directions ( $\Delta a = 45^\circ$ ) (Fig. 3c).

An individual state  $s_i$  of a saccadic pattern of length  $N$  is finally represented by the sequence of descriptor encodings  $\Gamma_j$  and actions  $a \in A$ , i.e.,

$$s_i = (\Gamma_{n-N}, a_{n-N-1}, \dots, \Gamma_{n-1}, a_n, \Gamma_n). \quad (1)$$



**Fig. 3.** Saccadic attention pattern. (a) Saccadic actions originating in a FOE, directed towards 4 possible target FOIs. (b) Learned attention pattern (scanpath) to recognize the object. (c) Discretization of the angular encoding for shifts of attention

Within the object learning stage, random actions will lead to arbitrary descriptor-action sequences. For each sequence pattern, we protocol the number of times it was experienced per object in the database. From this we are able to estimate a mapping from states  $s_i$  to posteriors, i.e.,  $s_i \mapsto P(o_k|s_i)$ , by monitoring how frequent states are visited under observation of particular objects. From the posterior we compute the conditional entropy  $H_i = H(O|s_i)$  and the

with respect to actions leading from state  $s_{i,t}$  to  $s_{j,t+1}$  by  $\Delta H_{t+1} = H_t - H_{t+1}$ . An efficient strategy aims then at selecting in each state  $s_{i,t}$  exactly the action  $a^*$  that would maximize the information gain  $\Delta H_{t+1}(s_{i,t}, a_{k,t+1})$  received from attaining state  $s_{j,t+1}$ , i.e.,  $a^* = \arg \max_a \Delta H_{t+1}(s_{i,t}, a_{k,t+1})$ .

## 4 Q-Learning of Attentive Saccades

In each state of the sequential attention process, a decision making agent is asked to select an actin to drive its classifier towards a reliable decision. Learning to recognize objects means then to explore different descriptor-action sequences, to quantify consequences in terms of a utility measure, and to adjust the control strategy thereafter.

The Markov decision process (MDP [8]) provides the general framework to outline sequential attention for object recognition in a multistep decision task with respect to the discrimination dynamics. A MDP is defined by a tuple  $(\mathcal{S}, \mathcal{A}, \delta, \mathcal{R})$  with state recognition set  $\mathcal{S}$ , action set  $\mathcal{A}$ , probabilistic transition function  $\delta$  and reward function  $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \Pi(\mathcal{S})$  describes a probability distribution over subsequent states, given the attention shai8ft action  $a \in \mathcal{A}$  executable in state  $s \in \mathcal{S}$ . In each transition, the agent receives reward according to  $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto R$ ,  $R_t \in R$ . The agent must act to maximize the utility  $Q(s, a)$ , i.e., the expected discounted reward  $Q(s, a) \equiv U(s, a) = E[\sum_{n=0}^{\infty} \gamma^n \mathcal{R}_{t+n}(s_{t+n}, a_{t+n})]$ , where  $\gamma \in [0, 1]$  is a constant controlling contributions of delayed reward.

We formalize a sequence of action selections  $a_1, a_2, \dots, a_n$  in sequential attention as a MDP and are searching for optimal solutions with respect to the object recognition task. In the posterior distribution on object hypotheses, the information gain received from attention shift  $a$   $\mathcal{R}(s, a) := \Delta H$ . Since the probabilistic transition function  $\Pi(\cdot)$  cannot be known beforehand, the probabilistic model of the task is estimated via reinforcement learning, e.g., by Q-learning [11] which guarantees convergence to an optimal policy applying sufficient updates of the Q-function  $Q(s, a)$ , mapping recognition states  $s$  and actions  $a$  to utility values. The Q-function update rule is

$$Q(s, a) = Q(s, a) + \alpha [R + \gamma (\max_{a'} Q(s', a') - Q(s, a))], \quad (2)$$

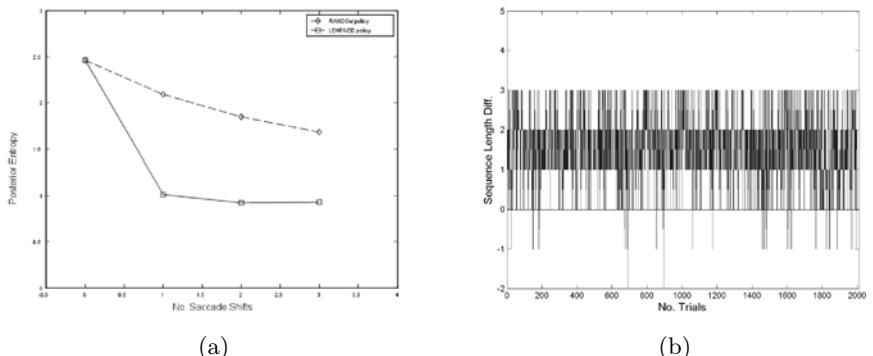
where  $\alpha$  is the learning rate,  $\gamma$  controls the impact of a current shift of attention action on future policy return values. The decision process in sequential attention is determined by the sequence of choices on shift actions at specific focus of interest (FOI). In response to the current visual observation represented by the

local descriptor and the corresponding history, i.e., represented by the recognition state, the current posterior is fused to a an integrated posterior. The agent selects then the action  $a \in \mathcal{A}$  with largest  $Q(s, a)$ , i.e.,  $a_T = \arg \max_{a'} Q(s_T, a')$ .

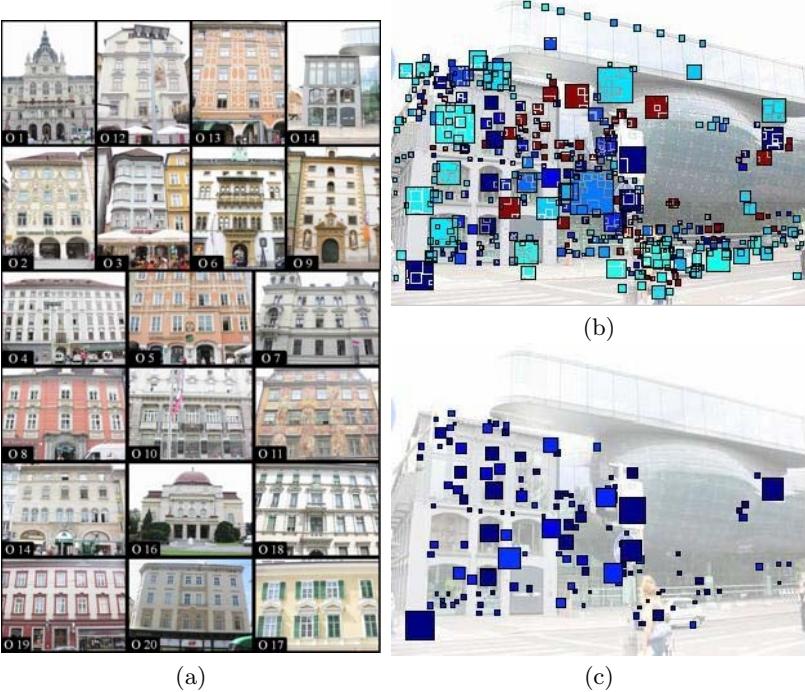
## 5 Experimental Results

The proposed methodology for cascaded sequential attention was applied to (i) an experiment with indoor imagery (i.e., the COIL-20 database), and to (ii) an experiment with outdoor imagery (i.e.m, the TSG-20 database) on the task of object recognition. The experimental results demonstrate that the informative descriptor method is robustly leading to similar saliency results under various environment conditions, and that the recursive integration of visual information from the informative foci of interest can find good matches to the stored perception-action object representation.

The indoor experiments were performed on 1440 images of the COIL-20 database (20 objects and 72 views by rotating each object by  $5^\circ$  around its vertical rotation axis), investigating up to 5 FOIs in each observation sequence, associating to  $k = 20$  codebook vectors from informative appearance patterns, in order to determine the recognition state, and deciding on the next saccade action to integrate the information from successive image locations. Fig. 4a represents the learning process, illustrating more rapid entropy decreases from the learned in contrast to random action selection policy. Fig. 4b visualizes the corresponding progress in requiring less actions to attain more informative recognition states. The recognition rate after the second action was 92% (learned) in contrast to 75% (random). A characteristic learned attention scanpath is depicted in Fig. 3b.



**Fig. 4.** Performance evaluation. (a) Rapid information gain from learned attention shift policy in contrast to random action selections. (b) The learned strategy requires shorter shift sequences to pass a given threshold on conditional entropy (threshold  $H_{goal} = 1.2$ )

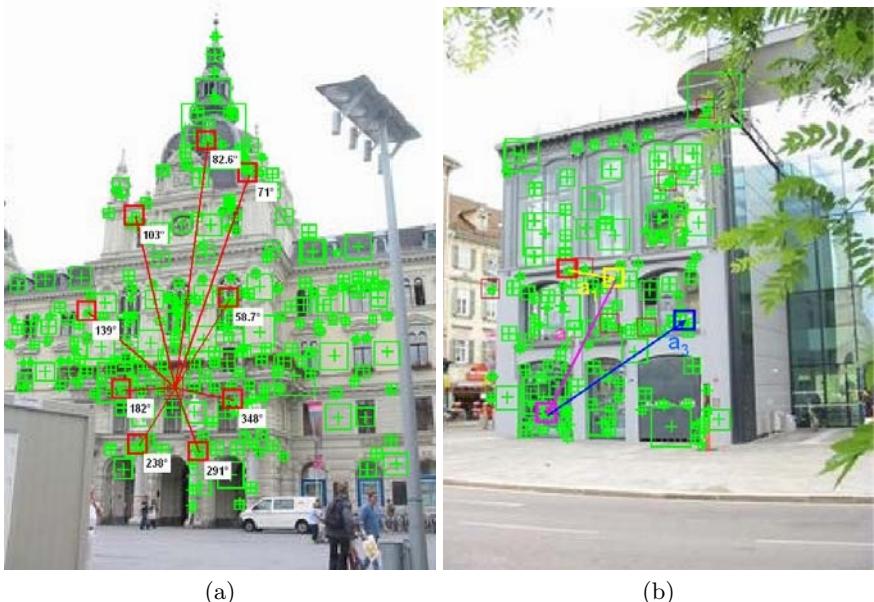


**Fig. 5.** (a) The TSG-20 database, consisting of images from 20 buildings in the city of Graz, displayed images were used for training (Sec. 5). (b,c) Informative descriptors for saliency

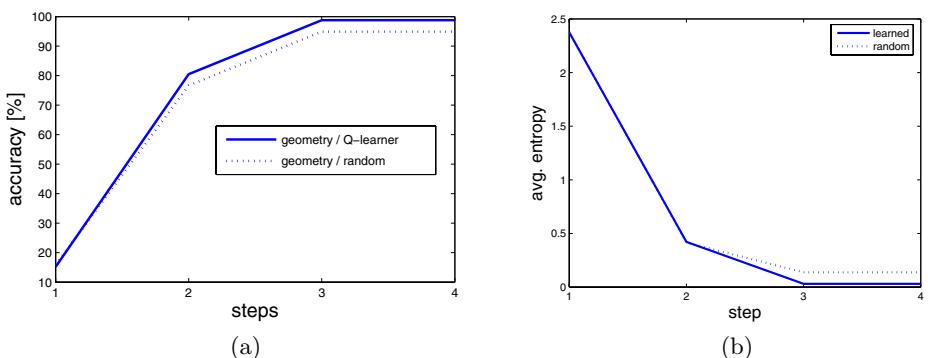
In the outdoor experiments, we decided to use a local descriptor, i.e., the SIFT descriptor ([7] Fig. 5) that can be robustly matched to the recordings in the database, despite viewpoint, illumination and scale changes in the object image captures. Fig. 5 depicts the principal stages in selecting the FOIs. (a) depicts the original training image. In (b), SIFT descriptor locations are overlaid with squares filled with color-codes of associated entropy values, from corresponding low (red) to high (blue) information values. (c) describes a corresponding posterior distribution over all object hypotheses from the MAP hypotheses in the informative SIFT descriptors (i-SIFTS). (d) depicts all selected i-SIFTS in the test image. Fig. 6 illustrates (a) descriptor selection by action and (b) a sample learned sequential attention sequence using the SIFT descriptor.

The experimental results were obtained from the images of the TSG-20 database<sup>1</sup> (20 objects and 2 views by approx. 30° viewpoint change), investigating up to 5 FOIs in each observation sequence, associating to  $k = 20$  codebook vectors to determine the recognition state, and deciding on the next saccade

<sup>1</sup> The TSG-20 (Tourist Sights Graz) database can be downloaded at the URL <http://dib.joanneum.at/cape/TSG-20>.



**Fig. 6.** (a) Saccadic actions originating in a FOI, directed towards 9 potential target FOIs, depicting angle values of corresponding shifts of attention starting in the center SIFT descriptor. (b) Learned descriptor-action based attention pattern (scanpath) to recognize an object



**Fig. 7.** Performance evaluation. (a) Accuracy improvement from learned attention shift policy in contrast to random action selections. (b) Information gain achieved by learned strategy with each additional perception-action cycle

action to integrate the information from successive image locations. Fig. 7a visualizes the progress gained from the learning process in requiring less actions to attain more informative recognition states. Fig. 7b reflects the corresponding learning process, illustrating more rapid entropy decreases from the learned in contrast to random action selection policy. The recognition rate after the second

action was  $\approx 98.8\%$  (learned) in contrast to  $\approx 96\%$  (random). A characteristic learned attention scanpath is depicted in Fig. 3b.

## 6 Conclusions and Future Work

The proposed methodology significantly extends previous work on sequential attention and decision making by providing a scalable framework for real world object recognition. The two-stage process of determining information theoretic saliency and integrating local descriptive information in a perception-action recognition dynamics is robust with respect to viewpoint, scale, and illumination changes, and provides rapid attentive matching by requiring only very few local samples to be integrated for object discrimination. Future work will be directed towards hierarchical reinforcement learning in order to provide local grouping schemes that will be integrated by means of a global saccadic information integration process.

## References

1. C. Bandera, F.J. Vico, J.M. Bravo, M.E. Harmon, and L.C. Baird III. Residual Q-learning applied to visual attention. In *International Conference on Machine Learning*, pages 20–27, 1996.
2. G. Deco. The computational neuroscience of visual cognition: Attention, memory and reward. In *Proc. International Workshop on Attention and Performance in Computational Vision*, pages 49–58, 2004.
3. G. Fritz, L. Paletta, and H. Bischof. Object recognition using local information content. In *Proc. International Conference on Pattern Recognition, ICPR 2004*, volume II, pages 15–18. Cambridge, UK, 2004.
4. G. Fritz, C. Seifert, L. Paletta, and H. Bischof. Rapid object recognition from discriminative regions of interest. In *Proc. National Conference on Artificial Intelligence, AAAI 2004*, pages 444–449. San Jose, CA, 2004.
5. J.M. Henderson. Human gaze control in real-world scene perception. *Trends in Cognitive Sciences*, 7:498 –504, 2003.
6. L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.
7. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
8. M.L. Puterman. *Markov Decision Processes*. John Wiley & Sons, New York, NY, 1994.
9. I.A. Rybak, I. Gusakova V., A.V. Golovan, L.N. Podladchikova, and N.A. Shevtsova. A model of attention-guided visual perception and recognition. *Vision Research*, 38:2387–2400, 1998.
10. S.P. Tipper, S. Grisson, and K. Kessler. Long-term inhibition of return of attention. *Psychological Science*, 14:19–25–105, 2003.
11. C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3,4):279–292, 1992.
12. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. European Conference on Computer Vision*, pages 18–32, 2000.

# Use of Quadrature Filters for Detection of Stellate Lesions in Mammograms

Hans Bornefalk

Dep. of Physics, KTH, SE-106 91 Stockholm, Sweden  
[bornefalk@particle.kth.se](mailto:bornefalk@particle.kth.se)

**Abstract.** We propose a method for finding stellate lesions in digitized mammograms based on the use of both local phase and local orientation information extracted from quadrature filter outputs. The local phase information allows efficient and fast separation between edges and lines and the local orientation estimates are used to find areas circumscribed by edges and with radiating lines. The method is incorporated in a computer-aided detection system and evaluation by FROC-curve analysis on a data set of 90 mammograms (45 pairs) yields a false positive rate of 0.3 per image at 90% sensitivity.

## 1 Introduction

Breast cancer is one of the leading causes of cancer deaths for women in the western world. Early detection with mammography is believed to reduce mortality [1], [2] and several countries, including Sweden, have nation-wide screening programs where middle-aged women are routinely invited to breast cancer screening. To reduce intra-reader variability and to increase overall efficacy several research groups have developed computer-aided methods that are intended to work as a “second look”, or spell checker, prompting the radiologist to take a second look at any suspicious areas. While there seems to be a need for more evaluation of the usefulness of computer-aided detection (CAD) systems in screening environments, the evidence is accumulating in favor of it increasing the overall efficacy of the screening process [3], [4]. With the advent of full-field digital mammography, which is expected to replace the nowadays commonplace film-screen mammography, the extra step of having to digitize the film mammogram will disappear and thus make computer-assisted methods even more attractive.

Depending on the type, carcinoma in the breast can manifest itself in several ways. Some are only revealed by the presence of microcalcifications, whereas other have a center mass which is visible as a denser region on the mammogram. This paper is concerned with the detection and classification of spiculated, or stellate, lesions. These are characterized by their branches which spread out in all different directions. Carcinoma of this subtype can either have or not have a well defined center body, and their detection, especially in dense breasts, is often hampered by superposition of normal anatomical structures of the breast. As their presence is a highly suspicious indicator of malignancy, especially combined

with a center mass, the detection of stellate patterns is a very important step in computer-aided detection of malignant lesions. It is therefore not surprising that so many different techniques for identifying these areas have been proposed.

Karssemeijer and te Brake [5] suggested a statistical method based on a map of pixel orientations. The idea is that if many pixels have orientations pointing toward a specific region, this region is more likely to be the center of a stellate distortion. The reported sensitivity was about 90% at a false positive rate of 1 per image. This method has later on been improved to 0.38 false positives per image at the same sensitivity [6]. Another popular method is based on first identifying individual spicules and then, via the Hough transform, accumulate evidence that they point in a certain direction [7], [8]. A third method for the detection of radiating spicules is based on histogram analysis of gradient angles [9]. The basic idea is that if the distribution of gradient angles in a certain local neighborhood has high standard deviation, this indicates that the gradients point in all different directions. This is indicative of a stellate pattern. On the other hand, if the distribution has a well defined peak, this indicates that the gradients in the neighborhood have a well defined orientation and thus that they do not belong to a stellate distortion. A common problem when detecting spiculated lesions is that they range in size from a few millimeters up to several centimeters. The solution adopted by Kegelmeyer [9] was simply to choose the size of the local neighborhood large enough to encompass even the largest lesions. However, a large local neighborhood makes detection of small spiculated lesions more difficult. Karssemeijer and te Brake addressed this issue by estimating the edge orientations at three different scales, and choosing the maximum response.

The objective of the study is to present an alternative algorithm for the detection of malignant tumors in digital mammograms. This method has been presented partially before by the author [10]. The method resembles all three methods mentioned above: pixel orientations are extracted [5], [6], the individual spicules are identified [7] and finally the angular distribution is analyzed in a way similar to Kegelmeyer's method [9]. To be able to evaluate the system, the whole CAD chain is implemented (finding regions-of-interest (ROIs), segmentation, extraction of image characteristics and classification) but the focus is on developing the ROI location technique using quadrature filter outputs.

## 2 Region of Interest Location

Several methods for finding seed points for ROIs exist. Most methods are intensity based using the fact that many tumors have a well defined central body. Other methods make use of spiculation features and look for regions where spicules seem to radiate from. We have chosen a combination of these two features, and also added a feature to capture the edge orientation. In order to minimize the risk of missing potential areas the entire image is uniformly sampled and each sampling point is evaluated in terms of contrast and spiculation. This evaluation is performed on three different scales. The contrast measure at node  $i, j$  is defined as the contrast between a circular neighborhood with radius  $r$

centered at  $i, j$  and the washer-shaped neighborhood with inner and outer radii given by  $r$  and  $2r$ . The spiculation and edge measures are based on orientation estimates extracted from quadrature filter outputs, as described below.

## 2.1 Introduction to Quadrature Filters

A filter  $f$  is a quadrature filter if there is a vector  $\hat{\mathbf{n}}$  such that the Fourier transform  $F$  of  $f$  is zero in one half of the Fourier domain:  $F(\omega) = 0$ , if  $\omega^T \hat{\mathbf{n}} \leq 0$ .  $\hat{\mathbf{n}}$  is called the directing vector of the filter. Quadrature filters and a method to construct orientation tensors from the quadrature filter outputs are described thoroughly in Granlund and Knutsson [11] and here we will only touch briefly on some essential concepts.

The directing vector of quadrature filter  $i$  is denoted  $\hat{\mathbf{n}}_i$  with  $\varphi_i = \arg(\hat{\mathbf{n}}_i)$ . The quadrature filters will be complex in the spatial domain since  $F(\omega) \neq F^*(-\omega)$ , and thus the convolution output with the image signal  $s$ ,  $\mathbf{q}_i = f_i * s$ , will be complex. We let  $q_i$  denote the magnitude of  $\mathbf{q}_i$  and similar for the phase angle:  $\theta_i = \arg(\mathbf{q}_i)$ .

The local orientation in an image is the direction in which the signal exhibits maximal variation. This direction can only be determined modulo  $\pi$  which is the rationale for introducing the double angle representation below. If  $\mathbf{v}$  is a vector oriented along the axis of maximal signal variation in a certain point, the double angle representation of the local orientation estimate is given by  $\mathbf{z} = ce^{i2\varphi}$  where the magnitude  $c$  is the certainty of the estimate and  $\varphi = \arg(\mathbf{v})$ .

It can be shown [11] that the double angle representation of the local orientation can be constructed as:  $\mathbf{z} = \sum_i q_i \hat{\mathbf{m}}_i$  where  $\hat{\mathbf{m}}_i = (\cos(2\varphi_i), \sin(2\varphi_i))$ . With four quadrature filters and  $\varphi_i = (i-1)\pi/4$  this leads to a convenient expression for the 2D-orientation vector:

$$\mathbf{z} = (q_1 - q_3, q_2 - q_4). \quad (1)$$

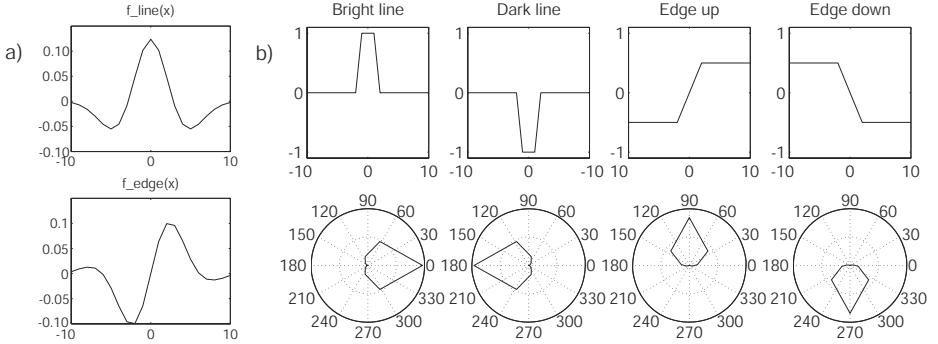
In the spatial domain, a quadrature filter can be written as the sum of a real line detector and a real edge detector:

$$f(x) = f_{line}(x) - i f_{edge}(x). \quad (2)$$

The line and edge detectors are related to each other the following way in the Fourier domain:

$$H_{edge}(u) = \begin{cases} -iH_{line}(u) & \text{if } u < 0 \\ iH_{line}(u) & \text{if } u \geq 0 \end{cases} \quad (3)$$

$f_{line}$  is an even function and  $f_{edge}$  is an odd function and this can be used to distinguish between lines and edges. The phase angle introduced above actually reflects the relationship between the evenness and oddness of the signal. A line would give a non-zero response only for the real part of  $f$  (since  $f_{line}$  is even) and hence the phase angle  $\theta = \arg(f * s)$  would be close to zero. On the other hand, an edge would give a phase angle of  $\pm\pi/2$  since only the odd part of Eq. 2 would give a contribution. This is illustrated in the right panel of Fig. 1 for the



**Fig. 1.** a) Line and edge filters used for construction of a 1D-quadrature filter. b) Lower row depicts filter responses for the signals in the top row

filter pair in the left panel. Note how the phase angle can be used to separate edges from lines.

The extension of the phase concept to two dimensions is not trivial, but will give us the necessary means to distinguish different events from each other (namely edges, bright lines and dark lines). The reason for the difficulties is that phase cannot be defined independently of direction, and as the directing vectors of the quadrature filters point in different directions (yielding opposite signs for similar events) care must be taken in the summation [12]. If  $\Re(\mathbf{q}_i)$  and  $\Im(\mathbf{q}_i)$  denote the real and imaginary parts of the filter output from the quadrature filter in direction  $\hat{\mathbf{n}}_i$ , the weighted filter output  $\mathbf{q}$  is given by

$$\Re(\mathbf{q}) = \sum_{i=1}^4 \Re(\mathbf{q}_i), \quad \Im(\mathbf{q}) = \sum_{i=1}^4 I_i \Im(\mathbf{q}_i) \quad (4)$$

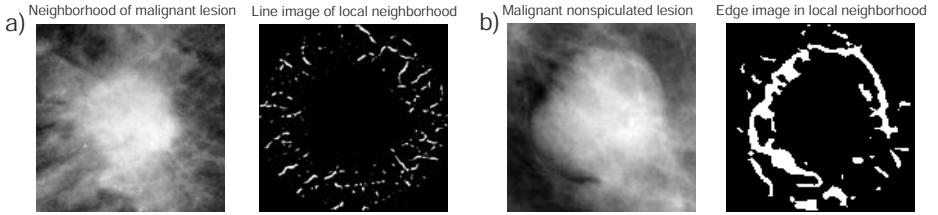
where

$$I_i = \begin{cases} 1 & \text{if } |\varphi_i - \varphi| \leq \frac{\pi}{2} \\ -1 & \text{if } |\varphi_i - \varphi| > \frac{\pi}{2} \end{cases} \quad (5)$$

The interpretation of the indicator function  $I_i$  is that when the local orientation in the image ( $\varphi$ ) and the directing vector of the filter ( $\varphi_i$ ) differ by more than  $\pi/2$  the filter output  $\mathbf{q}_i$  must be conjugated to account for the anti-symmetric imaginary part. The total phase  $\theta$  is now given as  $\theta = \arg(\mathbf{q}) = \arg(\Re(\mathbf{q}) + i\Im(\mathbf{q}))$ . As for the one-dimensional case, phase angles close to zero correspond to bright lines, phase angles close to  $\pm\pi$  correspond to dark lines and phase angles close to  $\pm\pi/2$  correspond to edges.

## 2.2 Illustration on a Real Mammographic Lesion

By thresholding the filter outputs on certainty and phase, a binary image is produced. This can be used to separate bright lines, and thus candidates for spicules, from the surrounding tissue. Edges can also be extracted if the thresholds are chosen appropriately. This is illustrated in Fig. 2 where phase angles



**Fig. 2.** a) Malignant lesion with radiating spicules in a washer-shaped local neighborhood. Detected by thresholding on local phase. b) Malignant lesion and prominent edges as detected by local phase

$\theta \in (-\pi/6, \pi/6)$  are extracted in panel a) and  $|\theta| \in (5\pi/6, 7\pi/6)$  in panel b). Note how the majority of lines in Fig. 2 seem to radiate away from the center of the lesion, and the edges seem to circumscribe the center. The question is how this can be quantified for subsequent use in classification?

### 2.3 Measure of Spiculatedness in a Local Neighborhood

Consider a coordinate  $\mathbf{x}_0 = (x_0, y_0)$  in the image. Next consider a pixel coordinate  $\mathbf{x} = (x, y)$  on a detected bright line. This is illustrated in panel a) in Fig. 3. The direction of maximal signal variation in this pixel is  $\mathbf{v}(\mathbf{x})$ . Let  $\varphi(\mathbf{x}) = \arg(\mathbf{v}(\mathbf{x}))$ . From Eq. (1) we get the following expression for the double angle representation of local orientation:

$$\mathbf{z}(\mathbf{x}) = ce^{i2\varphi(\mathbf{x})} = q_1 - q_3 + i(q_2 - q_4). \quad (6)$$

Let  $\hat{\mathbf{r}}$  be the normalized vector pointing from  $\mathbf{x}_0$  to the pixel  $\mathbf{x}$ :  $\hat{\mathbf{r}}(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{x}_0}{|\mathbf{x} - \mathbf{x}_0|}$ . Since the vector  $\hat{\mathbf{r}}$  is normalized, it can be expressed as  $(\cos \varphi_r, \sin \varphi_r)$ . Now define

$$\hat{\mathbf{r}}_{double}(\mathbf{x}) = (\cos(2\varphi_r), \sin(2\varphi_r)). \quad (7)$$

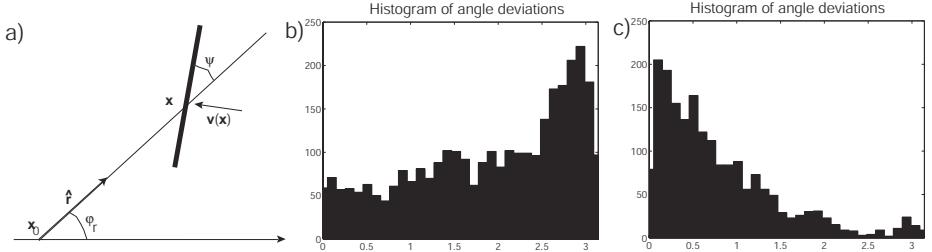
If  $\mathbf{x}$  is located on a line radiating away from the center coordinate (in the same direction as  $\hat{\mathbf{r}}(\mathbf{x})$ ), the angles between  $\hat{\mathbf{r}}_{double}$  and  $\mathbf{z}(\mathbf{x})$  will be  $\pi$ . On the other hand, if  $\mathbf{x}$  is located on a line perpendicular to  $\hat{\mathbf{r}}$ , the angle between  $\mathbf{z}$  and  $\hat{\mathbf{r}}_{double}$  will be zero. To see that, consider panel a) in Fig. 3 where  $\psi$  denotes the angle between  $\hat{\mathbf{r}}(\mathbf{x})$  and  $\mathbf{v}(\mathbf{x})$ . From the figure we see that  $\arg(\mathbf{v}) = \varphi_r + \psi \pm \pi/2$ . This means that

$$\arg(\mathbf{z}) = 2\varphi_r + 2\psi \pm \pi = 2\varphi_r + 2\psi + \pi \pmod{2\pi}. \quad (8)$$

Since  $\arg(\hat{\mathbf{r}}_{double}) = 2\varphi_r$  the absolute value of the difference between the angles (modulo  $2\pi$ ) is

$$\phi = |\arg(\mathbf{z}) - \arg(\hat{\mathbf{r}}_{double}) \pmod{2\pi}| = |2\psi + \pi \pmod{2\pi}|. \quad (9)$$

Now, with  $\psi$  close to zero, as it would be if the line is part of a stellate pattern, the angle difference will be close to  $\pi$ , as proposed above. On the other hand, if the



**Fig. 3.** a) Illustration of relevant angles. b) Angle difference distribution  $\phi$  for the thresholded bright lines in a local neighborhood of a spiculated lesion. Note that the distribution is clearly skewed to the right. c) Angle distribution for edge pixels. The leftward skew indicates that the edges are orthogonal to the radial direction

line is perpendicular to  $\hat{r}$  the angle difference  $\phi$  will be close to zero. Thus, if the distribution of the angle differences corresponding to the pixels identified in the line image in a local neighborhood is skewed towards  $\pi$ , this is an indication that many lines are radiating from the center. Conversely, if angle differences of the edge image are skewed towards the left, this is an indication that the prominent edges are perpendicular to lines radiating from the center. An illustration, using the actual data from Fig. 2, is shown in panels b) and c) of Fig. 3.

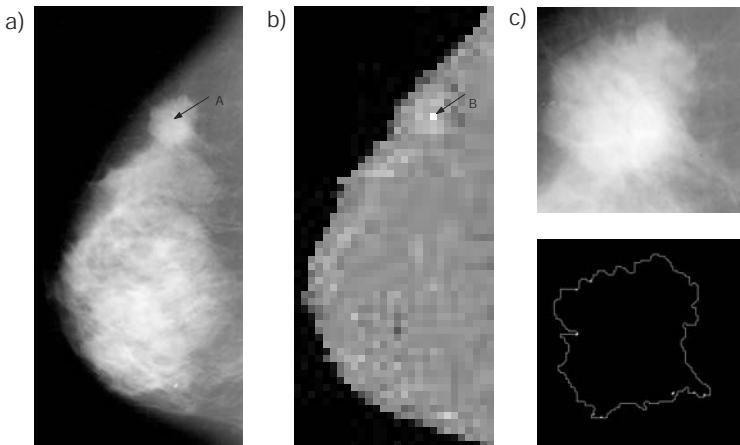
## 2.4 Training an ROI Extractor

Five features (or local image characteristics) are used in the ROI extractor. The first is the contrast as mentioned above. The second and the third are the fraction of points in the line image in the washer-shaped neighborhood that have angle deviations  $\phi$  in the intervals  $(\pi/2, \pi]$  and  $(3\pi/4, \pi]$ , respectively. The fourth and fifth features are similar measures for the edge image.

A freely available support vector machine (SVM) implementation [13] is trained to distinguish between areas that could potentially be malignant and those that could not. A washer-shaped neighborhood with inner radius matching the size of the radiologist supplied ground truth is used in extracting the features. Once the five features in the malignant area have been extracted, the same washer-shaped mask is used to extract the same five features in 100 randomly chosen neighborhoods in the image. These are the normal samples that will be used when training the SVM. To ensure independency between training and evaluation data, the leave-one-case-out technique is employed when the SVM is used to extract ROIs.

When features are extracted from a mammogram in the ROI extraction step, the size of the possible lesion is unknown and three different radii of the washer-shaped neighborhood are evaluated. The radius where the corresponding features give the highest SVM response is taken as the size of the ROI - much the same way as Karssemeijer and te Brake do [5], [6].

A typical intermediate result of the ROI extractor is shown in Fig. 4. In panel a) the output from the support vector machine is shown. Note that this



**Fig. 4.** a) Original mammogram with a stellate lesion in the upper part of the image (marked A). b) SVM output from ROI extraction step. The pixel with the highest intensity (marked B) is taken as the first seed point for the level set boundary refinement algorithm, the result of which is shown in panel c)

is not the final classifying decision of the CAD system, but rather the first step localizing the ROIs that should be further processed. The coordinates with the highest response, as shown in panel b) are then extracted and passed on to the segmentation step.

### 3 Segmentation and Feature Extraction

From the image in panel b) of Fig. 4 the coordinates of the intensity maxima are extracted and a boundary refinement algorithm is initiated around this neighborhood. The local neighborhood and the result of the segmentation is shown in panel c). In this paper a level set method is used. Once an ROI has been segmented from the background, its immediate background is determined as all pixels within a distance  $d$  from the ROI where  $d$  is chosen such that the area of the background roughly corresponds to the area of the ROI. Then the extended ROI (the ROI and its immediate background) is removed from the ROI extractor grid output. This process is repeated until we have 10 regions of interest to pass on to the next steps in the algorithm: feature extraction and classification. Using the segmentation results, three of the five features are recalculated using the segmented ROI and its immediate surrounding instead of the washer-shaped neighborhoods used in the ROI extraction step. These are the contrast and the angle orientations in the interval  $(\pi/2, \pi]$  and similar for the edge orientation. Four additional features are added to aid in the classification. The standard deviation of the interior of the ROI normalized with the square root of the intensity is a texture measure capturing the homogeneity of the area. An equivalent feature is extracted

for the immediate background as is the raw standard deviation of the of the surrounding pixel values. These three very simple texture features are complemented with the order at which the ROI was extracted in the ROI location step (1-10).

The same SVM implementation as mentioned above is trained with the features from these refined areas. The result is presented as a free-response receiver operating characteristic (FROC) curve.

## 4 Data Set, Scoring Protocol, and Evaluation Method

The data set consists of 90 mammograms from 45 cases of proven malignant cancers (one image from each of the CC and MLO view). All mammograms contain exactly one malignant tumor classified as a spiculated lesion by expert radiologists. The malignant areas have been manually annotated by radiologists and all images are taken from the Digital Database for Screening Mammography<sup>1</sup> (DDSM) at the University of South Florida [14]. The size and subtleties of the tumors are depicted in Figure 5. The tail towards large diameters is to some extent explained by some radiologists not marking the core outline of the tumor, but rather all tissue affected by metastasis. When available, the smaller core outline of the tumor has been used as ground truth. The skewed distribution of subtleties is somewhat concerning at first since a subtlety rating of 1 indicates that a tumor is very difficult to spot and a rating of 5 that it is obvious. However, this seems to be an effect of the rating protocol used by the radiologists at the institution where they were acquired and other researchers concluded that no clear bias towards easier cases is present in the DDSM (when compared to other data sets) [15].

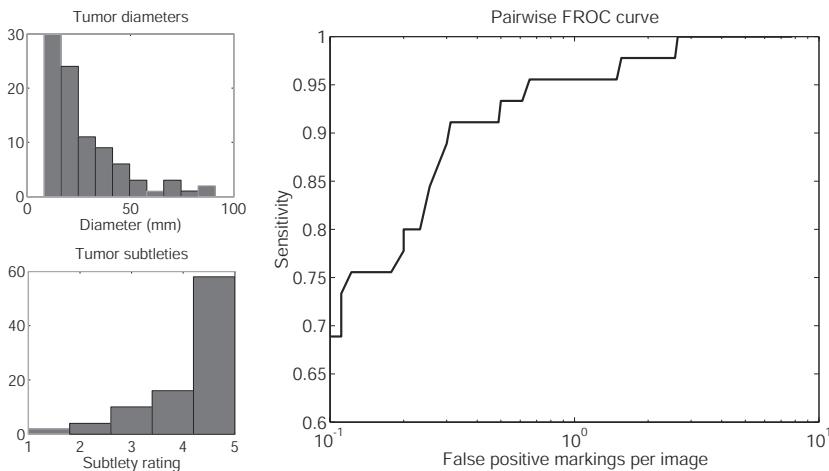
A CAD-marking is considered a true positive if at least 1/3 of the automatically segmented area lies within the area annotated by radiologists. If there is an overlap, but less than a third, the marking is considered a false positive as are all markings not overlapping the annotated area.

The FROC curve quantifies the inherent trade-off between high sensitivity (true positive fraction) and false positive markings per image and its use in evaluation of mammography CAD schemes is well established. The FROC curve is a parameter curve traced out by successively lowering the threshold level of the classifying machine above which to consider a region malignant. For high thresholds, there will be few false positive markings but at the cost of missed cancers. Lowering the threshold level will increase the sensitivity but at the cost of more false positive markings.

The training and evaluation is performed with the leave-one-out method: for each of the 45 runs training is performed on 44 cases and evaluation on the 45th. The evaluation is performed pair-wise,. a cancer is considered found if it is correctly marked in one of the two projections.

---

<sup>1</sup> <http://marathon.csee.usf.edu/Mammography/Database.html>



**Fig. 5.** Diameter and subtleties of tumors in the data set. Trade-off between false positive markings and sensitivity. The evaluation is done pairwise. At 90% sensitivity there are 0.32 false positive markings per image

## 5 Results

The resulting FROC curve is shown in Figure 5 and a brief comment on the specificity might be in place. On average, each image will have 0.3 false positive markings at a chosen sensitivity of 90%, thus making the method useless for diagnosis. The purpose of CAD (at least to date) has however not been diagnosis but detection, as mentioned in the introduction. The goal has rather been to draw the radiologists' attention to subtle areas that might otherwise have been overlooked, and then pass on the main responsibility for the classification to the human experts. In this context, 0.3 false positive markings per image is acceptable and is comparable to state-of-the-art algorithms. It must be noted however, that reported results are very dependent on the scoring protocol and data sets used [16], [17] and that the data set size used here (45 pairs of images) is too small to draw any strong conclusions about the relative performance of different algorithms. However, it has been demonstrated that spiculation and edge measures extracted using local phase and local orientation information from quadrature filter outputs could work well in a CAD algorithm designed to find spiculated lesions.

## References

1. Tabar, L.: Control of breast cancer through screening mammography. *Radiology* 174 (1990) 655–656
2. Tabar, L., Yen, M-F., Vitak, B., Chen, H-H.T., Smith, R.A., Duffy, S.W.: Mammography service screening and mortality in breast cancer patients: 20-year follow-up before and after introduction of screening. *Lancet* 361 (2003) 1405–1410

3. Freer, T.W., Ulissey, M.J.: Screening mammography with computer-aided detection: prospective study of 12,860 patients in a community breast center. *Radiology* 220 (2001) 781–786
4. Destounis, S.V. *et al.*: Can computer-aided detection with double reading of screening mammograms help decrease the false-negative rate? Initial experience, *Radiology* 232 (2004) 578–584
5. Karssemeijer, N., te Brake, G.M.: Detection of Stellate Distortions in Mammograms. *IEEE Trans. Med. Im.* 15(5) (1996) 611–619
6. te Brake, G.M., Karssemeijer, N.: Segmentation of suspicious densities in digital mammograms. *Med. Phys.* 28(2) (2001) 259–266
7. Kobatake, H., Yoshinaga, Y.: Detection of Spicules on Mammogram Based on Skeleton Analysis. *IEEE Trans. Med. Im.* 15(3) (1996) 235–245
8. Ng, S.L., Bischof, W.F.: Automated detection and classification of breast tumors. *Comput. Biomed. Res.* 25 (1992) 218–237
9. Kegelmeyer Jr, W.P.: Computer Detection of Stellate Lesions in Mammograms. In: Proc. SPIE 1660 (1992) 446–454
10. Bornefalk, H.: Use of phase and certainty information from quadrature filters in detection of stellate patterns in digitized mammograms. In: Proc. SPIE 5370 (2004) 97–107
11. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Dordrecht (1995)
12. Haglund, L.: Adaptive Multidimensional Filtering. PhD thesis, Linköping University, Sweden. Dissertation No. 284 (1992)
13. Cawley, G.C.: MATLAB Support Vector Machine Toolbox (v0.50 $\beta$ ). University of East Anglia, School of Information Systems, Norwich, 2000. <http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox>
14. Heath, M., Bowyer, K.W., Kopans, D.: Current status of the Digital Database for Screening Mammography. In: Digital Mammography. Kluwer Academic Publishers, Dordrecht (1998)
15. Petrick, N., Sahiner, B., Chan, H., Helvie, M.A., Paquerault, S., Hadjiiski, L.M.: Breast cancer detection: Evaluation of a mass-detection algorithm for computer-aided diagnosis—Experience in 263 patients. *Radiology* 224 (2002) 217–224
16. Nishikawa, R.M., Giger, M.L., Doi, K., Metz, C.E., Yin, F., Vyborny, C.J., Schmidt, R.A.: Effect of case selection on the performance of computer-aided detection schemes *Med. Phys.* 21(2) (1994) 265–269
17. Giger, M.L.: Current issues in CAD for mammography. In: Digital Mammography '96. Elsevier Science, Philadelphia (1996) 53–59

# A Study of the Yosemite Sequence Used as a Test Sequence for Estimation of Optical Flow

Ivar Austvoll

Signal and Image Processing Group,  
Department of Electrical Engineering and Computer Science,  
University of Stavanger, N-4036 Stavanger, Norway  
`Ivar.Austvoll@uis.no`

**Abstract.** Since the publication of the comparative study done by Barron et al. on optical flow estimation, a race was started to achieve more and more accurate and dense velocity fields. For comparison a few synthetic image sequences has been used. The most complex of these is the Yosemite Flying sequence that contains both a diverging field, occlusion and multiple motions at the horizon. About 10 years ago it was suggested to remove the sky region because the correct flow used in earlier work was not found to be the real ground truth for this region. In this paper we present a study of the sky region in this test sequence, and discuss its usefulness for evaluation of optical flow estimation.

## 1 Introduction

It took about 30 years from the introduction of the conception of **optical flow** by Gibson [1] till the breaking through of algorithms for estimation of optical flow [2, 3]<sup>1</sup>. Since then we have had a steady growth in the number of new approaches for optical flow estimation. An important event was the comparative study done by Barron et al. [5, 6], that started a competition towards better accuracy for a few synthetic image sequences. Here it is preposterous to mention all the participants in this struggle towards better results. We have chosen some of the best results reported and visualized the latest developments in this area graphically. Names will be missing, but we hope to present a representative choice of techniques and their results.

The focus in this paper is to demonstrate some of the peculiarities of the sky region for the Yosemite sequence and look at the consequences this has for the use as a test sequence.

In the next section we give some information on the Yosemite sequence. Then we discuss the error measure used for comparison of the different techniques, and

<sup>1</sup> We have here not mentioned the efforts in motion estimation connected to image sequence processing. Some of the earliest results in this field are from Limb and Murphy [4].

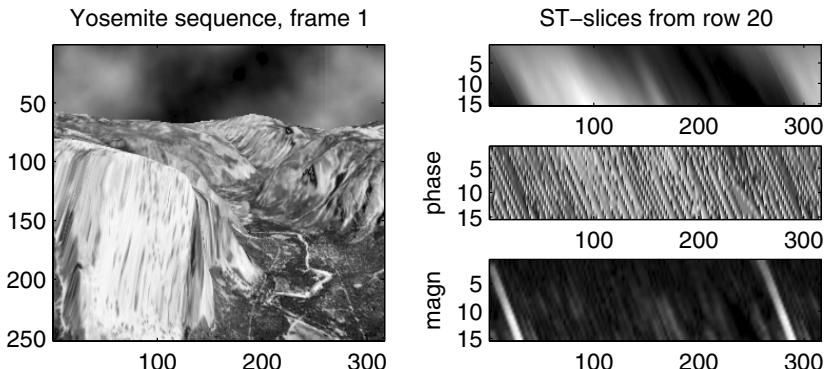
we present some of the results reported since 1980. We continue with some simple experiments on the sky region, and finally we give a discussion of the results and conclude our work.

## 2 The Yosemite Sequence

The Yosemite sequence was created by Lynn Quam at SRI [5]. This sequence is based on an aerial image of the Yosemite valley and synthetic generated by flying through the valley. In the sky region clouds are generated based on fractal textures. The ground truth for the motion field is known for the valley part, but it is questionable if the ground truth used for the sky region is correct [7]. According to Black in [7] .. . . . .

According to our view this statement is a bit too strong and only partly true.

The first image frame in the Yosemite sequence is shown in Figure 1, left. To analyze the situation in the sky region we will use a simple approach, where space-time-slices, ST-slices [8], are extracted for the rows (The motion is purely horizontal in the sky region.). Examples are found in Figure 1, right. The uppermost ST-slice is from the original signal, the middle is the phase output from a directional filter and the nethermost is the magnitude output from the directional filter. It is well known, and also illustrated in these ST-slices, that the component velocity in the direction of the ST-slice is given by the orientation of the ST-signal. To estimate the component velocity we can use a method for orientation estimation. This problem is discussed in [9], where a new set of quadrature filters, .. . . . is introduced. As stated in [9] the estimation of orientation and estimation of velocity is identical when the signal

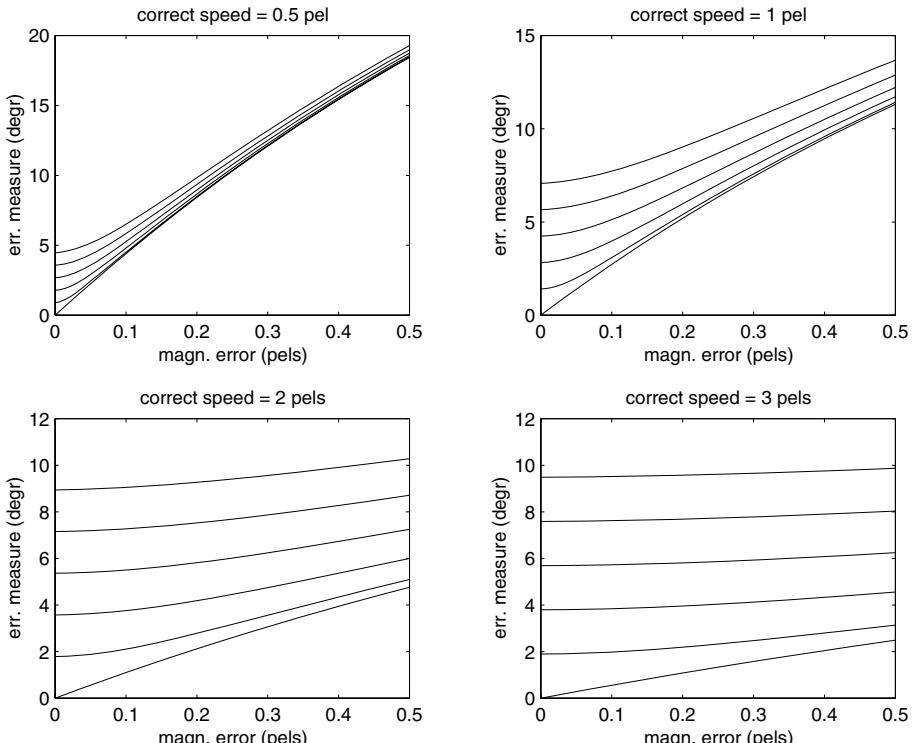


**Fig. 1.** Left: first image of the Yosemite sequence. Right: space-time-slices, original signal, phase and magnitude from directional filter

is bandlimited. In our work we have used a quadrature filter bank based on the discrete Prolate Spheroidal sequence (DPSS) (equivalent to the  $\text{Hann}$  window) [10, 11]. To bandlimit the signal we used the scale space filter suggested in [12].

## 2.1 Error Measure

In most published work on optical flow estimation after Barron et al. and their comparative study, the **angular error measure** introduced by Fleet and Jepson [13] has been used. The argument for introducing this error measure was that the length of the vector error ( $\mathbf{v} - \mathbf{v}^{\text{corr}}$ ), where  $\mathbf{v}$  is the measured velocity and  $\mathbf{v}^{\text{corr}}$  the ground truth, is not useful as a measure of error because it depends on the speed (speed = magnitude of the velocity vector). To compute this measure we use the 3D space-time velocity vectors,  $\mathbf{v} = [\mathbf{v} \ 1]^T$  and  $\mathbf{v}^{\text{corr}} = [\mathbf{v}^{\text{corr}} \ 1]^T$  with lengths,  $\|\mathbf{v}\|$  and  $\|\mathbf{v}^{\text{corr}}\|$  respectively. The **angular error measure** is then given by:



**Fig. 2.** Angular error measure as a function of magnitude error,  $v_{\text{err}}$ , for absolute angular errors,  $\alpha_e$ , of 0, 2, 4, 8, 10 degrees, at different correct speeds, computed from equation (1)

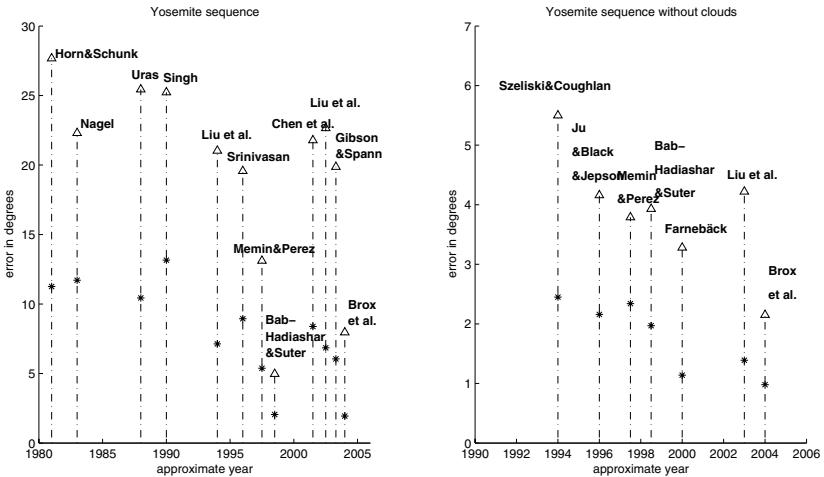
$$\varphi_e = \arccos\left(\frac{\mathbf{v}^T \mathbf{v}^{corr}}{\|\mathbf{v}\| \|\mathbf{v}^{corr}\|}\right). \quad (1)$$

For the estimated velocity vectors there can be an error in direction, **angular error**  $\alpha_e = \text{ang}(\mathbf{v}) - \text{ang}(\mathbf{v}^{corr})$ , and in the length of the vector, **magnitude error**  $v_{err} = |\mathbf{v}| - |\mathbf{v}^{corr}|$ . Some of the properties of this error measure can be seen from Figure 2.

## 2.2 Error Statistics

Succeeding the work of Barron et al. [5] a bunch of competing methods for estimation of optical flow has been published. We will here only focus on the average error and the standard deviation found for the Yosemite sequence. Two versions of the results exist, with clouds and without clouds. From about 1994 some researchers followed the attitude of Black [7] and skipped the clouds.

For the Yosemite sequence including clouds, error measures is shown in Figure 3, left. Here we have only included published results where the density is 100 % and have chosen the best in each time slot. The results without clouds is



**Fig. 3.** History of error measures for the Yosemite sequence. The asterisks is placed at the average error while the arrows indicates the standard deviation

shown in Figure 3, right. Here only results from about 1994 are given. We have excluded the newly published results in [14] because the numbers seem not to be reasonable <sup>2</sup>. The best results sofar is from Brox et al. [15] with less than

<sup>2</sup> The velocity arrows in Figure 1c) in the article does not correspond to the average error of 0.2 degrees that is given in the Table I for comparison with other methods! The published accuracy is almost an order of magnitude better than other methods! The reason is probably that a different error measure has been used.

2 degrees average error with clouds, and less than 1 without clouds. Very close is the work of Bab-Hadiashar and Suter [16]. For the Yosemite sequence with clouds these results stands out, especially with much better standard deviation. For the sequence without clouds the results are more even. In addition to the former, Liu et al. [17], Farnebäck [18] among others, have reported nice results. The best results seem to come from differential methods including multiscale approaches and both global and local smoothing [15] and methods based on robust statistics [16, 19, 20]. Orientation tensors combined with affine models [18, 17] is another approach with good results. Reasonable results is also achieved with a wavelet based method [21].

From the results of the average angular error the relative contribution of angular and magnitude error it is not known. This can be of importance for practical applications. From Figure 2 we can deduce some conclusions. If the correct velocity is 2 pels (picture elements, pixels) the average magnitude error is less than about 0.2 pels in the worst case (no angular error) and the average angular error is less than 2 degrees when the average error is less than 2 degrees. This is the case in the sky region.

### 3 Experiments

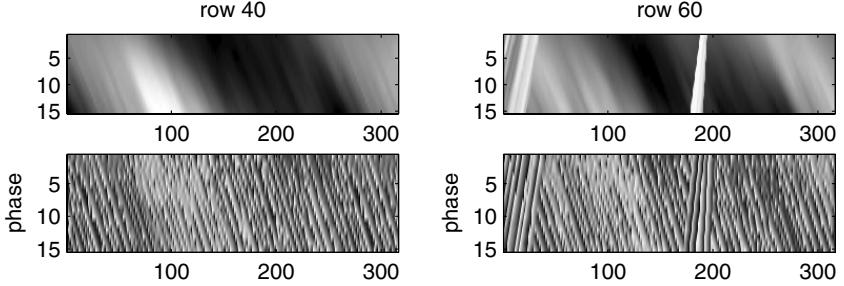
The aim of this work has not been to give prominence to a special technique for estimation of optical flow. We only want to demonstrate the relative ability to estimate motion in the sky region. For our experiments we have therefore used a method at our hand, available from earlier work, with some smaller modifications. The accuracy of the results is therefore also moderate.

#### 3.1 Velocity Estimation Method

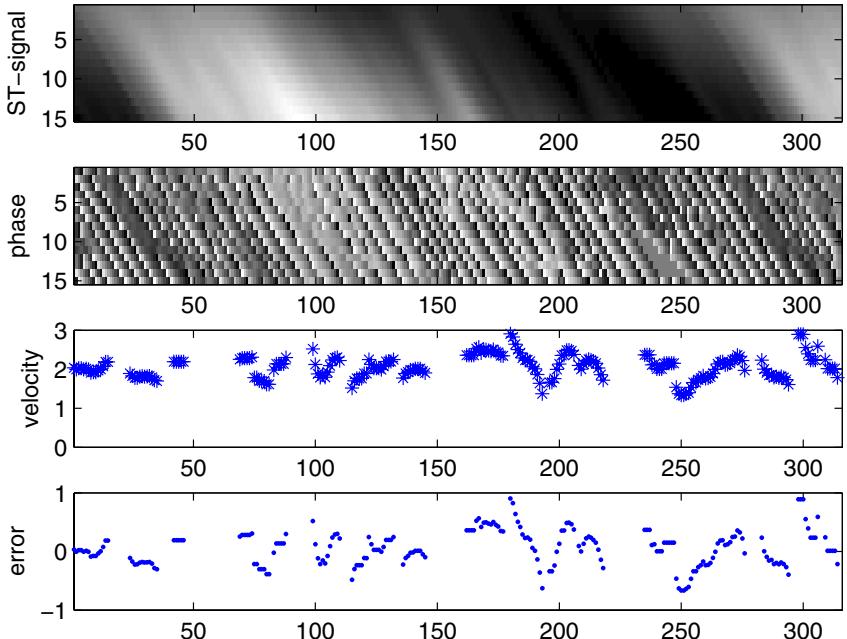
The ST-slices are smoothed by a scale-space filter and each row is filtered by a 1D one-sided complex bandpass filter. The method is a simplified version of the one described in [10]. The output is a complex ST-slice. The component velocity in the direction of the spatial axis for the ST-slice (in our case horizontal) can be computed from the direction of the pattern in the ST-slice, either from the original signal, the smoothed ST-slice or the phase from the filtered ST-slice. This can be done by a structure tensor [22, 8], by a set of quadrature filters [22, 10] or any suitable method for estimation of directional patterns. We have chosen the approach in [10], used only one scale and computed the component velocity in the horizontal direction.

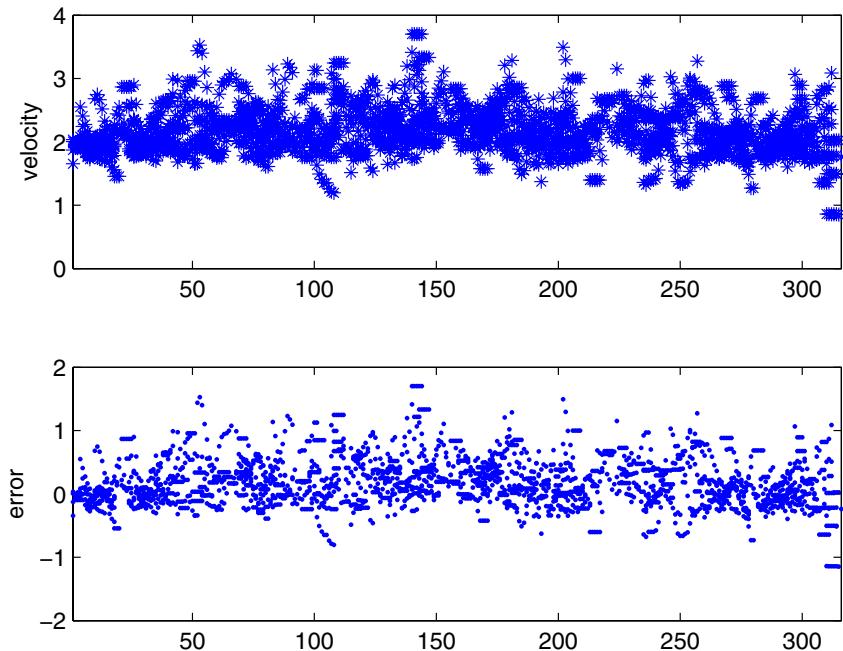
#### 3.2 Results

To demonstrate the problems introduced in the sky region we have computed the horizontal component of the velocity (the vertical component is zero in this region) for rows. From visual inspection of the ST-slices, see Figure 1, right, it is obvious that the velocity is about 2 pels from left to right. From the original signal we can also confirm that the brightness constancy is violated, but the

**Fig. 4.** ST-slices

space-time tracks have the expected direction. The signal from the directional filter is more conclusive. The phase has a very distinct direction in space-time, visually, when we use a global view. For local regions the case is not so obvious, with many dislocations from the main direction. The ST-slices for two more rows is found in Figure 4. The left figures show ST-slices at row number 40. The motion in a direction to the right is obvious. The right part of this figure show ST-slices from row number 60. This is just at the horizon.

**Fig. 5.** ST-signals, original and phase for row number 20, upper figures. Estimated magnitude velocity for row number 20 at frame number 7 and corresponding errors, lower plots



**Fig. 6.** Estimated magnitude velocity and corresponding errors for row number 5 to 55 for steps in 5, i.e. 11 rows, at frame number 7

The horizon is broken at two places, approximately columns 0 - 30 and 170 - 200, where the motion is to the left under the horizon. Accurate estimation of these velocities are not trivial. The result from estimation of the velocity for row number 20 is shown in Figure 5. The average error for this line is 0.25 pels when the correct velocity is 2.0 pels. The standard deviation is 0.19. The density is 67%. This corresponds to an average angular error less than 4 degrees when the absolute angular error is less than 4 degrees (see Figure 2). If the absolute angular error is approximately zero the average angular error is 2 degrees. The result for 11 rows covering the complete sky region is shown in Figure 6. It can be seen from this figure that there is a bias towards higher values for the magnitude velocity. The total average angular error is 0.3 with density 63%. This also corresponds to less than 4 degrees average angular error for small absolute angular errors. The accuracy is in the same range as the best reported results except for Brox et al. and Bab-Hadiashar and Suter (see Figure 3, left). This means that we have been able to detect and measure the velocity in the sky region despite the fact that the simulation of the clouds have been based on fractal patterns. The real problems come when we want to estimate the velocity for positions at the horizon. Here we have multiple motions as demonstrated in Figure 4, right. This is a much tougher problem than what

the clouds represents. We will not give any results on this, but just confirm that the challenge for the motion estimation algorithm is in this region and not in the sky.

## 4 Discussion

When we want to evaluate a method we should use realistic data. From our opinion the translating/diverging tree sequence and the Yosemite sequence without clouds are not sufficient to make a proper evaluation of a technique for estimation of optical flow. By excluding the sky region the occlusion boundary represented by the horizon is simplified. There will be a step between the velocities below the horizon and the zero velocity in the sky region, but this is much easier than the opposing velocities at the horizon when clouds move to the right.

We suggest to replace the Yosemite sequence with a new test sequence that includes occlusion boundaries and multiple motions. The Yosemite sequence could be modified by replacing the clouds with some flying objects (plane(s), balloon(s),etc.) with known velocity.

It is insufficient to publish the average angular error. We would like to know the distribution between angular and magnitude errors. In addition histograms of the errors, both magnitude and angular, will give a more complete picture of the performance for the actual algorithms.

To make a differentiation between different techniques it is also of interest to know what kind of structures or motion patterns that cause the largest errors.

## 5 Conclusions and Future Work

As we have argued in this paper, the problem is not the clouds, but it is mainly the occluding boundary with multiple motions. This challenge is reduced appreciable by removing the clouds such that the motion vectors in the sky region is zero. Our suggestion is to introduce flying objects in the sky instead. This will give a more realistic situation for test of the optical flow methods.

More work on error analysis is needed. This could be done in a manner as suggested in [23]. The performance at occlusion boundaries and more challenging motion patterns should be described in more detail for each algorithm.

In addition new and more challenging test sequences should be developed.

## 6 Call for New Test Sequences!

We will continue our work on estimation of optical flow and try to develop some new test sequences. We also calls for (thanks to the reviewer for this idea of a call for ...) suggestions from other workers in the field of motion estimation to suggest better test sequences and evaluation methods.

## References

1. Gibson, J.J.: *The Perception of the Visual World*. Houghton-Mifflin: Boston (1950)
2. Horn, B., Schunck, B.: Determining optical flow. *Artificial Intelligence* **17** (1981) 185–203
3. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. DARPA Image Understanding Workshop. (1981) 121–130
4. Limb, J.O., Murphy, J.A.: Estimating the velocity of moving images in television signals. *Computer Graphics and Image Processing* **4** (1975) 311–327
5. Barron, J., Fleet, D., Beauchemin, S., Burkitt, T.: Performance of optical flow techniques. Technical Report TR 299, Dept. Comp. Sci., Univ. Western Ontario (1993)
6. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. *International Journal of Computer Vision* **12** (1994) 43–77
7. Black, M.J.: Frequently asked questions. Technical report, Department of Computer Science, Brown University, <http://www.cs.brown.edu/people/black/> (2005)
8. Ngo, C.W., Pong, T.C., Zhang, H.J.: Motion analysis and segmentation through spatio-temporal slices processing. *IEEE Trans. Image Processing* **12** (2003) 341–355
9. Knutsson, H., Andersson, M.: Loglets: Generalized quadrature and phase for local spatio-temporal structure estimation. In: Proceedings of the Scandinavian Conference on Image Analysis, SCIA 2003, Springer- Verlag (2003) 741–748
10. Austvoll, I.: Directional filters and a new structure for estimation of optical flow. In: Proc. Int. Conf. Image Proc. Volume II., IEEE Signal Processing Soc. Los Alamitos, CA, USA (2000) 574–577
11. Austvoll, I.: Motion Estimation using Directional Filters. PhD thesis, NTNU/HIS, PoBox 2557 Ullandhaug, N-4091 Stavanger, Norway. (1999)
12. Pauwels, E.J., Van Gool, L.J., Fiddelaers, P., Moons, T.: An extended class of scale-invariant and recursive scale space filters. *IEEE Trans. Pattern Anal. and Machine Intell.* **17** (1995) 691–701
13. Fleet, D., Jepson, A.: Computation of component image velocity from local phase information. *International Journal of Computer Vision* **5** (1990) 77–104
14. Foroosh, H.: Pixelwise-adaptive blind optical flow assuming nonstationary statistics. *IEEE Trans. Image Processing* **14** (2005) 222–230
15. Brox, T., Bruhn, A., Papenberg, N., Joachim, W.: High accuracy optical flow estimation based on a theory for warping. In: Proc. European Conference on Computer Vision, ECCV 2004, Springer- Verlag (2004) 25–36
16. Bab-Hadiashar, A., Suter, D.: Robust optical flow computation. *Int. J. of Computer Vision* **29** (1998) 59–77
17. Liu, H., Chellappa, R., Rosenfeld, A.: Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Trans. Image Processing* **12** (2003) 1170–1180
18. Farnebäck, G.: Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In: Proc. Int. Conf. Comp. Vis. Volume I. (2001) 171–177
19. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. Image Processing* **7** (1998) 703–719

20. Ju, S., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'96. Volume 2182., San Francisco, CA (1996) 307–314
21. Chen, L.F., Liao, H.Y.M., Lin, J.C.: Wavelet-based optical flow estimation. *IEEE Trans. Circuits and Systems for Video Tech.* **12** (2002) 1–12
22. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers (1995)
23. Robinson, D., Milanfar, P.: Fundamental performance limits in image registration. *IEEE Trans. Image Processing* **13** (2004) 1185–1199

# A Versatile Model-Based Visibility Measure for Geometric Primitives

Marc M. Ellenrieder<sup>1</sup>, Lars Krüger<sup>1</sup>, Dirk Stöbel<sup>2</sup>, and Marc Hanheide<sup>2</sup>

<sup>1</sup> DaimlerChrysler AG, Research & Technology, 89013 Ulm, Germany

<sup>2</sup> Faculty of Technology, Bielefeld University, 33501 Bielefeld, Germany

**Abstract.** In this paper, we introduce a novel model-based visibility measure for geometric primitives called visibility map. It is simple to calculate, memory efficient, accurate for viewpoints outside the convex hull of the object and versatile in terms of possible applications. Several useful properties of visibility maps that show their superiority to existing visibility measures are derived. Various example applications from the automotive industry where the presented measure is used successfully conclude the paper.

## 1 Introduction and Motivation

Finding viewpoints from which certain parts of a known object are visible by a two-dimensional, perspective sensor is not a straightforward task. Especially in complex environments like industrial inspection or object recognition applications, where several geometric primitives (often called  $\dots$  [11]) on complex objects have to be inspected, finding an easy to calculate, versatile, storage-effective and not too simplifying visibility measure is of great importance. Up until now, several visibility measures with their particular advantages and disadvantages have been developed. However, most of them are tailor-made for specific applications so that the underlying algorithmic and conceptual framework cannot be used in other tasks. At the same time, they do not provide a consistent modeling of visibility for different feature-types like points, lines, polygons or volumetric features. Also, feature covisibility is addressed by only a few of the many existing visibility measures. Restricting the sensor positions to very few viewpoints further limits possible applications.

One of the first visibility measures for surface areas is the aspect graph by Koenderink et.al. [8]. Aspect graphs assume the object's model to be centered at the origin of a sphere which encapsulates the entire model. Every possible view of the model can then be represented as a point on that sphere. Each equivalent view on the sphere is represented by a node in the aspect graph, with the connection between graph nodes representing a possible transition between views. These transitions are due to changes in the occlusion relationships between object surfaces. However, no method whatsoever of generating the three-dimensional viewpoint region of an object's aspect graph node is presented in

this paper. In [2], Cowan and Kovesi show a way to actually generate an aspect graph, although only for simple polyhedra. They describe a method to calculate the three-dimensional region where a convex polyhedral object  $\mathcal{O}$  occludes a portion of a convex polyhedral surface  $\mathcal{S}$ . The boundary of the occlusion zone is described by a set of separating support-planes. In the case of multiple occluding objects, the union of all non-occlusion zones is calculated. The presented algorithm has quadratic computational complexity in the number of edges of the polygon. This is of significant disadvantage in complex scenes and real applications.

Tarabanis et.al. [11] have presented a method of computing the spacial visibility regions of features. They define a feature as a polygonal and manifold subset of a single face of a polyhedral object. The visibility region of a feature is defined as the open and possibly empty set consisting of all viewpoints in free space for which the feature is visible in its entirety. Instead of calculating the visibility region directly, Tarabanis et.al. devised a three-step algorithm that calculates the occlusion region of a feature  $\mathcal{T}$  in linear time (in terms of object vertices). The occlusion region is the complementary area to the visibility region with respect to free space. For each element of a subset  $\mathcal{L}$  of the faces of the polyhedral object, the (polyhedral) occluding region is calculated in a similar manner to the method shown by Cowan and Kovesi [2]. The elements of  $\mathcal{L}$  are those faces that satisfy certain topological and morphological properties with respect to  $\mathcal{T}$ . The occluding regions of all elements of  $\mathcal{L}$  are merged into the complete polyhedral occlusion region  $\mathcal{O}$  of the feature  $\mathcal{T}$ . A check for visibility of  $\mathcal{T}$  from a certain viewpoint can thus be reduced to a point-in-polyhedron classification. However, since the polyhedral occlusion region  $\mathcal{O}$  has to be stored as a whole, the presented method requires a considerable amount of storage memory. This makes it difficult to employ in scenarios with highly complex parts.

Another method of calculating the visibility region of a feature is presented by Trucco et.al. [12]. Their method restricts the possible sensor viewpoints to positions at manually fixed intervals on a spherical grid surrounding the object. The viewing direction at each grid point connects the viewpoint with the center of the spherical grid. Visibility of a feature is determined by rendering the object from each viewpoint and counting the number of pixels of the feature in the resulting image. Covisibility, i.e. the visibility of several features at once, can be determined by counting the collective number of pixels. An advantage of this approach is that it yields a quantitative and not just boolean visibility measure for each viewpoint. Also, in terms of storage memory, the approximate visibility space is very efficient. However, the restriction to a spherical grid, and the high computational complexity for the rendering process limits its use to rather simple inspection tasks.

Various other visibility measures exist in the literature. Some publications address visibility in terms of a scalar function  $\mathcal{V}$  that is evaluated for each viewpoint. Khawaja et.al. [7] use the number of visible features, the number of visible mesh-faces on each feature, and the number of image pixels associated with each face as parameters of an empirically postulated formula. The necessary

parameters are generated for each viewpoint by rendering the model of the inspected object from this viewpoint. Other publications, e.g. [1], use even simpler methods: the dot-product of the viewing direction and the surface normal of the inspected feature. If the dot-product is negative, the feature is considered to be visible. Hence, visibility can only be determined correctly for strictly convex objects.

As we have seen, existing visibility measures do not account for all of the points mentioned above. Especially the lack of versatility concerning both the variety of possible applications and the correctness of the visibility determination for 1d, 2d and 3d features is apparent. In the remainder of this paper, we therefore want to introduce the concept of visibility maps to determine the visibility of arbitrary geometric primitives. We will show some of their properties and demonstrate their versatility in various machine vision applications.

## 2 Visibility Maps

The term „...“ is very generic. It is used for example in computer graphics as a synonym for a graph characterizing the visible triangles of a mesh. In this paper, we use the term to describe a matrix that is used to determine the visibility of a geometric primitive. In principle, a visibility map is defined for points on the surface of an object. It is calculated by projecting the inspected object (and possibly the whole scene) onto a unit sphere centered at the point on the object for which the map is calculated. The unit sphere is then sampled at constant azimuth / elevation intervals  $\nu$  and the boolean information whether something has been projected on the current point on the sphere or not, is transcribed into a matrix called „...“. Fig. 1 illustrates this concept.

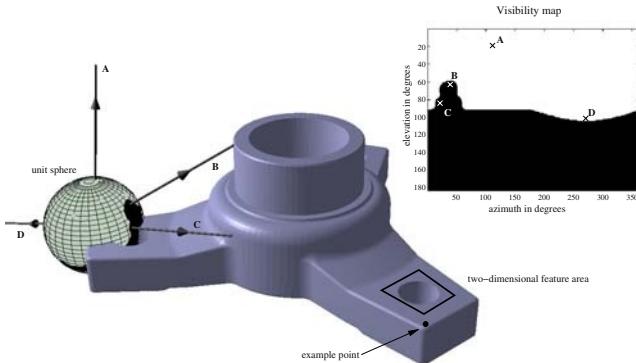
### 2.1 Calculating Visibility Maps

Calculating visibility maps can be implemented effectively, if the object surface geometry is given as a triangulated mesh with three-dimensional vertex coordinates  $v_i$ . The vertices are projected onto a unit sphere centered at the point whose visibility has to be calculated. Without loss of generality, we assume this point to be the origin of the coordinate system. Using the four-quadrant arcus-tangent function the vertices' spherical coordinates  $\theta$  (azimuth),  $\phi$  (elevation) and  $r$  (radius) result to

$$\begin{aligned}\theta(\tilde{v}_i) &= \arctan 2(v_{i,y}, v_{i,x}), \\ \phi(\tilde{v}_i) &= \frac{\pi}{2} - \arctan 2\left(v_{i,z}, \sqrt{v_{i,x}^2 + v_{i,y}^2}\right), \text{ and} \\ r(\tilde{v}_i) &\equiv 1.\end{aligned}\tag{1}$$

Suppose, two vertices  $v_i$  and  $v_j$  are connected by a mesh-edge. Then, the mesh-edge is sampled at  $k$  intervals. The sampled edge points are then projected onto

the sphere and an approximate spherical mesh-triangle is constructed by connecting the projected edge samples using BRESENHAM's algorithm. The resulting triangle-outline is filled using a standard flood-fill algorithm. This process is repeated for every triangle of the object. Afterwards, the unit sphere is sampled in both azimuth and elevation direction at intervals  $\nu$  and the result whether something has been projected or not is transcribed into the matrix  $M$ , i.e. the visibility map. To account for numerical errors and to get a smooth visibility map we further apply standard dilation / erosion operators to  $M$ . The computational complexity of this method is  $\mathcal{O}(n)$  for objects comprised of  $n$  triangles. An example visibility map can be seen in Fig. 1.



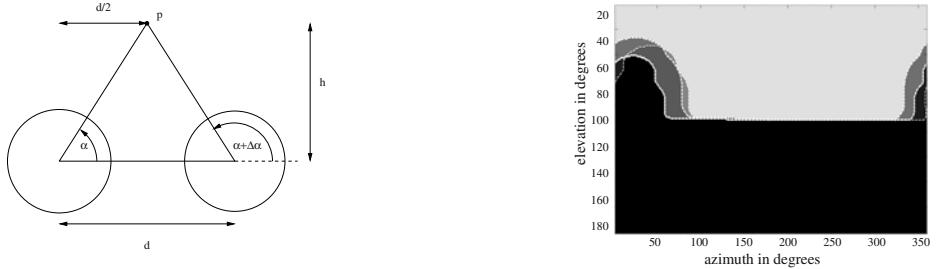
**Fig. 1.** Visibility map of a point on the right arm of the flange. The same point on the right arm of the flange is highlighted. Black regions in the visibility map represent viewing directions, where a camera positioned at the center of the sphere would see parts of the flange. The position of four example viewing directions (A-D) are marked in the map. For illustration purposes, the size of the unit sphere has been exaggerated

## 2.2 Distance Transform

Sometimes, it can be useful to quickly determine viewing directions where the visibility of features is as stable as possible towards slight viewpoint deviations. This means, the sensors should be positioned as far away as possible from the occlusion-zone boundaries. One way to achieve this is to calculate the distance transform of a visibility map. However, since visibility maps are spherical projections, we need to use a spherical rather than an Euclidean distance measure. Using the Haversine function  $h(x) = \sin^2(x/2)$ , the distance of two points  $p_1 = (\theta_1, \phi_1)$  and  $p_2 = (\theta_2, \phi_2)$  on the unit sphere can be expressed as

$$d(p_1, p_2) = 2 \cdot \arctan 2 \left( \sqrt{a}, \sqrt{1-a} \right), \quad (2)$$

where  $a = h(\theta_2 - \theta_1) + \cos(\theta_1) \cos(\theta_2) h(\phi_2 - \phi_1)$ . By convention, we define  $d < 0$  for visible viewing directions and  $d > 0$  for occluded directions. The actual distance transformation is calculated using standard propagation algorithms [3] which efficiently exploit the transitivity of the minimum relation.



**Fig. 2.** Geometric relations for 2d feature visibility (left) and full feature visibility map (5) of the polygonal area shown in Fig. 1. The different corner point visibility maps are shown for illustration purposes

### 2.3 Visibility of 2D and 3D Features

In many applications a visibility measure for 2D or even 3D features is required. By concept however, the visibility map is defined for one single point on the surface of an object. Nevertheless, it can be generalized for higher-dimensional geometric primitives, i.e. lines, polygonal areas, and volumes, if it is assumed that the camera is far away at distance  $h$  in comparison to the maximum extension  $d$  (i.e. the length of the longest eigenvector) of the primitive. This is equal to the assumption of using the same camera direction (azimuth / elevation) for each of the  $N$  corner points of the primitive. The resulting error is depending on the relation  $d/h$  and can be estimated as follows: let  $p$  be a 3D-point that is projected onto two unit spheres at located at a distance  $d$ . Figure 2 shows the geometric relations in the plane spanned by the sphere centers and  $p$ . The point is projected onto the first sphere at an elevation angle  $\alpha$  and at elevation  $\alpha + \Delta\alpha$  onto the second. It is clear, that for any fixed  $h$ ,  $\Delta\alpha \rightarrow \max$  if  $p$  is located in the middle of the spheres at  $d/2$ . We have

$$\Delta\alpha = \arctan\left(\frac{-2h}{d}\right) - \arctan\left(\frac{2h}{d}\right). \quad (3)$$

In order to get a value of  $\Delta\alpha$  that is smaller than the angular spacing  $\nu$  of the visibility map, we need to find  $|\Delta\alpha(\frac{d}{h})| < \nu$ . From (3) we get

$$\frac{d}{h} \leq -\frac{1}{2} \tan \frac{\nu}{2}. \quad (4)$$

For a typical sampling interval of  $\nu = 1$ , this results to  $d/h \leq 0.004$ .

For higher dimensional primitives, there are basically two notions of visibility: full visibility and partial visibility. In industrial inspection applications, full visibility of a feature is often required, e.g. to check for complete visibility of bar codes. In other applications, e.g. object recognition applications, partial visibility might however be sufficient. For primitives, where full visibility is required, it can be calculated by forming a union of the visibility maps of each corner point, leading to

$$M = \bigcup_{k=0}^{N-1} M_{k,\text{corner}}. \quad (5)$$

This concept is applicable for 2D polygonal areas, e.g. single meshes, or parts of 3D primitives, e.g. several sides of a cube. It can be further extended to the covisibility of several features, resulting in the  $\bar{M}$ . If full visibility is not required or possible, e.g. in case of three-dimensional primitives, (5) will not yield a correct visibility measure. Nevertheless, the visibility map can also be used, if the visibility maps of the primitive vertices  $M_{k,\text{corner}}$  are combined into,  $\bar{M}$ , by

$$M = \sum_{k=0}^{N-1} 2^{k-1} M_{k,\text{corner}}. \quad (6)$$

Then, the visibility of each vertex can be evaluated separately. Using the distance transform of the visibility maps of two vertices  $p_1$  and  $p_2$  (2) it is also possible to calculate the visible length  $d_{\text{vis}}$  of a mesh-edge connecting these vertices. Figure 3 shows the geometric relations in the plane  $E$  spanned by  $p_1$ ,  $p_2$  and viewpoint  $v$ . The plane cuts a great circle from both visibility maps. All angles and distances used in the following are measured in this plane. For the sake of simplicity, we define a coordinate system with unit vectors  $x_E$  and  $y_E$  originating in  $p_1$ . Then, the  $x_E$ - $y_E$  components of  $k$  are given by

$$k_{x_E} = \left( 1 - \frac{\tan(\alpha_1 + \delta_1)}{\tan(\alpha_2 + \delta_2)} \right)^{-1} \cdot d, \quad k_{y_E} = d \cdot \tan(\alpha_1 + \delta_1). \quad (7)$$

Angles  $\alpha_k$  and  $\delta_k$  can be directly drawn from the visibility maps and their distance transforms. Calculating the intersection of  $g_3$  and the  $x_E$ -axis results in

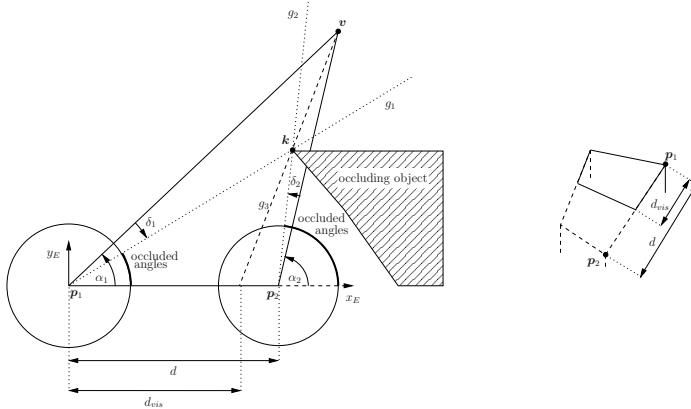
$$d_{\text{vis}}(d) = (v_{x_E} - v_{y_E}) \frac{v_{x_E} - k_{x_E}}{v_{y_E} - k_{y_E}}. \quad (8)$$

Here,  $v_{x_E}$  and  $v_{y_E}$  describe the coordinates of the viewpoint projected onto the plane  $E$ . If applied to each mesh-edge, (8) allows to calculate the visible area or volume of the primitive directly from the visibility maps (Fig. 3). In viewpoint planning applications this property can be used to directly assign a quantitative visibility value to a viewpoint.

## 2.4 Visibility Ratio and Memory Efficiency

By using visibility maps, it is possible to estimate the size of the space, from which one or more features are completely visible. This can be used to determine, whether e.g. a sensor head mounted on a robotic arm can be positioned such that certain primitives are visible. For this, we define the term,  $V(F_j)$ , of a single feature  $F_j$ :

$$V(F_j) = \frac{\text{visible area of } M(F_j)}{\text{total area of } M(F_j)}. \quad (9)$$



**Fig. 3.** The geometric relations in the plane spanned by  $p_1$ ,  $p_2$  and viewpoint  $v$  (left) for calculating the visible area of a partially visible mesh facet (right). All units are measured in this plane. Angles  $\delta_{1,2}$  are derived from the visibility maps' distance transforms (2)

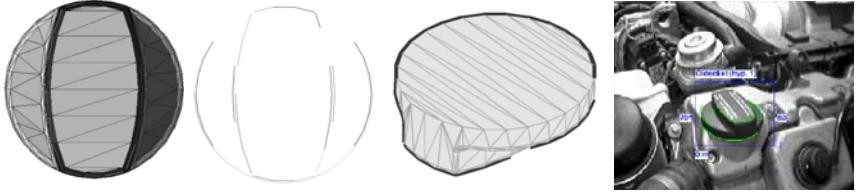
This property can be interpreted as the relation of the visible solid angle of the feature to the full sphere. Equally, we define the combined feature visibility ratio  $\bar{V}(C_i)$  of the features associated to camera  $i$  by using the combined feature visibility map  $\bar{M}(C_i)$ . It has to be noted that for full feature visibility maps  $\bar{V} < V \leq 0.5$  for all two-dimensional primitives other than a single line, since they are on the surface of the inspected objects and are thus not visible from viewpoints on the backside of the object. One can therefore assume that there is at least one large connected region in the visibility map. Hence, visibility maps can be stored very effectively by using simple run-length-encoding as a means to compress them.

### 3 Applications Using Visibility Maps

To show the versatility of the visibility map, we are going to present several systems using the presented concepts in various applications from object recognition to viewpoint planning. In our opinion, this versatility together with the simplicity of its concept renders the visibility map superior to other existing visibility measures.

#### 3.1 Object Recognition and Pose Estimation

One of the first applications, where visibility maps have been used, was object recognition. In [9] a system for multi-feature, multi-sensor classification and localization of 3D objects in 2D image sequences is presented. It uses a hypothesize-and-test approach to estimate type and pose of objects. Characteristic Localized Features (CLFs), e.g. contours, 3D corners, etc., are extracted from the geomet-



**Fig. 4.** From left to right: An oil cap, its CLF graph seen from two different viewpoints, aligned to the image. Visibility of the CLFs was calculated using visibility maps

ric models of the different objects and viewpoint dependant graphs of the CLFs projected onto the image plane are generated for each pose and object type hypothesis. By using elastic graph matching algorithms [6], the graph is aligned with the features in the image. Viewpoint dependant graph rendering is only possible, since visibility of each CLF was calculated using the visibility map. The system is used for both optical inspection for quality control and airport ground-traffic surveillance. An example CLF-graph and the recognized object type and pose is shown in Fig. 4. Since there are typically several hundreds of CLFs per object whose visibility has to be evaluated several times, both storage memory and speed of the employed visibility measure are crucial.

### 3.2 Optimal Sensor-Feature Association

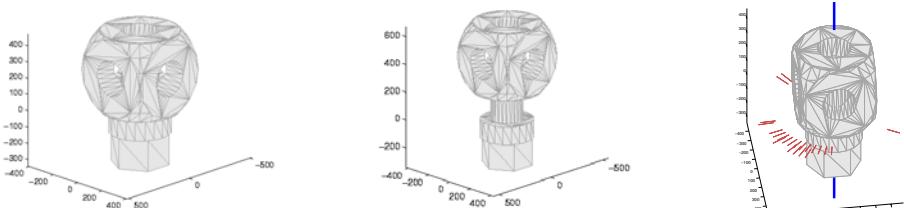
The cost of automatic inspection systems for quality control directly depends on the number of installed sensors. Clearly, more than  $r$  sensors for  $r$  features is therefore not an option, less, i.e.  $k < r$ , sensors would even be better. For  $r$  feature areas, there are  $2^r - 1$  possible feature combinations that can be assigned to one sensor. To find the optimal association, we need to define a criterion that compares different combinations. The combined feature visibility map's visibility ratio can be used for finding an optimal assignment matrix  $C$  of size  $k \times r$  with  $k \leq r$  whose column-sums equal to 1 and whose elements  $C_{ij}$  are 1, if feature  $j$  is associated to camera  $i$ . Using a row vector  $\bar{V}$  that represents the associated features of camera  $i$ , the  $\tilde{V}$  of an assignment matrix  $C$  with  $k$  rows

$$\tilde{V}(C) = \frac{1}{k} \sum_{i=1}^k \bar{V}(C_i) \quad (10)$$

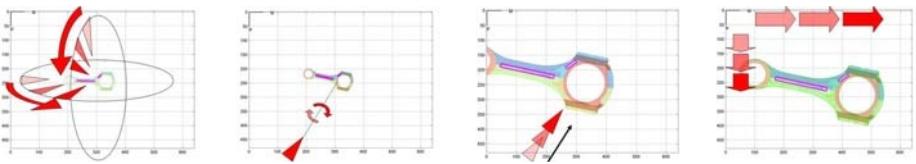
to compare all possible assignments is introduced in [4]. Based on this equation, an algorithm to find the optimal sensor-feature association matrix  $C_{\text{opt}}$  with a minimum number of sensors is also presented.

### 3.3 Entropy Based Viewpoint Selection

Stel et al. [10] extend the notion of CLFs to an entropy based viewpoint selection method. There, the best viewpoints for distinguishing different aggregate models, i.e. different bolt-nut combinations (Fig. 5), and their respective 2D



**Fig. 5.** Two different nut-bolt aggregations (left / middle). The entropy index mapped into visibility maps according to [10] determines the 20 best (light lines) and worst viewing directions (bold lines from above / beneath) to distinguish them (right)



**Fig. 6.** Simple four step algorithm to find a good initial viewpoint using visibility maps

projections are calculated. The collective entropy mapped onto a visibility map is used to derive a distinction measure for different viewing directions. High entropy determines bad viewing directions, low entropy values mark good ones.

### 3.4 Optimal Camera and Illumination Planning

Another application using visibility maps is presented in [4]. There, the optimal viewpoints for industrial inspection tasks are calculated from a geometric model of the inspected objects (Fig. 6). The distance transform of the visibility map allows to automatically position the sensor in a viewing direction that is far from vanishing points in the viewing volume. Further refinements regarding additional viewpoint requirements, e.g. minimum resolution or viewing angle, are implemented using convex scalar functions dependant on the viewpoint. Using the distance transformation of the feature visibility maps, a scalar cost can be assigned to each six-dimensional viewpoint. The final, optimal viewpoints are found by minimizing the scalar cost functions. Similarly, the visibility map can be used to find an illumination position [5], from which a desired feature illumination condition, e.g. specular or diffuse reflection, can be observed.

## 4 Summary and Conclusion

We have presented a versatile model-based visibility measure for geometric primitives called visibility map. It is easy to calculate, memory efficient, quick to use, and provides an accurate way of determining the visibility of 1d, 2d, or 3d geometric primitives of triangulated meshes from viewpoints outside the convex hull

of the whole object. Several applications that prove the versatility and usefulness of this concept for object recognition, pose estimation, as well as sensor and illumination planning are presented. Since a visibility map has to be calculated only once, a test for visibility from a specific viewing direction comprises only a table lookup. In our opinion, this versatility together with the simplicity of its concept renders the visibility map superior to other existing visibility measures. One shortcoming of the visibility map, however, is the fact that visibility can only be determined correctly for viewpoints outside of the complex hull of the object.

## References

1. S. Y. Chen and Y. F. Li, A method of automatic sensor placement for robot vision in inspection tasks, in *Proc. IEEE Int. Conf. Rob. & Automat.*, Washington, DC, May 2002, pp. 2545–2550.
2. C. Cowan and P. Kovesi, Automatic sensor placement from vision task requirements, *IEEE Trans. Pattern Anal. Machine Intell.*, 10 (1988), pp. 407–416.
3. O. Cuisenaire, Distance transformations: fast algorithms and applications to medical image processing, PhD thesis, Univ. cath. de Louvain, Belgium, Oct. 1999
4. M. M. Ellenrieder and H. Komoto, Model-based automatic calculation and evaluation of camera position for industrial machine vision, in *Proc. SPIE Computational Imaging III*. 2005.
5. M. M. Ellenrieder et al., Reflectivity Function based Illumination and Sensor Planning for industrial inspection, *Proc. SPIE Opt. Metrology Symp.*, Munich, 2005
6. E. Kefalea, O. Rehse, and C. v. d. Malsburg, Object Classification based on Contours with Elastic Graph Matching, *Proc. 3rd Int. Workshop Vis. Form*, 1997
7. K. Khawaja et al. , Camera and light placement for automated visual assembly inspection, in *Proc. IEEE Int. Conf. Robotics & Automation*, Minneapolis, MN, April 1996, pp. 3246–3252.
8. J. J. Koenderink and A. J. van Doorn, The internal representation of solid shape with respect to vision, *Biol. Cybern.*, 32 (1979), pp. 151–158.
9. T. Klzow and M. M. Ellenrieder, A general approach for multi-feature, multi-sensor classification and localization of 3d objects in 2d image sequences, in *Proc. SPIE Electronic Imaging Conf.*, vol. 5014, 2003, pp. 99–110.
10. D. Stel et al., Viewpoint selection for industrial car assembly, in *Springer LNCS*, vol. 3175 - Proc. 26<sup>th</sup> DAGM Symp. 2004, pp. 528–535.
11. K. A. Tarabanis et al., Computing occlusion-free viewpoints, *IEEE Trans. Pattern Anal. Machine Intell.*, 18 (1996), pp. 279–292.
12. E. Trucco et al., Model-based planning of optimal sensor placements for inspection, *IEEE Trans. Robot. Automat.*, 13 (1997), pp. 182–194.

# Pose Estimation of Randomly Organized Stator Housings

Thomas B. Moeslund and Jakob Kirkegaard

Laboratory of Computer Vision and Media Technology,  
Aalborg University, Denmark  
`tbm@cvmt.dk`

**Abstract.** Machine vision is today a well-established technology in industry where especially conveyer belt applications are successful. A related application area is the situation where a number of objects are located in a bin and each has to be picked from the bin. This problem is known as the automatic bin-picking problem and has a huge market potential due to the countless situations where bin-picking is done manually. In this paper we address a general bin-picking problem present at a large pump manufacturer, Grundfos, where many objects with circular openings are handled each day. We pose estimate the objects by finding the 3D opening based on the elliptic projections into two cameras. The ellipses from the two cameras are handled in a unifying manner using graph theory together with an approach that links a pose and an ellipse via the equation for a general cone. Tests show that the presented algorithm can estimate the poses for a large variety of orientations and handle both noise and occlusions.

## 1 Introduction

Machine vision is today a well-established technology in industry and is becoming more and more widespread each year. The primary area of success for machine vision is conveyer belt applications, e.g., quality control and robot guiding. The latter is the task of providing robots with positioning data for objects located on a moving conveyer belt. Normally a machine vision system is combined with some kind of mechanical device that ensures that only one object is presented to the system at a time, i.e., no occlusion is present.

A related application area is the situation where a number of objects are located in a bin, see figure 1, and each has to be picked from the bin and placed on a conveyer belt in a predefined pose. This problem is known as the automatic bin-picking problem [13]. A general solution to this problem has a huge market potential due to the countless situations where bin-picking is done manually.

Many different approaches to the bin-picking (and related) problems have been suggested. They can be divided into two categories: model-based approaches and appearance-based approaches.

In the „*„*“ approaches, a large number of images are obtained of the object and these different appearances are then used when pose estimat-

ing the objects. The immediate advantage of this is, that the scene data and the model data are expressed in the same terms together with its capability of handling objects with no apparent features like lines or corners. The disadvantage of „...“ methods is that the appearance of an object is highly dependent on illumination, viewpoint and object pose [14]. For example, in [1] between 4.000 and 12.000 images of different viewpoints are applied to learn the appearance of the pose of a particular object. In [10] a model of the object is represented as a probability distribution describing the range of possible variations in the object's appearance.

The „...“ approach on the other hand, represents objects through features, their type and spatial relations. The advantage of „...“ representations is, that they generate compact object descriptions and offer some robustness against occlusion and some invariance with respect to illumination and pose variations. The disadvantage is that the feature representation cannot be compared directly with the intensity images and that a scene feature extraction therefore is needed. For example, in [7] a wire frame of the model is used and compared with edges in the image. In [9] the CAD (Computer Aided Design) model of the object is used together with relevant object feature points pointed out by the user. In [8] distinct corner features are found in two images and triangulated in order to find the pose of the object.

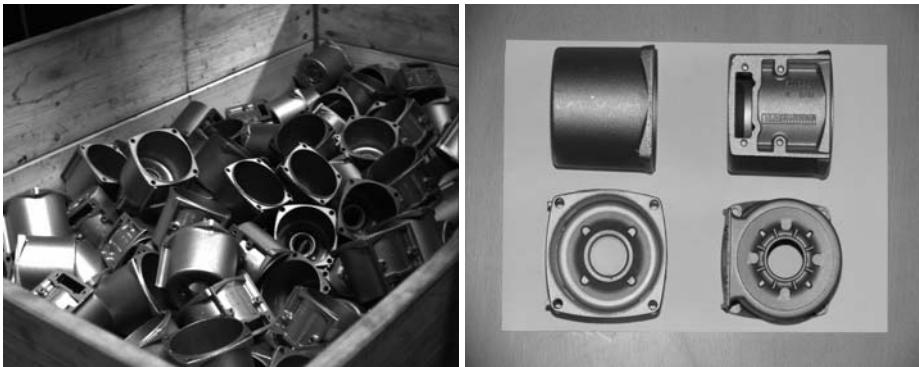
### 1.1 Content of this Work

Grundfos [15] is one of the world's leading pump manufacturers and produces more than 25.000 pumps per day using more than 100 robots and 50 machine vision systems. Some of the robots and machine vision systems are used in fully automatic bin-picking applications with well organized objects. However, a large number of unsolved bin-picking problem remain and therefore Grundfos are interested in general solutions for handling the bin-picking problem. This paper presents research in this context.

From a machine vision point of view the bin-picking problem is extremely difficult due to the very high number of objects potentially occluding each other and changing the illumination locally due to shadows and reflections. To make the problem tractable we reformulate it to be a matter of picking one and only one object from the bin and „...“ finding the pose of this isolated object. The idea being that the combined complexity of the two new problems is less than the complexity of the original problem. The latter problem can be handled by showing the picked object to a camera during the flight from the bin to the conveyer belt, and this pose estimation problem of a single known object in a controlled environment can "easily" be solved. What remains is to find a way of picking one and only one object from the bin.

Many of the objects being produced at Grundfos can roughly be considered as having a cube-like shape, i.e., six sides. Our approach is to have different picking tools for the robot corresponding to the different sides of the cube. We then view the problem of finding and picking an object, as a matter of finding one of the sides and then apply the appropriate picking tool. By looking at the

object in figure 1 it can be seen, that it has six "sides" where three are similar (the smooth sides). To find and pick an object having this type of side facing the camera can, e.g., be done using structured light and stereo-vision followed by a vacuum gripping device [2].



**Fig. 1.** **Left:** A bin containing randomly organized Stator Housings. **Right:** The Stator Housing object shown from four different viewpoints on a piece of A4 paper for reference

In this work we seek a solution to the machine vision problem of finding the "side" representing the opening of such objects, see figure 1, and as a case study we use a bin containing Stator Housings, see figure 1.

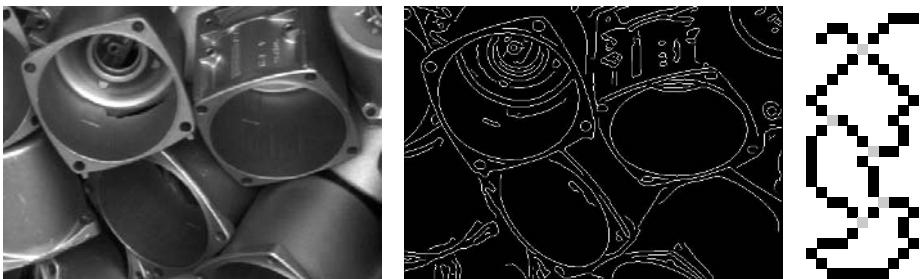
The "opening-side" of an object can be characterized by a circle in 3D which projects to an ellipse in an image. Therefore, the problem we are addressing is that of estimating the pose of a circle given its elliptic projection into two cameras. The paper is structured as follows. In section 2 edge pixels belonging to the same ellipse are grouped and fitted to an ellipse. In section 3 the ellipses found in the cameras are used to estimate the pose of the circles. In section 4 the results are presented and in section 5 a conclusion is given.

## 2 Estimating the Ellipses

As described above our strategy is to estimate the pose of the objects based on elliptic features in the stereo images. In this section we first describe how the edges extracted from the intensity images are grouped into meaningful segments (denoted  $\dots$ ) each corresponding to an elliptic arch. Secondly, we describe how each edgel is fitted to an ellipse.

Initially we apply the,  $\dots$  edge detector [14] as it not only finds edges, but also groups them into one pixel wide connected segments - edgels, see figure 2. The edgels are then post-processed in two steps in order to ensure that they each contain pixels from one and only one ellipse. The first step is carried out

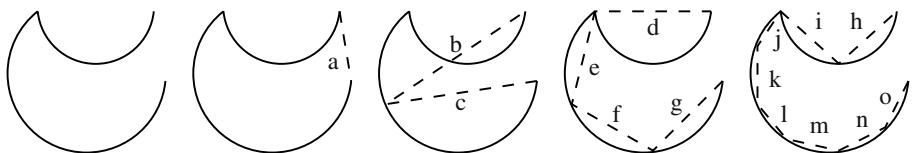
in order to ensure that each edgel only contains two end-points. This is done by removing crossing point in the edge images, i.e., pixels with more than two neighbors in an 8-connectivity sense, see figure 2.



**Fig. 2. Left:** Small part of an input image. **Middle:** Edge image. **Right:** Crossing points in gray

As seen in figure 2 edge pixels from different ellipses are often part of the same edgel. The second step therefore removes large concavities by dividing the edge pixels into separate edgels by removing the pixels with a too high curvature. Using standard measures for the curvature turned out to be too sensitive when evaluating the derivatives at particular points. As a result we follow a different approach. Instead of measuring the curvature at one point we filter the curve by dividing it into a number of straight line-segments and then measure the angle between adjacent line segments. We use a modified version of [11].

A typical curve often appearing in the Stator Housing edge images is the one shown in the left most part of figure 3 and clearly has a large concavity at the point of intersection.



**Fig. 3.** The principle behind dividing an edgel into straight line-segments

An edgel is segmented into straight lines by first making the crudest approximation possible, i.e., a straight line connecting the end points. The algorithm then proceeds recursively by dividing each approximating line segment into two

line segments at the point of largest deviation, i.e., at the curve point with the largest distance to the approximating line segment.

Each line segment is assigned a significance value, which is the ratio between the length of the line and the largest deviation from the points it approximates. This can be interpreted as, the shorter a line segment is the less deviation is tolerated.

The algorithm continues until the most significant lines are found, i.e., a line segment is only subdivided if its significance is less than the sum of the significances of the children. Furthermore, if the length of the line or the deviation becomes too small the sub-division is also terminated.

The problem can be posed as a graph search where each node in the graph represents a line segment and has a weight equal to the significance of the line. The tree is then traversed breadth-first and bottom-up searching for nodes that have greater significance than all their children together.

## 2.1 Ellipse Fitting

After having removed the edge pixels resulting in multiple end-points or large concavities, we are left with a number of edgels<sup>1</sup>. To each of these is fitted an ellipse using the direct fitting method by Fitzgibbon et al. [4], which is based on minimizing the algebraic distance between the edge pixels and an ellipse. The method is based on solving a generalized eigenvalue problem constructed from the edgel's points for obtaining the optimal ellipse parameters, i.e., a closed-form solution is obtained.

This algorithm results in a number of ellipses some of which might be very similar. The reason being that one ellipse might be represented as a number of edgels due to noise. We therefore find the ellipses which are similar and merge the corresponding edgels into a new set of pixels which is used to find the parameters for the joint ellipse. The similarity measure is based on a box classifier in the five dimensional space spanned by the ellipse parameters. Finally ellipses with unrealistic parameters are ignored.

## 3 Pose Estimation

Given a number of ellipses estimated in each image we are now faced with the problem of calculating the corresponding 3D circles. We apply the approach described in [12] where the idea is to find a general cone that both embodies the ellipse and the corresponding 3D circle with radius of the Stator Housing openings. This approach works well but it has the same problem as similar algorithms, namely that two different 3D circles correspond to the same ellipse in the image (and the same general cone). See [5] for details. We therefore need to validate which of the two is the correct solution.

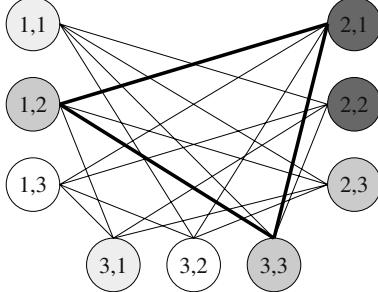
---

<sup>1</sup> Note that edgels with less than 30 [pixel] are ignored altogether.

### 3.1 Circle Pose Validation

In order to solve the validation problem we apply an ellipse matching procedure between the camera frames in order to select the correct circle candidate.

The problem is posed as a graph search problem as shown by the figure 4. A node in the graph represents a match between an ellipse in the left and right frame (e.g., 1,3 indicates a match between the 1st left and the 3rd right ellipse), while an edge between two nodes in the graph indicates compatibility between two matches.



**Fig. 4.** A graph representing the problem of matching a set of ellipses between the camera frames. Darker node fillings indicate higher weighted nodes and the bold lines indicate the maximally weighted clique

The association graph is build by creating a node for every pair of ellipses from each camera frame. Many of these nodes can immediately be discarded by investigating the ellipse parameters, while the remaining nodes are given a weight according to  $1/(c_{dist} + a_{dist} + b_{dist} + i_{dist})$ , where the denominator accumulates distances between the horizontal ellipse centers ( $c_{dist}$ ) together with differences in major axes ( $a_{dist}$ ), minor axes ( $b_{dist}$ ) and inclination angle ( $i_{dist}$ ) of the ellipse parameters<sup>2</sup>.

The problem of finding the best match between the ellipses in the left and right camera frames is then reduced to the problem of finding the maximally weighted set of mutually compatible nodes, i.e., the maximally weighted clique in the association graph [3]. This NP complete problem is approached by stochastic optimization (simulated annealing with linear cooling scheme, [6]) of the function given by equation 1.

$$f(\mathbf{x}) = \sum_{i=1}^n w_i x_i - \lambda \sum_{i=1}^n w_i c_i \quad (1)$$

where  $f(\cdot)$  calculates a gain for the given set of graph nodes defined by the membership vector  $\mathbf{x}$ . The length of the membership vector is given by  $n$  and the

<sup>2</sup> Note that rectified images are used.

term  $w_i$  states the weight while the binary variable  $x_i$  denotes clique membership of the  $i'th$  node. The binary variable  $c_i$  states whether the  $i'th$  node can be part of the clique stated by the membership vector  $\mathbf{x}$ . The state of  $c_i$  for a given node is determined from the inverse graph, i.e., for a set of nodes to form a clique, no two clique nodes must be connected in the inverse graph. This is formally stated by equation 2, where  $E$  is the edge set for the association graph. The factor  $\lambda$  (set to unity in the current implementation) is included to control the balance between the gain and the penalty.

$$\forall i, j \in \overline{E} : x_i + x_j \leq 1 \quad (2)$$

The result of the optimization is a membership vector indicating the nodes in the (approximated) maximally weighted clique (i.e., a number of compatible ellipse matches). Each match states that two pose candidates in the left frame have been matched with two pose candidates from the right frame. The final pose candidate for each match is then chosen by transforming the two right camera frame pose candidates into the left camera frame (stereo rig is assumed calibrated) and then finding which of the two possible combinations that are most similar. The similarity is measured using the distance between the circle centers and the angle between the normal vectors of the intersecting planes.

### 3.2 Circle Matching and Quality Measure

Having estimated the 3D pose of the different circles is the same as estimating the 5 DoF for the Stator Housings. Before communicating these results to the robot we also have to calculate a quality measure for each circle, i.e., which object should the robot pick first. A high quality object is an object which is not occluded, which is rotated so that the opening is maximum in the direction of the robot, and which is one of the top objects in the bin. The occlusion is measured using Chamfer matching, i.e., we synthesize the estimated pose of the object into the image and count the distance from each projected point to the nearest edge pixel. To avoid the influence of the actual distance of the projected object, the measure is normalized. The second measure is simply expressed as the *cosine* of the angle between the normal vector for the circle and the camera (both in the left camera). The third measure is the ratio between the distance from the circle to the camera and the distance from the camera to the circle closest to the camera. In mathematical terms the quality measure for the  $i'th$  circle is

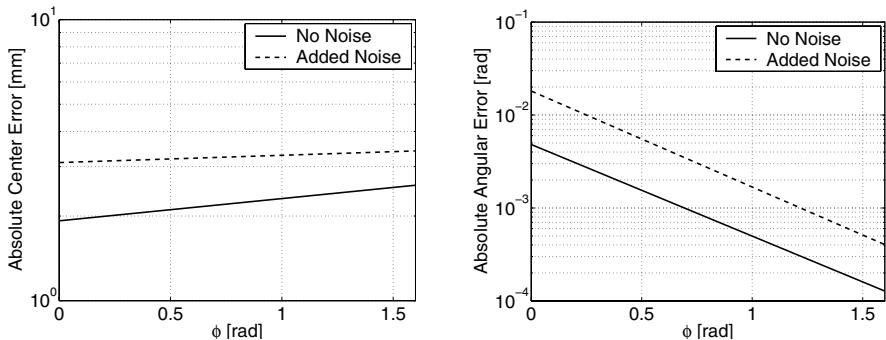
$$q(i) = w_1 \frac{M_i}{\max \left\{ M_i, \sum_j \varepsilon_j \right\}} + w_2 \cos(\varphi_i) + w_3 \frac{\delta_i}{\max \{\delta_j\}} \quad (3)$$

where  $w_1$ ,  $w_2$ , and  $w_3$  are weight factors,  $M$  is the number of projected points,  $\varepsilon_j$  is the distance from the  $j'th$  projected point to the nearest edge pixel in the image,  $\varphi_i$  is the angle between the normal vector of the circle and the camera,

and  $\delta_j$  is the distance from the center of the  $j'th$  circle to the camera. Note that each term in the quality measure is normalized to the interval  $[0; 1]$ .

## 4 Results

The first test is done on synthetic data in order to clarify how the angle between the simulated circle normal vector and the view point vector (denoted  $\phi$  in the following) affects the estimated pose. The test is based on 10.000 random circles with realistic parameters. Each circle is projected into the two images and the corresponding 3D circle is estimated. For each reconstructed circle we measure 1) the absolute error between the simulated circle and the estimated circle center and 2) the angle between the simulated circle normal vector and the estimated circle normal vector. Both measures are calculated with or without noise (each pixel is translated with a random value in the range  $[-3; 3]$  in both x and y directions), see figure 5.



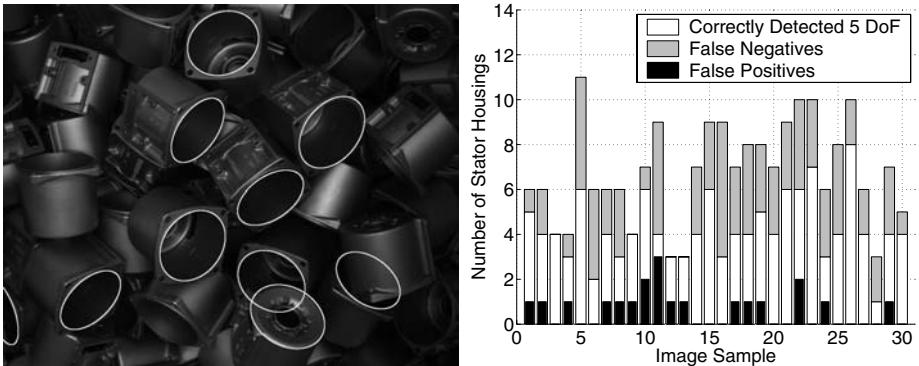
**Fig. 5.** Errors in the system based on 10.000 randomly generated circles. See text for further details. Note that each data set is fitted to an exponential curve, i.e., a line in a semilogarithmic coordinate system

When no noise is added the errors between the simulated and estimated centers are around 2 [mm]. In the case of added noise, however, the error increases with view point angle. This phenomenon can be explained by the type of noise introduced. As the circle is seen from an oblique angle (view point angle approaching  $\pi/2$ ) the projected ellipse will contain fewer pixels. As the noise are introduced on a pixel level, the particular noise will be most effective when the viewing angle increases.

The circle orientation tests show a tendency towards smaller errors as the view angle increased. This somehow non-intuitive property has been further investigated and the result has been supported by an analytic sensitivity analysis [5].

The second test is a quantitative test where 30 images like the one in figure 6 are used for each camera. We manually judged which circles the system should be able to find (based on visibility in both images) and used this as ground truth. In figure 6 (right) the height of each column is the ground truth. Furthermore the figure also illustrates the number of false positive and the number of false negatives. The false positives are mainly due to the fact that the opposite side of a Stator Housing contains a similar circle. The false negatives are mainly a result of corrupted edgels due to the illumination or incorrect splitting and merging of edgels.

Recall that we are only interesting in picking one object per time instance, i.e., after the object is removed a new image of the bin might provide a new set of ellipses. Therefore our success criterion is not a good recognition rate in general, but a good recognition rate among the Stator Housings with the best quality measures. For the 30 test examples the objects with the two or three highest quality measures are virtually always pose estimated correct.



**Fig. 6. Left:** Estimated 3D circles projected into the left camera image. **Right:** Quantitative test results for the 30 test examples

## 5 Conclusion

A general solution to the bin-picking problem has a huge potential in industry. The problem is, however, very difficult and therefore we have reformulated it as a matter of picking one and only one object from the bin and .. . finding the pose of this isolated object. The latter task is doable using standard techniques. The former is addressed by the notion of different algorithms and picking tools for each "side" of the object. In this paper we have presented a general solution to pose estimating objects containing circular openings which is a typical characteristic for objects manufactured at Grundfos. The presented algorithm can estimate the poses for a large variety of orientations and handle noise and

occlusions primary due to a pose-processing step where information from multiple edge segments in two different camera images are combined. Tests show the approach to be a solid solution to this problem.

Future work includes merging this algorithm with algorithms developed for the other sides of the object and then combining the quality measure for the different methods into one unifying scheme allowing the robot to pick the "best" object at any particular time instance.

## References

1. I. Balslev and R.D. Eriksen. From belt picking to bin picking. *Proceedings of SPIE - The International Society for Optical Engineering*, 4902:616–623, 2002.
2. M. Berger, G. Bachler, and S. Scherer. Vision Guided Bin Picking and Mounting in a Flexible Assembly Cell. In *13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE2000*, New Orleans, Louisiana, USA, June 2000.
3. O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. The MIT Press, first edition.
4. A.W. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
5. J. Kirkegaard. Pose estimation of randomly organised stator housings with circular features. Technical report, Aalborg University, Laboratory of Computer Vision and Media Technology, 2005.
6. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
7. D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.
8. T.B. Moeslund, M. Aagaard, and D. Lerche. 3D Pose Estimation of Cactus Leaves using an Active Shape Model. In *IEEE Workshop on Applications of Computer Vision (WACV)*, Breckenridge, Colorado, Jan 2005.
9. Y. Motai and A. Kosaka. Concatenate feature extraction for robust 3d elliptic object localization. *Applied Computing 2004 - Proceedings of the 2004 ACM Symposium on Applied Computing*, 1:21–28, 2004.
10. A.R. Pope. Learning to recognize objects in images: Acquiring and using probabilistic models of appearance. Technical report, The University of British Columbia, Department of Computer Science, 1995.
11. P.L. Rosin. Nonparametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1140–1153, 1995.
12. R. Safaei-Rad, I. Tchoukanov, K. Carless Smith, and B. Benhabib. Three-dimensional location estimation of circular features for machine vision. *IEEE Transactions on Robotics and Automation*, 8(5):624–640, 1992.
13. C. Torras. *Computer Vision - Theory and Industrial Applications*. Springer-Verlag, first edition, 1992.
14. E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice Hall, first edition, 1998.
15. [www.grundfos.com](http://www.grundfos.com).

# 3D Reconstruction of Metallic Surfaces by Photopolarimetric Analysis

P. d'Angelo and C. Wöhler

DaimlerChrysler Research and Technology, Machine Perception  
P. O. Box 2360, D-89013 Ulm, Germany  
{pablo.d\_angelo, christian.woehler}@daimlerchrysler.com

**Abstract.** In this paper we present a novel image-based 3D surface reconstruction technique that incorporates both reflectance and polarisation features into a variational framework. Our technique is especially suited for the difficult task of 3D reconstruction of rough metallic surfaces. An error functional consisting of several error terms related to the measured reflectance and polarisation properties is minimised in order to obtain a 3D reconstruction of the surface. We show that the combined approach strongly increases the accuracy of the surface reconstruction result, compared to techniques based on either reflectance or polarisation alone. We evaluate the algorithm based on synthetic ground truth data. Furthermore, we report 3D reconstruction results for a raw forged iron surface, thus showing the applicability of our method in real-world scenarios, here in the domain of quality inspection in the automotive industry.

1 Introduction

A well-known passive image-based surface reconstruction method is ... . This approach aims at deriving the orientation of the surface at each pixel by using a model of the reflectance properties of the surface and knowledge about the illumination conditions – for a detailed survey, cf. e. g. [2]. The integration of shadow information into the shape from shading formalism is described in detail in [8].

A further approach to reveal the 3D shape of a surface is to make use of polarisation data. For smooth dielectric surfaces, the direction and degree of polarisation as a function of surface orientation are governed by elementary physical

laws as described in detail e. g. in [5]. Information about the polarisation state of light reflected from the surface is utilised in [5] to reconstruct the 3D shape of transparent objects, involving multiple light sources equally distributed over a hemisphere and a large number of images acquired through a linear polarisation filter at different orientations. Reflectance and polarisation properties of metallic surfaces are examined in [9], but no physically motivated polarisation model is derived. In [7] polarisation information is used to determine the orientation of a surface. Applications of such approaches to real-world scenarios, however, are rarely described in the literature.

In this paper we present an image-based method for 3D surface reconstruction based on the simultaneous evaluation of information about reflectance and polarisation. The corresponding properties of the surface material are obtained by means of a series of images acquired through a linear polarisation filter under different orientations. Both reflectance and polarisation features are integrated into a unified variational framework. The method is systematically evaluated based on a synthetically generated surface to obtain information about its accuracy and is applied to the real-world example of a raw forged iron surface.

## 2 Combination of Reflectance and Polarisation Features for 3D Surface Reconstruction

In our scenario, we will assume that the surface  $z(x, y)$  to be reconstructed is illuminated by a point light source and viewed by a camera, both situated at infinite distance in the directions  $\mathbf{s}$  and  $\mathbf{v}$ , respectively. Parallel incident light and an orthographic projection model can thus be assumed. For each pixel location  $(u, v)$  of the image we intend to derive a depth value  $z(u, v)$ . The surface normal is given by the vector  $\mathbf{n} = (-p, -q, 1)^T$  with  $p = \partial z / \partial x$  and  $q = \partial z / \partial y$ . The angle  $\theta_i$  is defined as the angle between  $\mathbf{n}$  and  $\mathbf{s}$ , the angle  $\theta_e$  as the angle between  $\mathbf{n}$  and  $\mathbf{v}$ , and the angle  $\alpha$  as the angle between  $\mathbf{s}$  and  $\mathbf{v}$ . The observed pixel intensity is denoted by  $I(u, v)$ , the corresponding modelled intensity by  $R(\mathbf{n}, \mathbf{s}, \mathbf{v})$ . The 3D surface reconstruction formalism we utilise throughout this paper is related to the shape from shading scheme described in detail in [2, 3, 4]. It relies on the global minimisation of an error function that consists of a weighted sum of several individual error terms. One part of this error function is the term

$$e_I = \sum_{u,v} [I(u, v) - R(\mathbf{n}, \mathbf{s}, \mathbf{v})]^2. \quad (1)$$

A regularization constraint  $e_s$  is introduced which requires local continuity of the surface. A smooth surface implies small absolute values of the directional derivatives of the surface gradients. In this paper we will therefore make use of the error term (cf. also [2, 4])

$$e_s = \sum_{u,v} [p_x^2 + p_y^2 + q_x^2 + q_y^2]. \quad (2)$$

Alternatively, one might replace (2) by the error term described in detail in [3].

In this paper we regard metallic surfaces with a strongly non-Lambertian reflectance function  $R(\mathbf{n}, \mathbf{s}, \mathbf{v})$ . We can assume, however, that the surface interacts with the incident light in an isotropic manner in the sense that the reflectance function exclusively depends on  $\theta_i$ ,  $\theta_e$ , and  $\alpha$ . It will be determined by means of a suitable measurement procedure (cf. Section 3).

In our scenario, the incident light is unpolarised. For smooth metallic surfaces the light remains unpolarised after reflection at the surface. Rough metallic surfaces, however, cause the reflected light to be partially polarised [9]. When observed through a linear polarisation filter, the reflected light will have a transmitted radiance that oscillates sinusoidally as a function of the orientation of the polarisation filter between a maximum  $I_{\max}$  and a minimum  $I_{\min}$ . The

$\Phi \in [0^\circ, 180^\circ]$  denotes the orientation under which the maximum transmitted radiance  $I_{\max}$  is observed. The is defined by  $D = (I_{\max} - I_{\min})/(I_{\max} + I_{\min}) \in [0, 1]$ . As no physical model exists so far which is able to accurately describe the polarisation properties of rough metallic surfaces, the functions  $R_\Phi(\mathbf{n}, \mathbf{s}, \mathbf{v})$  and  $R_D(\mathbf{n}, \mathbf{s}, \mathbf{v})$  describing polarisation angle and degree of the material, respectively, have to be determined empirically (cf. Section 3).

To integrate polarisation features into the 3D surface reconstruction framework, we define the error terms

$$e_\Phi = \sum_{u,v} [\Phi(u, v) - R_\Phi(\mathbf{n}, \mathbf{s}, \mathbf{v})]^2 \quad (3)$$

and

$$e_D = \sum_{u,v} [D(u, v) - R_D(\mathbf{n}, \mathbf{s}, \mathbf{v})]^2 \quad (4)$$

which denote the mean square deviation between observed and modelled polarisation angle and degree, respectively. To obtain surface gradients  $p(u, v)$  and  $q(u, v)$  that optimally fit the observed reflectance and polarisation properties, the overall error

$$e = e_s + \lambda e_I + \mu e_\Phi + \nu e_D \quad (5)$$

has to be minimized. The Lagrange parameters  $\lambda$ ,  $\mu$ , and  $\nu$  denote the relative weights of the error terms. We obtain an iterative update rule for  $p(u, v)$  and  $q(u, v)$  by setting the derivatives of  $e$  with respect to  $p(u, v)$  and  $q(u, v)$  to zero (see also [4]):

$$p_{n+1} = \bar{p}_n + \lambda (I - R) \frac{\partial R}{\partial p} + \mu (\Phi - R_\Phi) \frac{\partial R_\Phi}{\partial p} + \nu (D - R_D) \frac{\partial R_D}{\partial p} \quad (6)$$

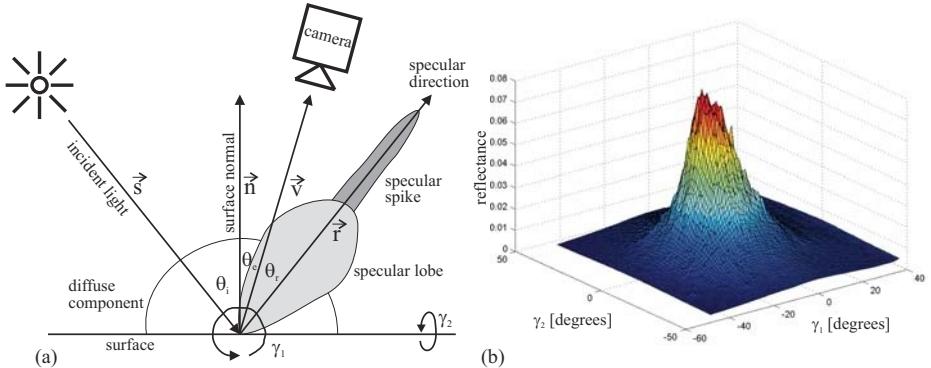
with  $\bar{p}_n$  as a local average. An analogous expression is obtained for  $q$ . The initial values  $p_0(u, v)$  and  $q_0(u, v)$  must be provided based on a-priori knowledge about the surface (cf. Section 4). Numerical integration of the gradient field, employing e. g. the algorithm described in [4], yields the surface profile  $z(u, v)$ . The partial derivatives are computed at  $(\bar{p}_n, \bar{q}_n)$ , respectively.

### 3 Measurement of Photopolarimetric Image Data

This section explains how the reflectance and polarisation properties of the surface are measured and described in terms of suitable analytical functions for further processing.

#### 3.1 Measurement of Reflectance Properties

The reflectance function of a typical rough metallic surface consists of three components: a diffuse (Lambertian) component, the specular lobe, and the specular spike [6]. The diffuse component is generated by internal multiple scattering processes. The specular lobe, which is caused by single reflection at the surface, is distributed around the specular direction and may be rather broad. The specular spike is concentrated in a small region around the specular direction and represents mirror-like reflection, which is dominant in the case of smooth surfaces. For an illustration, see Fig. 1a.



**Fig. 1.** (a) Plot of the three components of the reflectance function (cf. also [6]). (b) Measured reflectance of a raw forged iron surface. In the least mean squares fit of (7) to these measurements, the specular lobe is described by  $\sigma_1 = 4.49$  and  $m_1 = 6.0$ , the specular spike by  $\sigma_2 = 1.59$  and  $m_2 = 57.3$

With  $\theta_i$  as the incidence angle and  $\theta_r$  as the angle between the specular direction  $\mathbf{r}$  and the viewing direction  $\mathbf{v}$  (cf. Fig. 1a), which can be expressed as  $\cos \theta_r = 2 \cos \theta_i \cos \theta_e - \cos \alpha$ , we define an analytical form for the reflectance function:

$$R(\theta_i, \theta_e, \alpha) = \rho \left[ \cos \theta_i + \sum_{n=1}^N \sigma_n \cdot (2 \cos \theta_i \cos \theta_e - \cos \alpha)^{m_n} \right]. \quad (7)$$

For  $\theta_r > 90^\circ$ , only the diffuse component is considered. The albedo  $\rho$  is assumed to be constant over the surface. We set  $N = 2$  to model both the specular lobe and the specular spike. The coefficients  $\{\sigma_n\}$  denote the strength of the specular components relative to the diffuse component, while the parameters  $\{m_n\}$  denote their widths.

To obtain reflectance measurements of a surface, a sample part is attached to a goniometer, which allows for a rotation of the sample around two orthogonal axes. The corresponding goniometer angles  $\gamma_1$  and  $\gamma_2$  can be adjusted at an accuracy of a few arcseconds. As illustrated in Fig. 1a, adjusting  $\gamma_1$  is equivalent to rotating the surface normal  $\mathbf{n}$  around an axis perpendicular to the plane spanned by the vectors  $\mathbf{s}$  and  $\mathbf{r}$ , while adjusting  $\gamma_2$  causes a rotation of  $\mathbf{n}$  around an axis lying in that plane. The phase angle  $\alpha$ , given by the relative position of the light source and the camera, is assumed to be constant over the image. It is straightforward to determine the surface normal  $\mathbf{n}$ , the incidence angle  $\theta_i$ , and the emission angle  $\theta_e$  from the goniometer angles  $\gamma_1$  and  $\gamma_2$  and the vectors  $\mathbf{s}$  and  $\mathbf{v}$ .

The average greyvalue over an image area containing a flat part of the sample surface is regarded as the reflectance (in arbitrary units) under the given illumination conditions, respectively. A typical reflectance measurement is shown in Fig. 1b.

### 3.2 Measurement of Polarisation Properties

The measurement of the polarisation properties of the surface is similar to the reflectance measurement. For each configuration of goniometer angles, five images are acquired through a linear polarisation filter at an orientation angle  $\omega$  of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ , and  $180^\circ$ . For each filter orientation  $\omega$ , an average pixel intensity over an image area containing a flat part of the sample surface is computed as described in Section 3.1. A function of the form

$$I(\omega) = I_c + I_v \cos(\omega - \Phi) \quad (8)$$

is then fitted to the measured pixel intensities. The filter orientation  $\Phi$  for which maximum intensity is observed corresponds to the polarisation angle defined in Section 2. The polarisation degree now becomes  $D = I_v/I_c$ . In principle, three measurements would be sufficient to determine the three parameters  $I_c$ ,  $I_v$ , and  $\Phi$ , but the fit becomes less noise-sensitive when more measurements are used. The value  $I_c$  represents the reflectance of the surface.

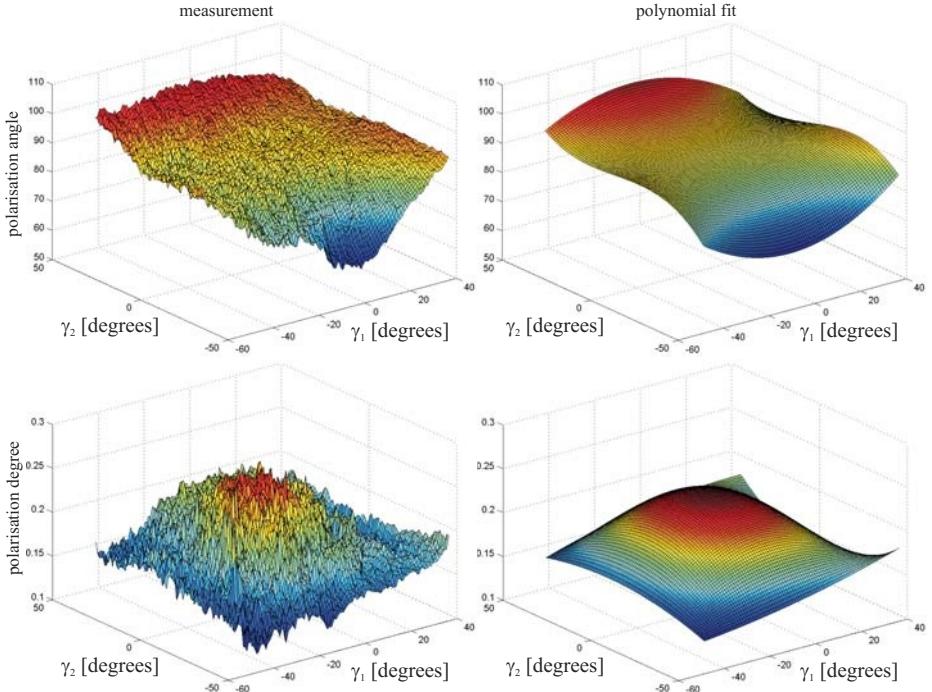
As no accurate physically motivated model for the polarisation properties of rough metallic surfaces is available so far, we perform a polynomial fit in terms of the goniometer angles  $\gamma_1$  and  $\gamma_2$  to the measured polarisation angle and degree. The polarisation angle is represented by an incomplete third-degree polynomial of the form

$$R_\Phi(\gamma_1, \gamma_2) = a_\Phi + b_\Phi \gamma_2 + c_\Phi \gamma_1 \gamma_2 + d_\Phi \gamma_1^2 \gamma_2 + e_\Phi \gamma_2^3, \quad (9)$$

which is antisymmetric in  $\gamma_2$ , and  $R_\Phi(\gamma_1, 0) = a_\Phi = \text{const}$ , corresponding to coplanar vectors  $\mathbf{n}$ ,  $\mathbf{s}$ , and  $\mathbf{v}$ . In an analogous manner, the polarisation degree is represented by an incomplete polynomial of the form

$$R_D(\gamma_1, \gamma_2) = a_D + b_D \gamma_1 + c_D \gamma_1^2 + d_D \gamma_2^2 + e_D \gamma_1^2 \gamma_2^2, \quad (10)$$

which is symmetric in  $\gamma_2$ . The symmetry properties are required for geometrical reasons as long as an isotropic interaction between incident light and surface



**Fig. 2.** Measured and modelled polarisation properties of a raw forged iron surface. Top: polarisation angle. Bottom: polarisation degree

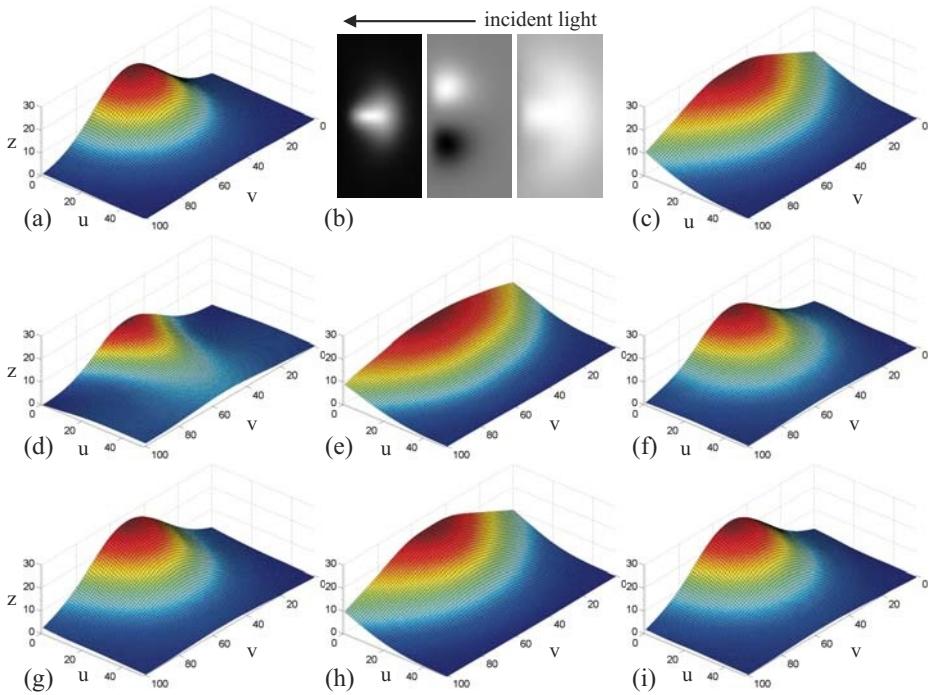
material can be assumed. The polarisation properties of a raw forged iron surface measured at a phase angle of  $\alpha = 79^\circ$  are illustrated in Fig. 2, along with the polynomial fits according to (9) and (10).

## 4 Experimental Results

### 4.1 Evaluation Based on Synthetic Ground Truth Data

To examine the quality of 3D reconstruction, dependent on which reflectance and polarisation features are used, we apply the algorithm described in Section 2 to the synthetically generated surface shown in Fig. 3a. We assume a perpendicular view on the surface along the  $z$  axis. The scene is illuminated in image  $u$  direction under an angle of  $11^\circ$  with respect to the horizontal plane, resulting in a phase angle of  $\alpha = 79^\circ$ . The surface albedo  $\rho$  is computed based on the specular reflections, which appear as regions of maximum intensity  $I_{\text{spec}}$  and for which we have  $\theta_r = 0^\circ$  and  $\theta_i = \theta_e = \alpha/2$ , such that from (7) we obtain

$$\rho = I_{\text{spec}} \cdot \left[ \cos(\alpha/2) + \sum_{n=1}^N \sigma_n \right]^{-1}. \quad (11)$$



**Fig. 3.** 3D reconstruction of a synthetically generated surface. (a) Ground truth. (b) From the left: Reflectance image, polarisation angle image, polarisation degree image. The 3D reconstruction result is obtained based on (c) reflectance, (d) polarisation angle, (e) polarisation degree, (f) polarisation angle and degree, (g) reflectance and polarisation angle, (h) reflectance and polarisation degree, (i) reflectance, polarisation angle and degree. The reconstruction results have been obtained based on noise-free images

The initial values for  $p(u, v)$  and  $q(u, v)$  must be provided relying on approximate knowledge about the surface orientation. In the synthetic surface example,  $p(u, v)$  and  $q(u, v)$  are initialised with zero values.

The reflectance, polarisation angle, and polarisation degree images shown in Fig. 3b have been generated by means of the polynomial fits to the measured reflectance and polarisation properties presented in Figs. 1 and 2. The weights for the corresponding error terms according to (5) are set to  $\lambda = 0.0075$  ( $I$  in arbitrary units, with a maximum of the order 0.1),  $\mu = 4.0$  ( $\Phi$  in radian), and  $\nu = 0.8$  ( $D$  between 0 and 1). When an error term is neglected, the corresponding weight is set to zero. The 3D reconstruction results obtained based on all possible combinations of reflectance and polarisation features are shown in Fig. 3c-i. The corresponding RMS deviations from the ground truth for  $z$ ,  $p$ , and  $q$  are given in Table 1. The best 3D reconstruction result is obtained when all three features are used, while the second-best result is achieved for a combination of reflectance and polarisation angle. According to Table 1, the performance of our method hardly decreases when noise is added to the images.

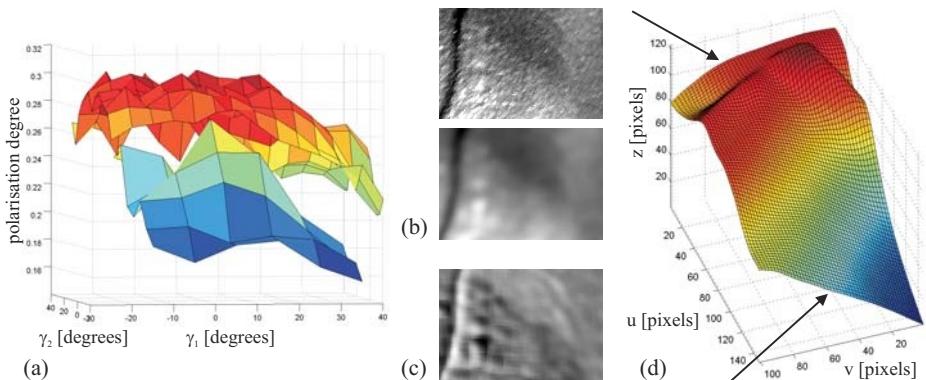
**Table 1.** Results of evaluation on synthetic ground truth data. Noise level is  $\pm 0.5$  for the reflectance (maximum value 6.74),  $\pm 4^\circ$  for the polarisation angle and  $\pm 0.05$  for the polarisation degree. The noise is uniformly distributed over the corresponding intervals. The residual error in (g) and (i) in the noise-free case is primarily due to the assumption of a smooth surface (2)

Features	Figure 3	RMS error (without noise)			RMS error (with noise)		
		<i>z</i>	<i>p</i>	<i>q</i>	<i>z</i>	<i>p</i>	<i>q</i>
Reflectance	(c)	2.95	0.072	0.331	2.95	0.075	0.331
Pol. angle	(d)	4.24	0.177	0.088	4.06	0.165	0.097
Pol. degree	(e)	3.50	0.131	0.330	3.47	0.130	0.330
Pol. angle and degree	(f)	3.87	0.076	0.039	4.48	0.077	0.059
Reflectance and pol. angle	(g)	0.45	0.030	0.047	0.70	0.044	0.062
Reflectance and pol. degree	(h)	2.85	0.074	0.326	2.85	0.076	0.327
Reflectance and polarisation (angle and degree)	(i)	0.37	0.017	0.034	0.69	0.035	0.056

Reflectance  $R$  and polarisation degree  $R_D$  appear to contain somewhat redundant information, as their combination hardly reduces the reconstruction error, compared to the results obtained by means of one the features alone. Both  $R$  and  $R_D$  display a maximum in the specular direction ( $\theta_r = 0^\circ$ ) and decrease in a similar manner for  $\theta_r > 0^\circ$ . The dependence on surface orientation, however, is much lower for  $R_D$  than for  $R$ , while the measurement error tends to be significantly smaller for the reflectance. In practical applications (cf. Section 4.2) it turns out that 3D surface reconstruction should preferably be performed based on a combination of reflectance and polarisation angle, neglecting the polarisation degree.

## 4.2 Application to a Rough Metallic Surface

In this section, we will describe the application of our 3D surface reconstruction method to the raw forged iron surface of an automotive part. For such a kind of surface, the polarisation degree for specular reflection tends to vary over the surface by up to 20 percent, depending on the locally variable microscopic roughness, in a rather erratic manner (cf. Fig. 4a). In contrast, the behaviour of the polarisation angle turns out to be very stable over the surface. The polarisation degree is thus an unreliable feature, such that we perform a 3D reconstruction of the forged iron surface based on a combination of reflectance and polarisation angle. For each pixel, polarisation angle and degree are determined as described in Section 3. The surface albedo is obtained according to (11). As the microscopic surface roughness produces a strong scatter in the measured polarisation data, the images are blurred before the polarisation properties are computed. This is permitted because we desire to obtain information about the surface shape on scales significantly larger than the microscopic roughness.



**Fig. 4.** Application of the described 3D surface reconstruction method to a raw forged iron surface. (a) Polarisation degree, measured at two different locations of the same industrial part consisting of forged iron. (b) Reflectance image, original (above) and blurred (below). The greyvalues are scaled logarithmically. (c) Polarisation angle image. (d) Reconstructed 3D profile. Due to a fault caused during the forging process, the cross-section of the surface significantly changes its shape over the image (arrows)

The 3D reconstruction along with the reflectance and polarisation images is shown in Fig. 4b-d. At the left image border, the cross-section of the surface is roughly cylindrical, while it is flat at the right image border (arrows). The ground truth, i. e. independently derived accurate  $z(u, v)$  quantities, is not available for this surface, but the real surface shows a very similar asymmetric behaviour. A flawless part would display the cylindrical cross-section (upper arrow) over the complete surface region shown in Fig. 4d. The observed asymmetric shape is due to a fault caused during the forging process. This anomaly is hardly visible in the reflectance image (Fig. 4b) but is revealed by additional evaluation of polarisation information. The ridge at  $u \approx 20$ , running vertically in Fig. 4b, is well apparent in the reconstructed profile.

## 5 Summary and Conclusion

In this paper we have presented an image-based method for 3D surface reconstruction based on the simultaneous evaluation of reflectance and polarisation information. The reflectance and polarisation properties of the surface material have been obtained by means of a series of images acquired through a linear polarisation filter under different orientations. Analytical phenomenological models are fitted to the measurements, allowing for an integration of both reflectance and polarisation features into a unified variational framework. The method has been evaluated based on a synthetically generated surface, illustrating that for a single view and a single light source, both reflectance and polarisation features are necessary to fully recover the surface shape. We have furthermore applied the method to the 3D reconstruction of a raw forged iron surface, showing that the approach is suitable for detecting anomalies of the surface shape.

The approach described in this paper relies on a single view of the surface, illuminated by a single light source. Concerning future work, we expect that a less ambiguous and more accurate reconstruction result can be obtained by taking into account multiple views and light sources, which will involve an approach based on photopolarimetric stereo.

## References

1. J. Batlle, E. Mouaddib, J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern Recognition*, vol. 31, no. 7, pp. 963-982, 1998.
2. B. K. P. Horn, M. J. Brooks. Shape from Shading. MIT Press, Cambridge, Massachusetts, 1989.
3. B. K. P. Horn. Height and Gradient from Shading. MIT technical report 1105A. <http://people.csail.mit.edu/bkph/AIM/1105A-TEX.pdf>
4. X. Jiang, H. Bunke. Dreidimensionales Computersehen. Springer-Verlag, Berlin, 1997.
5. D. Miyazaki, M. Kagesawa, K. Ikeuchi. Transparent Surface Modeling from a Pair of Polarization Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 73-82, 2004.
6. S. K. Nayar, K. Ikeuchi, T. Kanade. Surface Reflection: Physical and Geometrical Perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 611-634, 1991.
7. S. Rahmann, N. Canterakis. Reconstruction of Specular Surfaces using Polarization Imaging. *Int. Conf. on Computer Vision and Pattern Recognition*, vol. I, pp. 149-155, Kauai, USA, 2001.
8. C. Wöhler, K. Hafezi. A general framework for three-dimensional surface reconstruction by self-consistent fusion of shading and shadow features. *Pattern Recognition*, vol. 38, no. 7, pp. 965-983, 2005.
9. L. B. Wolff. Constraining Object Features Using a Polarization Reflectance Model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 635-657, 1991.
10. L. B. Wolff. Polarization vision: a new sensory approach to image understanding. *Image and Vision Computing*, vol. 15, pp. 81-93, 1997.

# Inferring and Enforcing Geometrical Constraints on a 3D Model for Building Reconstruction

Franck Taillandier

Institut Géographique National 2-4 avenue Pasteur, 94160 Saint-Mandé, France  
franck.taillandier@ign.fr

**Abstract.** We present a method for inferring and enforcing geometrical constraints on an approximate 3D model for building reconstruction applications. Compared to previous works, this approach requires no user intervention for constraints definition, is inherently compliant with the detected constraints and handles over-constrained configurations. An iterative minimization is applied to search for the model *subject to geometric constraints* that minimizes the distance to the initial approximate model. Results on real data show the gain obtained by this algorithm.

## 1 Introduction

### 1.1 Motivations

Reconstruction of man-made objects or environments from multiple images is of primary importance in many fields such as Computer Aided Design, architectural design, building reconstruction. Taking into account some geometrical properties can greatly improve the quality of reconstruction [1], is essential to reduce the degree of freedom of the problem and to provide a good modelling of the environment.

The application described in this article is part of a global project [2] aiming at automatic building reconstruction using plane primitives as base primitives and a modelling of building as polyhedral models without overhang. Integration of constraints in this scheme enables to reduce the combinatorial while taking into account the specificity of man-made environment. The part described here focuses on building reconstruction *once topology is known* by integrating constraints directly inferred on the model.

### 1.2 Related Work

In building reconstruction applications and more generally in architectural modelling, methods used to reconstruct man-made environments can be classified as “model-based” and “constraints-based”. In the former ones, objects are decomposed into known base primitives (parallelepiped, rectangles...) grouped together to build more complex shapes [3, 4, 5, 6, 7, 8]. To identify or to validate building parts, matching is performed between a model and 3D or 2D primitives through reprojection of hypotheses on image gradients or corners [4, 5], perceptual or model-based grouping and model verification [6, 7, 8]). Constraints are a priori known in the model and must be enforced on

the matched primitives for reconstruction purposes, which is a difficult task because redundant constraints and over-constraints are not handled in the geometrical definition of the model. The latter methods are more general. They only rely on some geometrical properties very common in urban or man-made environments, without a priori models, to reconstruct any polyhedral shape. Several works take advantage of these properties [1, 9, 10, 11, 12], but due to the complexity of the problem, they all rely on user intervention for topological *and* geometrical description of models, which is very time-consuming and often prohibitory for dense and large urban areas reconstruction.

As for the techniques used to enforce constraints on an approximate 3D model, geometrical conditions are expressed in two different ways. Some embed the conditions in the Least Square minimization process [11, 12, 1] thus leading to a solution where it is hard to evaluate whether constraints are actually verified or not. Other perform an implicit modelling of the scene before the minimization process [10, 9]. It has the advantage of inherently being compliant with the provided constraints. However, for these systems, although methods are available to detect over-constrained situations, no automatic solution is provided.

As for the constraints handled by the authors, all of them use planarity information to enforce topology constraints that intrinsically result from object description (points belong to the same facet and thus to the same plane). [10, 9] use also purely geometrical constraints that arise between object features: orthogonality, parallelism...

### 1.3 General Scheme

The algorithm presented in this article extends the model developed in [9] to avoid the drawbacks recalled in section 1.2. First, constraints are inferred automatically on the initial shape. *No user intervention is needed*. Then constraints are enforced on the model and *degeneracies or over-constraints are handled* by iteratively suppressing some constraints. The application presented is dedicated to automatic building reconstruction therefore we focus on constraints recurrent in these environment: parallelisms, orthogonality, horizontal edges, vertical symmetry.

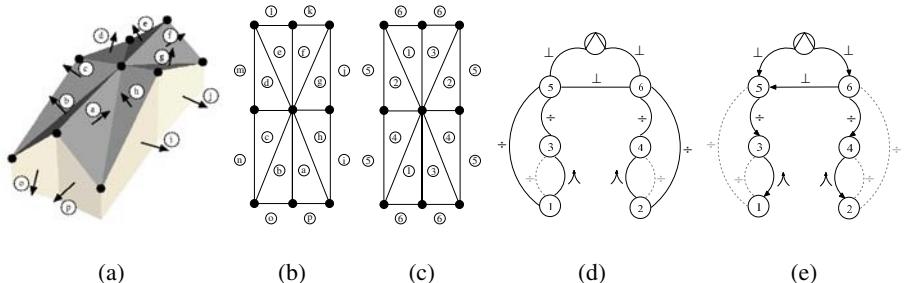
The algorithm assumes an initial approximate 3D model whose topological relations are known. In our example, this model may come from automatic algorithms [2]. In a first step, normal vectors of the different facets are clustered (section 2.2), enabling to handle parallelism properties and horizontal planes. A constraints graph is then deduced from these clusters, integrating all the geometrical constraints that will be applied on the model: orthogonality, horizontal edges, symmetries (sections 2.3 and 2.4). In a second step, a set of equations is built from planarity equations between points and normal vectors. This implicit parameterization *ensures that all constraints are verified* (section 3.1). Through iterative minimization, one searches then for the constrained model that minimizes the distance to the initial approximate model (section 3.2). Over-constraints situations are detected and handled by iteratively suppressing some constraints until convergence (section 4) A qualitative and quantitative evaluation assess the validity of the approach controlled by two parameters  $\sigma_{\parallel}$  and  $\sigma_{\perp}$  that are easily chosen according to the noise present in the approximate model (section 5).

## 2 Inferring Constraints

### 2.1 Topology and Geometry: Notations

The application depicted here aims at building roof geometrical reconstruction. Facades making up the border of buildings are inferred from the roofs definition. They will be of primary importance to impose constraints on the external edges of buildings. A facet is added to the initial set of facets for each external edge of the building, therefore what will be called a facet is not necessarily a closed polygon but may be constituted of only two points.

In the following, for the initial model whose topological relations are known,  $\mathcal{F}$  is the set of facets whose cardinal is  $F$ . From the topological point of view, each facet  $f$  has  $|f|$  points and from a geometrical point of view, it is linked to a normal vector or direction  $v_i$ . The set of directions  $\mathcal{D}$  has cardinality  $D$  (initially  $D = F$ ). Figure 1(a) shows an example of a 3D building and its schematic representation.



**Fig. 1.** Approximate 3D building (a) and its schematic representation with directions (letters) annotated on facets (b) where black circles are initial approximate points that do not verify constraints and vertical planes making up the border are inferred from them. (c) shows the result of clustering. (d) and (e) depict constraints graph and dependences graph respectively. Each arc represents a constraint between two directions.  $\oslash$  defines the vertical direction,  $\perp$ ,  $\div$  and  $\wedge$  symbolize orthogonality, horizontal edge and vertical symmetry constraints respectively. Grey arcs are constraints suppressed from the graph (see text for details). The sequence of directions of this dependences graph is  $(\oslash, 5, 4, 2, 6, 3, 1)$  and renumbering can be performed

### 2.2 Directions Clustering

From the initial model, all normal vectors are estimated for each facet of the model, thus leading to  $F$  normal vectors hereafter called directions  $v$ . To these directions, in the case of building reconstruction, the vertical normal is added because of its obvious importance in man-made scene. In order to handle parallelism and horizontal properties, one has to identify directions nearly identical and directions close to the vertical direction. To achieve this, a pairwise centroid-linkage clustering is performed on the initial directions. A cluster is created by grouping the couple of normal vectors whose distance is the lowest. In this case, the distance is the absolute value of the angle between both

normal vectors. A new normal is recomputed for the new cluster by the mean value of the normals enclosed by the cluster. Iteratively, normals are grouped together and the process goes on until the minimum distance is above a threshold  $\sigma_{\parallel}$ . At the end of the process, each pair of clusters are distant of *more than*  $\sigma_{\parallel}$  and there is only  $D \leq F$  directions. Let us mention that, in the case of building reconstruction applications, in order not to alter the “vertical” direction, the value of the direction for its cluster is kept vertical and not averaged, thus enabling to handle facets considered as horizontal.

Each facet of the initial model is then identified with a direction, which will be useful in the following to build the constrained system.

### 2.3 Constraints Graph

Constraints between normal vectors have to be determined on the reduced set of directions. These constraints are represented as arcs in a so-called “constraints graph” (see figure 1(e)). In this unoriented graph, each direction is a node. Several types of valued arcs are added between two nodes  $\mathbf{v}_i$  and  $\mathbf{v}_j$  whenever a condition is met:

- *orthogonality* if  $|\mathbf{v}_i \bullet \mathbf{v}_j| > \cos(\sigma_{\triangleleft})$
- *vertical symmetry* if  $|(\mathbf{v}_i + \mathbf{v}_j) \bullet \mathbf{z}| > \cos(\sigma_{\triangleleft}) \cdot \|(\mathbf{v}_i + \mathbf{v}_j)\|$
- *horizontal line* if  $|\mathbf{v}_i \otimes \mathbf{v}_j| < \sin(\sigma_{\triangleleft}) \cdot \|\mathbf{v}_i \otimes \mathbf{v}_j\|$

where  $\bullet$  represents the dot product and  $\mathbf{z}$  is the vertical direction. Each arc  $c$  corresponds to a constraint between two directions  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , which will be noted  $c_f = \mathbf{v}_i$ ,  $c_t = \mathbf{v}_j$ . In the constraints graph, arc (or constraint) orientation has no importance (graph is un-oriented). In the following however, the previous notation will state that  $c$  is oriented from  $\mathbf{v}_i$  to  $\mathbf{v}_j$ , thus that  $\mathbf{v}_i$  *constraints*  $\mathbf{v}_j$ . Only one arc may exist between two nodes, therefore in case of redundancy between vertical symmetry and horizontal line, symmetry is preferred since it brings more constraints on directions. In the case of building reconstruction, orthogonality is only tested between vertical planes. Let us mention that, as  $\sigma_{\parallel}$ ,  $\sigma_{\triangleleft}$  is easily chosen according to the error estimated in the approximate model or the degree of caricature desired in the reconstruction.

### 2.4 Dependences Graph

One of the difficulties of methods performing an implicit parameterization of the model [10, 9] is that directions must be computed *sequentially* so that they can be deduced from some previous ones in the sequence and some constraint rules. This is equivalent to orient the constraints graph [10] while ensuring that no over-constraint occurs. Some attempts have been made to solve this complex problem in general configurations [13] but they are not well adapted to the 3D case. Another approach has been developed to handle general configurations (with constraints on points, planes, lines...) with procedures designed to maximize the number of constraints applied in the system [10]. We reuse some of the procedures to orient our constraint graph.

The algorithm explicitly builds the sequence of directions. Initially, all constraints in the constraints graph are considered inactive. The algorithm iteratively integrates all directions in the sequence and orients therefore the constraints (equivalent to arcs) of the constraints graph linked to them. Let us mention that each oriented arc reduces the

degree of freedom of the node it gets to by  $f(e)$ : 2 when it corresponds to vertical symmetry, 1 for any of the two other constraints. Initially, each node has a degree of freedom of 2 and the current degree of freedom can easily be computed at each step and for each node.

First the vertical direction (that has a peculiar significance in building reconstruction) is introduced in the sequence and all constraints linked to that node are made active and oriented from it. Then iteratively, the algorithm chooses in the most constrained directions one that has the highest number of potential (still non active) constraints related to it. It activates these constraints by orienting them from this node. If, at any point a direction gets over-constrained (its degree of freedom becomes negative), some constraints are suppressed until there is no more over-constraint.

At the end of this process, the “dependences graph” is the result of the orientation of the arcs in the constraints graph. Let us note that some constraints may disappear between the constraints graph and the dependences graph (see figure 1(e)) due to some possible cases of overconstraints during the process or because of redundancy. Once the dependences graph is built, it can be shown that for each direction, only 6 situations recalled in table 1 may occur. From the previous procedure, each direction may also be numbered so that they can be computed sequentially, each one being defined by the previous ones and some constraints applied on it, which are given by arcs of the dependences graph. In the following, we will assume that the directions have been renumbered and that  $\forall i$ ,  $\mathbf{v}_i$  depend only on previous direction(s)  $\mathbf{v}_j$ ,  $j < i$ . Whenever a direction  $\mathbf{v}_i$  is not completely constrained by previous directions in the sequence (for instance arbitrary directions that do not depend on any previous direction in the sequence or perpendicular directions that depend on only one other direction), one may assume that it depends on some parameters  $\theta_i$ . Some of the directions may not require any parameter since they are completely defined by other directions (or are known!). However, for simplicity, we will keep these useless  $\theta_i$  in our notation and consequently consider the set  $\theta_1 \dots \theta_D$ . Then rules recalled in the table 1 enable to sequentially compute each direction from these parameters and directions of dependences so that *geometrical constraints imposed to them are verified*. From this table, derivatives  $\partial \mathbf{v}_i / \partial \theta_k \quad \forall i, k$  can easily be deduced from given hints and chain rule. They will be of use in the iterative minimization.

### 3 Enforcing Constraints

#### 3.1 Geometrical Parametrization

This section mostly recalls principles of the method depicted in [14]. As explained above, each direction is computed from some parameters  $\theta = [\theta_1^T, \dots, \theta_D^T]^T$ . Each pair of points belong to the same plane whose normal direction is  $\mathbf{v}_i$  if and only if their coordinates  $\mathbf{X}_m, \mathbf{X}_n$  verify the planarity constraint:

$$\mathbf{v}_i^T (\mathbf{X}_m - \mathbf{X}_n) = 0 \quad (1)$$

For each facet of the initial model with  $|f|$  points,  $|f| - 1$  equations can thus be inferred. Concatenating all equations leads to the system  $\mathbf{B}(\theta_1, \theta_2, \dots, \theta_D)\mathbf{X} = \mathbf{O}$  where  $\mathbf{X} =$

**Table 1.** Computations of direction  $\mathbf{v}_i$  and its derivatives with respect to some parameters and according to relations constraining them.  $\mathbf{S}_v$  is the skew symmetric matrix for a vector  $\mathbf{v}$

Type	Parameter	Value	Derivatives
known	No	unchanged	0
arbitrary	$\theta_i \neq 0$	$\mathbf{v}_i = \frac{\theta_i}{\ \theta_i\ }$	$\frac{\partial \mathbf{v}_i}{\partial \theta_i} = \frac{1}{\ \theta_i\ } \left( I - \frac{1}{\ \theta_i\ ^2} \theta_i \theta_i^T \right)$
orthogonal to $\mathbf{v}_j$	$\theta_i$	$\mathbf{u}_1 = (I - \mathbf{v}_j \mathbf{v}_j^T) \theta_i$ $\mathbf{u}_2 = \frac{\mathbf{u}_1}{\ \mathbf{u}_1\ }, \mathbf{v}_i = \mathbf{u}_2$	$\frac{\partial \mathbf{v}_i}{\partial \theta_i} = \frac{1}{\ \mathbf{u}_1\ } (I - \mathbf{u}_2 \mathbf{u}_2^T - \mathbf{v}_j \mathbf{v}_j^T)$ $\frac{\partial \mathbf{v}_i}{\partial \mathbf{v}_j} = \frac{1}{\ \mathbf{u}_1\ } \left( (\mathbf{v}_j^T \theta_i) (I - \mathbf{u}_2 \mathbf{u}_2^T) + \mathbf{v}_j \theta_i^T \right)$
2 orthogonalities: $\mathbf{v}_j, \mathbf{v}_k$	No	$\mathbf{v}_i = \frac{\mathbf{v}_j \otimes \mathbf{v}_k}{\ \mathbf{v}_j \otimes \mathbf{v}_k\ }$	$\frac{\partial \mathbf{v}_i}{\partial \mathbf{v}_j} = -\frac{1}{\ \mathbf{v}_j \otimes \mathbf{v}_k\ } (I - \mathbf{v}_i \mathbf{v}_i^T) \mathbf{S}_{v_k}$ $\frac{\partial \mathbf{v}_i}{\partial \mathbf{v}_k} = \frac{1}{\ \mathbf{v}_j \otimes \mathbf{v}_k\ } (I - \mathbf{v}_i \mathbf{v}_i^T) \mathbf{S}_{v_j}$
horizontal line with $\mathbf{v}_j$	$\theta_i$	$\mathbf{h} = \mathbf{z} \otimes \mathbf{v}_j$ $\mathbf{u}_1 = (I - \mathbf{h} \mathbf{h}^T) \theta_i$ $\mathbf{u}_2 = \frac{\mathbf{u}_1}{\ \mathbf{u}_1\ }, \mathbf{v}_i = \mathbf{u}_2$	$\frac{\partial \mathbf{v}_i}{\partial \theta_i} = \frac{1}{\ \mathbf{u}_1\ } (I - \mathbf{u}_2 \mathbf{u}_2^T - \mathbf{h} \mathbf{h}^T)$ $\frac{\partial \mathbf{v}_i}{\partial \mathbf{v}_j} = \frac{1}{\ \mathbf{u}_1\ } \left( (\mathbf{h}^T \theta_i) (I - \mathbf{u}_2 \mathbf{u}_2^T) + \mathbf{h} \theta_i^T \right) \mathbf{S}_z$
vertical symmetry with $\mathbf{v}_j$	No	$\mathbf{v}_i = 2(\mathbf{v}_j \cdot \mathbf{z}) - \mathbf{v}_j$	$\frac{\partial \mathbf{v}_i}{\partial \mathbf{v}_j} = 2\mathbf{z}\mathbf{z}^T - I$

$[\mathbf{X}_1^T, \dots, \mathbf{X}_N^T]^T$  holds all the points coordinates and  $\mathbf{B}(\theta_1, \theta_2, \dots, \theta_D)$ , a  $P \times 3N$  matrix, holds the geometrical constraints. It is easy to see that the dimension  $M$  of the nullspace of  $\mathbf{B}$  is greater than 3 and thus an *implicit parameterization* of the points  $\mathbf{X}$  is

$$\mathbf{X} = \mathbf{U}(\theta_1, \theta_2, \dots, \theta_D) \mathbf{V} \quad (2)$$

with  $\mathbf{V} \in \mathbb{R}^M$  and where  $\mathbf{U}(\theta_1, \theta_2, \dots, \theta_D)$  ( $\mathbf{U}$  for brevity) is a  $3N \times M$  matrix whose columns form an orthonormal basis of the nullspace of  $\mathbf{B}$ . Thus  $\mathbf{X}$  *implicitly verify all the geometrical constraints and  $\mathbf{V}$  holds the implicit parameters of the shape*. By sequential construction of the dependences graph, normal vectors verify the geometrical constraints and thus *any values* for  $\mathbf{V}$  supply a solution for the constrained system. Consequently,  $M$  is the number of degrees of freedom of  $\mathbf{X}$ .

As for the unknowns,  $\theta$  and  $\mathbf{V}$ , they are collected in a single vector  $\Theta = [\theta, \mathbf{V}]$ .

## 3.2 Maximum Likelihood Reconstruction

We choose to estimate the solution *subject to geometrical constraints* that is most likely to verify the initial 3D observations. The solution should then minimize the function:

$$\mathcal{Q}(\Theta) = \|\mathbf{X}_{ini} - \mathbf{X}(\Theta)\|^2 \quad (3)$$

A classical iterative Levenberg-Marquardt procedure is performed for this minimization. We assume the reader is familiar with Levenberg-Marquardt algorithm and highlight some of the important difficulties related to the computations of the cost function and its derivatives at different steps  $i$ . The initialization of  $\theta_i$  is directly linked to the initial values of the normal vectors of each facet. As for  $\mathbf{V}$ , we choose  $\mathbf{V}^0 = (\mathbf{U}^0)^+ \mathbf{X}_{ini}$  where  $(\mathbf{U}^0)^+$  stands for the pseudo-inverse of  $\mathbf{U}^0$ .  $\mathbf{U}^0$  can be chosen arbitrarily as orthonormal basis of the nullspace of  $\mathbf{B}$  but some provisions must then be taken to ensure the unicity of  $\mathbf{U}$  in the next steps of the iterative process. Starting from an initial value  $\mathbf{U}^s$ , and assuming  $\mathcal{U}_1$  is any unitary matrix whose columns

form a basis of  $\text{Null}(\mathbf{B}(\boldsymbol{\theta}^s))$ , the procedure 4 ensures that  $\mathbf{U}^i$  is uniquely defined.

$$\begin{aligned}\mathbf{U}_1^T \mathbf{U}^s &\stackrel{\text{SVD}}{=} \mathcal{U} \mathcal{D} \mathcal{V}^T \\ \mathbf{U}^i &= \mathbf{U}_1 \mathcal{U} \mathcal{V}^T\end{aligned}\quad (4)$$

It can also be shown that, thanks to this procedure,  $\mathbf{U}^i$  is differentiable in a neighborhood of  $\boldsymbol{\theta}^s$  and derivatives are given by 5, where  $\mathbf{B}^+$  stands for the pseudo-inverse of  $\mathbf{B}$  and  $\frac{\partial \mathbf{B}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^s)$  is trivial to compute. From these relations, evaluation and derivatives of  $\mathcal{Q}(\boldsymbol{\Theta}^i)$  become straightforward.

$$\frac{\partial \mathbf{U}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^s) = -\mathbf{B}^+ \frac{\partial \mathbf{B}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^s) \mathbf{U}^s \quad (5)$$

## 4 Handling Over-Constraints and Degeneracies

### 4.1 Over-Constraints and Degeneracies Detection

The dependences graph generation ensures that no over-constraint *on directions* can occur. However, in the constrained system applied to points, nothing ensures that a point is not over-constrained. This remark comes from the fact that the minimization is done in two steps as in [1]: first, directions are computed, and then points *must* conform to these directions. It may be possible that points can not verify all planarity relations because, for instance, in the directions computation, no test has been performed to verify some situations that topology may impose. Degeneracies occur when values are too far away from the constrained situation imposed by the application. In this case, unable to find a acceptable solution, implicit parameterization merge two points and supplies a degenerate solution.

In our scheme, over-constrained situations as well as degeneracy situations are easily detected, as stated in [14] by examining  $\mathbf{U}^i$  at any step of the algorithm. If any triplets  $\{3m-2, 3m-1, 3m\}$  and  $\{3n-2, 3n-1, 3n\}$  ( $m \neq n$ ) of rows, corresponding to two different points are equal, it means that the only available solution was to merge these two points and a degeneracy has been detected.

### 4.2 Over-Constraints and Degeneracies Handling

A good way of dealing with this would be an automatic property detection that could add some additional constraints to the directions set due to topology considerations [10]. In our case, when degeneracies or overconstraints are detected during geometric parametrization, constraints are iteratively removed until no more degeneracies or overconstraints occur, and the minimization process is reinitialized. The heuristic algorithm uses procedures to favor constraints whose suppressing decrease the least the global number of degrees of freedom in order to keep as many constraints as possible. At the end of the algorithm, in case of success, all topological relations are verified insofar as all planarity constraints on facets embedded in the implicit parameterization are kept.

## 5 Results

### 5.1 General Protocol

For simulation purpose, 25 reference buildings have been obtained from automatic algorithm [2] (without constraints), corresponding to 238 points. From this modelling, constraints are inferred and enforced using the algorithm depicted, thus giving a constrained reference that will be the basis of our comparison: all results will be made with regard to this (constrained) reference. To evaluate our algorithm, gaussian noise is added to points with an increasing standard deviation and reconstruction is performed, taking these noisy points as  $\mathbf{X}_{ini}$  in the algorithm. For all results given, average is made on 25 iterations. parameters used were:  $\sigma_{\parallel} = 12^\circ$  and  $\sigma_{\perp} = 10^\circ$ .

Two cases are considered for evaluation on noisy buildings: a priori constraints and a posteriori constraints. The first case is comparable to that of model-based reconstruction: constraints are known *a priori* by an external information (brought by constraints inferred from the reference models), thus avoiding the directions clustering and constraint graph building step (but initial values are obviously computed from the noisy buildings). In the second case, constraints are systematically inferred from the noisy model. It is therefore obvious that some constraints will be lost compared to the reference buildings.

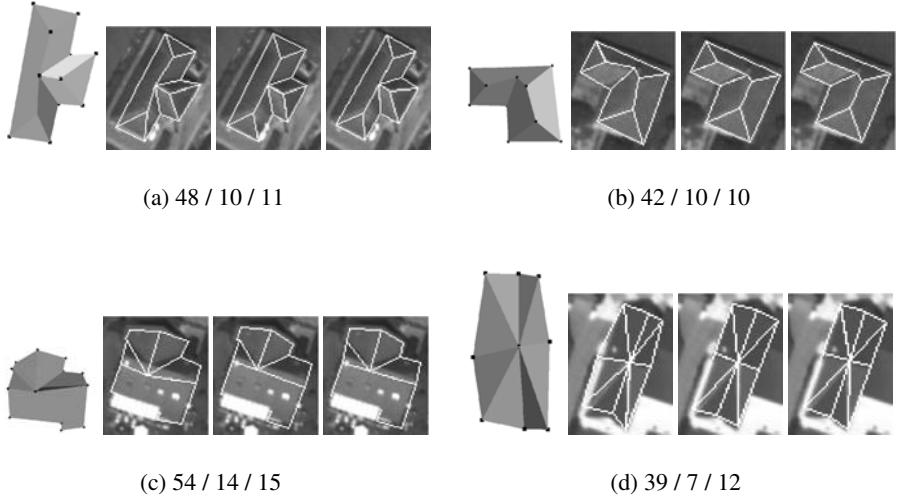
On the one hand, we analyze the gain in precision, that is the average distance between reconstructed buildings and constrained reference. Recall that minimization is done however with regard to the noisy points ! On the other hand gain in degree of freedom is studied to show the rigidity that constraints bring to reconstruction. Without any constraint, the global degree of freedom of the 25 modelled buildings is 966.

### 5.2 A priori

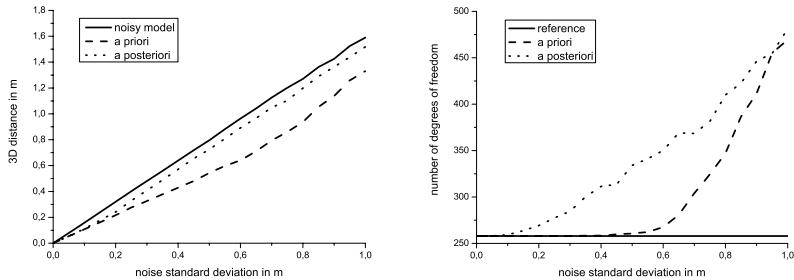
In this case, as can be seen for some noise in figure 2, reconstruction is close to what is expected, the global shape is much more acceptable. Figure 3 shows more quantitative results. One can see that distance is always lower than without constraint, which states that constraints naturally tend to get closer to reference. In the case of a priori constraints, reduction of degree of freedom is only due to over-constraints or degeneracies detected in the minimization process and that led to constraints suppression. A great stability to noise can be seen for a priori constraint that is not sensitive to noise until  $\sigma = 0.5m$ . Let us note also that a manual evaluation of degree of freedom of the buildings has showed that 258 was the expected degree of freedom, which is the number found by the algorithm when noise is low. It proves thus that no constraints is lost during the process and particularly during the dependence graph construction step on the reference and therefore validates our procedures.

### 5.3 A posteriori

Although all the constraints are not inferred as soon as noise increases, global shape of buildings is still much more acceptable than without constraints (Figure 2). Let us note that  $\sigma = 0.3$  is an important noise especially when one-meter edges are present in the



**Fig. 2.** Some views of several buildings with  $\sigma = 0.5m$ . first column: views of noisy buildings in 3D. 3 next columns: views of a noisy builing, a priori constraints enforcing and a posteriori constraints enforcing. The three numbers indicates the number of degrees of freedom for reconstruction without constraint, with a priori constraints and with a posteriori constraints respectively



**Fig. 3.** left: distance to the reference. right: number of degrees of freedom. In each graph, x-axis corresponds to standard deviation of noise added on points

scene. As far as degree of freedom is concerned, a relative stability can be observed up to  $\sigma = 0.3$  where only 10% of degrees of freedom is lost.

## 6 Summary and Conclusion

We have presented a method for automatic inferring and enforcing constraints on 3D models. This method requires no user intervention for constraints definition and, by implicitly parameterizing the model, enables to know exactly which constraints are verified on the reconstructed model. The approach uses few thresholds and han-

dles over-constraints by iteratively removing some constraints. Results show the gain obtained by this algorithm as far as degree of freedom and precision are concerned.

Future work include the integration of image features in the whole loop so as to fit the constrained models on points of interest in the images. This would solve some of the artefacts visible in 2 where edges of the constrained model do not match perfectly segments of images. Fitting a 3D model on image features is a difficult problem since mismatched and non-matched features must be handled. In this case, the model should take full benefit of the reduction of degree of freedom due to constraints to overcome non-matched or mismatched features. Another point to study is the possibility of overcoming over-constraint without heuristic suppressing of constraints. Automatic property detection seems promising to add implicit constraints due, for instance, to topology.

## References

1. Shum, H.Y., Han, M., Szeliski, R.: Interactive construction of 3d models from panoramic mosaics. In: Conference on Computer Vision and Pattern Recognition. (1998) 427–433
2. Taillandier, F., Deriche, R.: Automatic buildings reconstruction from aerial images: a generic bayesian framework. In: Proceedings of the XXth ISPRS Congress, Istanbul, Turkey (2004)
- 3.Debevec, P., Taylor, C., Malik, J.: Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. In: SIGGRAPH 96. (1996) 11–20
4. Suveg, I., Vosselman, G.: 3D reconstruction of building models. In: Proceedings of the XIXth ISPRS Congress. Volume 33., Amsterdam (2000) B3:538–545
5. Fischer, A., Kolbe, T., Lang, F., Cremers, A., Förstner, W., Plümer, L., Steinhage, V.: Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D. Computer Vision and Image Understanding **72** (1998) 163–185
6. Jaynes, C., Riseman, E., Hanson, A.: Recognition and reconstruction of buildings from multiple aerial images. Computer Vision and Image Understanding **90** (2003) 68–98
7. Noronha, S., Nevatia, R.: Detection and modeling of buildings from multiple aerial images. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 501–518
8. Kim, Z., Huertas, A., Nevatia, R.: Automatic description of buildings with complex rooftops from multiple images. In: Conference on Computer Vision and Pattern Recognition. Volume 2. (2001) 272–279
9. Grossmann, E., Santos-Victor, J.: Maximum likelihood 3d reconstruction from one or more images under geometric constraints. In: British Machine Vision Conference. (2002)
10. Bondyfalat, D.: Interaction entre Symbolique et Numérique; Application à la Vision Artificielle. Thèse de doctorat, Univ de Nice Sophia-Antipolis (2000)
11. van den Heuvel, F.A.: A line-photogrammetric mathematical model for the reconstruction of polyhedral objects. In: Videometrics VI, SPIE. Volume 3641. (1999) 60–71
12. Ameri, B., Fritsch, D.: Automatic 3D building reconstruction using plane-roof structures. In: ASPRS, Washington DC (2000)
13. Hoffmann, C., Vermeer, P.: Geometric constraint solving in  $R^2$  and  $R^3$ . In: Computing in Euclidean Geometry. 2<sup>nd</sup> edn. World Scientific Publishing (1995) 266–298
14. Grossmann, E.: Maximum Likelihood 3D Reconstruction From One or More Uncalibrated Views Under Geometric Constraints. PhD thesis, Universidade Tecnica de Lisboa (2002)

# Aligning Shapes by Minimising the Description Length

Anders Ericsson and Johan Karlsson

Centre for Mathematical Sciences,

Lund University, Lund, Sweden

{anderse, johank}@maths.lth.se

**Abstract.** When building shape models, it is first necessary to filter out the similarity transformations from the original configurations. This is normally done using Procrustes analysis, that is minimising the sum of squared distances between the corresponding landmarks under similarity transformations. In this article we propose to align shapes using the Minimum Description Length (MDL) criterion. Previously MDL has been used to locate correspondences. We show that the Procrustes alignment with respect to rotation is not optimal.

The MDL based algorithm is compared with Procrustes on a number of data sets. It is concluded that there is improvement in generalisation when using Minimum Description Length. With a synthetic example it is shown that the Procrustes alignment can fail significantly where the proposed method does not.

The Description Length is minimised using Gauss-Newton. In order to do this the derivative of the description length with respect to rotation is derived.

## 1 Introduction

Statistical models of shape [5] has turned out to be a very effective tool in image segmentation and image interpretation. Such models are particularly effective in modelling objects with limited variability, such as medical organs.

The basic idea behind statistical models of shape is that from a given training set of known shapes be able to describe new formerly unseen shapes, which still are representative. The shape is traditionally described using landmarks on the shape boundary.

When building shape models, it is first customary to filter out the similarity transformations from the original configurations. The common way to align shapes before building shape models is to do a Procrustes analysis [8, 5]. It locates the unknown similarity transformations by minimising the sum of squared distances from the mean shape. Other methods exist, in Statistical Shape Analysis [10, 2, 13, 5] Bookstein and Kendall coordinates are commonly used to filter out the effect of similarity transformations.

Minimum Description Length, (MDL) [12], is a paradigm that has been used in many different applications, often in connection with evaluating a model. In

recent papers [4, 3, 7, 9] this paradigm is used to locate a dense correspondence between the boundaries of shapes.

In this paper we apply the theory presented in [11] and derive the gradient of the description length [7] and propose to align shapes using the Minimum Description Length (MDL) criterion. In this paper the experiments are done for 2D-shapes, but in principle it would work for any dimension.

It turns out that when aligning shapes using MDL instead of Procrustes the translation becomes the same but the optimal rotation is different. In this paper the scale of all shapes are normalised to one.

The gradient of the description length with respect to the rotation is derived and used to optimise the description length using Gauss-Newton.

The proposed algorithm is tested on a number of datasets and the mean square error of leave one out reconstructions turns out to be lower than or the same as for Procrustes analysis. Using a synthetic example we show that the alignment using description length can get more intuitive and give much lower mean square error on leave one out reconstructions than when using Procrustes.

This paper is organised as follows. In Section 2 the necessary background on shape models, MDL and SVD is given. In Section 3, the gradient of the description length is derived by calculating the gradient of the SVD. This is used to optimise the description length using Gauss-Newton. In Section 4 results of experiments are presented and it is shown that better models are achieved using the description length to align shapes.

## 2 Preliminaries

### 2.1 Statistical Shape Models

When analysing a set of  $n_s$  similar (typically biological) shapes, it is convenient and usually effective to describe them using Statistical Shape Models. Each shape is typically the boundary of some object and is in general represented by a number of landmarks. After the shapes  $\mathbf{x}_i$  ( $i = 1 \dots n_s$ ) have been aligned and normalised to the same size, a PCA-analysis [9] is performed. The alignment is what has been improved in this paper. The  $i$ -th shape in the training set can now be described by a linear model of the form,

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i \quad , \quad (1)$$

where  $\bar{\mathbf{x}}$  is the mean shape, the columns of  $\mathbf{P}$  describe a set of orthogonal modes of shape variation and  $\mathbf{b}_i$  is the vector of shape parameters for the  $i$ -th shape.

### 2.2 MDL

The description length (DL) is a way to evaluate a shape model. The cost in Minimum Description Length (MDL) is derived from information theory and is, in simple words, the effort that is needed to send the model and the shapes bit by bit. The MDL - principle searches iteratively for a model that can transmit

the data the cheapest. The cost function makes a tradeoff between a model that is general (can represent any instance of the object), and compact (it can represent the variation with as few parameters as possible). Davies and Cootes relates these ideas to the principle of Occam's razor : the simplest explanation generalises the best.

Since the idea of using MDL for shape models was first published [4] the cost function has been refined and tuned. Here we use a refined version of the simple cost function stated in [14] and derived in [6]

$$DL = \sum_{\lambda_i \geq \lambda_c} \left(1 + \log \frac{\lambda_i}{\lambda_c}\right) + \sum_{\lambda_i < \lambda_c, \lambda_i \geq \lambda_t} \frac{\lambda_i}{\lambda_c} . \quad (2)$$

This is a simplified expression, where terms that are constant with respect to alignment has been cancelled. Each term in 2 is the description length for each principal axis. The scalar  $DL$  is the description length and is the cost to transmit the model according to information theory. The scalars  $\lambda_i$  are the eigenvalues of the covariance matrix  $(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T$ , where  $\mathbf{X}$  is the matrix, which rows are the shape configurations in the training set. The constants  $\lambda_c$  and  $\lambda_t$  are calculated from  $\Delta$ , which describes the precision of the landmark coordinates. Information can only be sent up to a certain degree of accuracy and  $\Delta$  expresses this accuracy. The constant  $\lambda_c = 2\Delta$  is the limit between what is expected to be information and what is expected to be noise. When the range of the data in a mode is smaller than  $\Delta$  no information needs to be sent. This limit is set by  $\lambda_t = \Delta$ .

There are two important properties of this cost-function. It is more intuitive than those formerly presented and the derivative is continuous.

### 2.3 Recapitulation of the SVD

In the rest of the paper, bold letters will be used for denoting vectors and matrices. The transpose of matrix  $\mathbf{M}$  is denoted by  $\mathbf{M}^T$  and  $m_{ij}$  refers to the  $(i,j)$  element of  $\mathbf{M}$ . The  $i$ -th non-zero element of a diagonal matrix  $\mathbf{D}$  is referred to by  $d_i$  while  $\mathbf{M}_i$  designates the  $i$ -th column of matrix  $\mathbf{M}$ . The  $i$ -th element of the vector  $\mathbf{x}$  is designated  $\mathbf{x}(i)$ .

A basic theorem of linear algebra states that any real or complex  $M \times N$  matrix  $\mathbf{A}$  can be factored into the product of an  $M \times M$  orthogonal matrix  $\mathbf{U}$ , an  $M \times N$  diagonal matrix  $\mathbf{S}$  with non-negative diagonal elements (known as the singular values), and an  $N \times N$  orthogonal matrix  $\mathbf{V}$ .

In other words,

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T = \sum_{i=1}^N s_i \mathbf{U}_i \mathbf{V}_i^T . \quad (3)$$

The singular values are the square roots of the positive eigenvalues of the matrix  $\mathbf{A}^T \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^T$ .

### 3 Optimising the DL

#### 3.1 Computing the Jacobian of the Singular Values

Here we recapitulate on the theory presented in [11]. For a more mathematical investigation in this field we recommend Alan Andrew's work, especially [1].

All matrices  $\mathbf{A}$  can be factored into  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ , where  $\mathbf{S}$  is a diagonal matrix holding all singular values. We are interested in computing the derivatives of the singular values,  $\frac{\partial s_k}{\partial a_{ij}}$  for every element  $a_{ij}$  of the  $M \times N$  matrix  $\mathbf{A}$ . Taking the derivative of  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  with respect to  $a_{ij}$  gives the following equation

$$\frac{\partial \mathbf{A}}{\partial a_{ij}} = \frac{\partial \mathbf{U}}{\partial a_{ij}} \mathbf{S}\mathbf{V}^T + \mathbf{U} \frac{\partial \mathbf{S}}{\partial a_{ij}} \mathbf{V}^T + \mathbf{U}\mathbf{S} \frac{\partial \mathbf{V}^T}{\partial a_{ij}} . \quad (4)$$

Clearly,  $\forall (k, l) \neq (i, j)$ ,  $\frac{\partial a_{kl}}{\partial a_{ij}} = 0$ , while  $\frac{\partial a_{ij}}{\partial a_{ij}} = 1$ . Since  $\mathbf{U}$  is an orthogonal matrix, we have

$$\mathbf{U}\mathbf{U}^T = \mathbf{I} \Rightarrow \frac{\partial \mathbf{U}^T}{\partial a_{ij}} \mathbf{U} + \mathbf{U}^T \frac{\partial \mathbf{U}}{\partial a_{ij}} = \omega_{\mathbf{U}}^{ijT} + \omega_{\mathbf{U}}^{ij} = \mathbf{0} , \quad (5)$$

where  $\omega_{\mathbf{U}}^{ij}$  is given by

$$\omega_{\mathbf{U}}^{ij} = \mathbf{U}^T \frac{\partial \mathbf{U}}{\partial a_{ij}} . \quad (6)$$

From Equation (5) it is clear that  $\omega_{\mathbf{U}}^{ij}$  is an antisymmetric matrix. Similarly, an anti symmetric matrix  $\omega_{\mathbf{V}}^{ij}$  can be defined for  $\mathbf{V}$  as

$$\omega_{\mathbf{V}}^{ij} = \frac{\partial \mathbf{V}^T}{\partial a_{ij}} \mathbf{V} . \quad (7)$$

Notice that  $\omega_{\mathbf{U}}^{ij}$  and  $\omega_{\mathbf{V}}^{ij}$  are specific to each differentiation  $\frac{\partial}{\partial a_{ij}}$ . By multiplying Equation (4) by  $\mathbf{U}^T$  and  $\mathbf{V}$  from left and right respectively, and using Equations (6) and (7), the following is obtained:

$$\mathbf{U}^T \frac{\partial \mathbf{A}}{\partial a_{ij}} \mathbf{V} = \omega_{\mathbf{U}}^{ij} \mathbf{S} + \frac{\partial \mathbf{S}}{\partial a_{ij}} + \mathbf{S} \omega_{\mathbf{V}}^{ij} . \quad (8)$$

Since  $\omega_{\mathbf{U}}^{ij}$  and  $\omega_{\mathbf{V}}^{ij}$  are antisymmetric matrices, all their diagonal elements are equal to zero. Recalling that  $\mathbf{S}$  is a diagonal matrix, it is easy to see that the diagonal elements of  $\omega_{\mathbf{U}}^{ij} \mathbf{S}$  and  $\mathbf{S} \omega_{\mathbf{V}}^{ij}$  are also zero. Thus, Equation (8) yields the derivatives of the singular values as:

$$\frac{\partial s_k}{\partial a_{ij}} = u_{ik} v_{jk} . \quad (9)$$

#### 3.2 The Gradient of the DL

Let  $\mathbf{x}_1, \dots, \mathbf{x}_{n_s}$  be  $n_s$  shapes centred at the origin. The rotation of shape  $m$  is denoted  $\theta_m$ . Differentiating (2) with respect to  $\theta_m$ , we get the following expression

$$\frac{\partial DL}{\partial \theta_m} = \sum_{\lambda_k \geq \lambda_c} \frac{1}{\lambda_k} \frac{\partial \lambda_k}{\partial \theta_m} + \sum_{\lambda_k < \lambda_c, \lambda_k \geq \lambda_t} \frac{1}{\lambda_c} \frac{\partial \lambda_k}{\partial \theta_m} . \quad (10)$$

We want to calculate  $\frac{\partial \lambda_k}{\partial \theta_m}$ . Let the  $m$ -th row of  $\mathbf{X}$  be the configuration of landmarks for shape  $m$  after moving the centre of gravity to the origin, normalising scale so that the Euclidian norm is one and rotating according to  $\theta_m$ .

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 e^{i\theta_1} \\ \vdots \\ \mathbf{x}_{n_s} e^{i\theta_{n_s}} \end{bmatrix}$$

Let  $\mathbf{Y}$  be the matrix holding the deviations from the mean shape,

$$\mathbf{Y} = \mathbf{X} - \bar{\mathbf{X}} ,$$

where each row in  $\bar{\mathbf{X}}$  is the mean shape  $\bar{\mathbf{x}}$ ,

$$\bar{\mathbf{x}} = \frac{1}{n_s} \sum_{j=1}^{n_s} \mathbf{x}_j e^{i\theta_j} .$$

If we apply principal component analysis to  $\mathbf{Y}$ , we can describe our shapes with the linear model in equation (1). A singular value decomposition of  $\mathbf{Y}$  gives us  $\mathbf{Y} = \mathbf{USV}^T$ . Here  $\mathbf{V}$  corresponds to  $\mathbf{P}$  in equation (1) and the diagonal matrix  $\mathbf{S}^T \mathbf{S}$  holds the eigenvalues  $\lambda_k$ .

Now, if  $y_{mj}$  is the  $j$ -th landmark on shape  $m$  and  $\frac{\partial y_{mj}}{\partial \theta_m}$  is the derivative of the  $j$ -th landmark on shape  $m$  with respect to the rotation  $\theta_m$  then

$$\frac{\partial \lambda_k}{\partial \theta_m} = \frac{\partial s_k^2}{\partial \theta_m} = 2s_k \frac{\partial s_k}{\partial \theta_m} = 2s_k \sum_{pq} \frac{\partial s_k}{\partial y_{pq}} \frac{\partial y_{pq}}{\partial \theta_m} = 2s_k \sum_{pq} u_{pk} v_{qk} \frac{\partial y_{pq}}{\partial \theta_m} , \quad (11)$$

Here it is used that  $\frac{\partial s_k}{\partial y_{pq}} = u_{pk} v_{qk}$ , where  $u_{pk}$  and  $v_{qk}$  are elements in  $\mathbf{U}$  and  $\mathbf{V}$ , see section 3.1.

$$\frac{\partial y_{pq}}{\partial \theta_m} = \frac{\partial \mathbf{x}_p(q) e^{i\theta_p}}{\partial \theta_m} - \frac{1}{n_s} \sum_j \frac{\partial \mathbf{x}_j(q) e^{i\theta_j}}{\partial \theta_m} = \begin{cases} i(1 - \frac{1}{n_s}) \mathbf{x}_m(q) e^{i\theta_m} & p = m \\ -i \frac{1}{n_s} \mathbf{x}_m(q) e^{i\theta_m} & p \neq m \end{cases} \quad (12)$$

$$\text{since } \frac{\partial \mathbf{x}_p(q) e^{i\theta_p}}{\partial \theta_m} = \begin{cases} i \mathbf{x}_m(q) e^{i\theta_m} & p = m \\ 0 & p \neq m \end{cases} .$$

If  $n_s$  is large (this is assumed in our implementation), the second term in (12) can be ignored. Then  $\frac{\partial \lambda_k}{\partial \theta_m}$  can be written as

$$\frac{\partial \lambda_k}{\partial \theta_m} = 2s_k i u_{mk} \mathbf{y}_m \mathbf{V}_k , \quad (13)$$

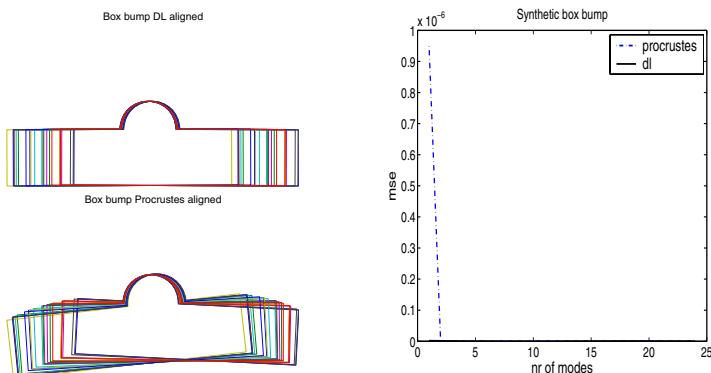
where  $\mathbf{V}_k$  is the  $k$ -th column in  $\mathbf{V}$  and  $\mathbf{y}_m$  is the  $m$ -th row in  $\mathbf{Y}$ .

### 3.3 Optimisation

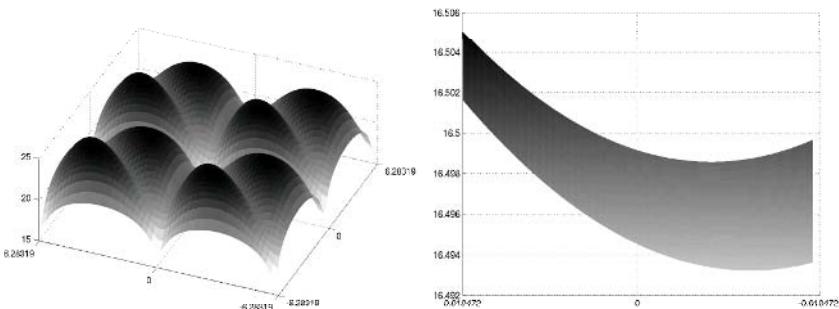
When the gradient of an objective function is known for a specific optimisation problem, it generally pays off to use more sophisticated optimisation techniques that use the gradient. Initially the configurations are translated to the origin, since this is optimal for the description length goal function. Then the shapes are normalised so that the Euclidian norm is one for all shapes. It could be interesting to also optimise scale but the global scale must then be preserved. This means that it would be necessary to optimise under constraints. In this work only rotation is optimised using Gauss-Newton.

## 4 Experimental Validation

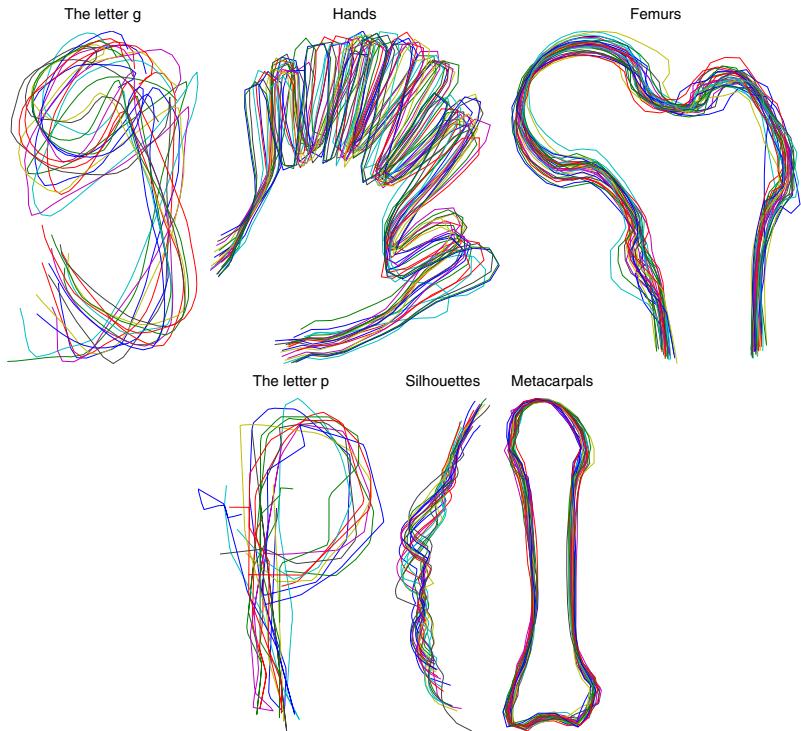
The experiments were conducted in the following way. Given a dataset the centre of gravity was moved to the origin for all shapes and scale was normalised to



**Fig. 1.** A synthetic example shows that Procrustes alignment can fail (lower). Note that the description length approach succeeds (upper)



**Fig. 2.** The description length goal function. In the left figure the range on each axis is  $-2\pi$  to  $2\pi$ . The right figure zooms in on the origin showing that the minimum of the 3D surface is not at the origin



**Fig. 3.** The description length aligned datasets

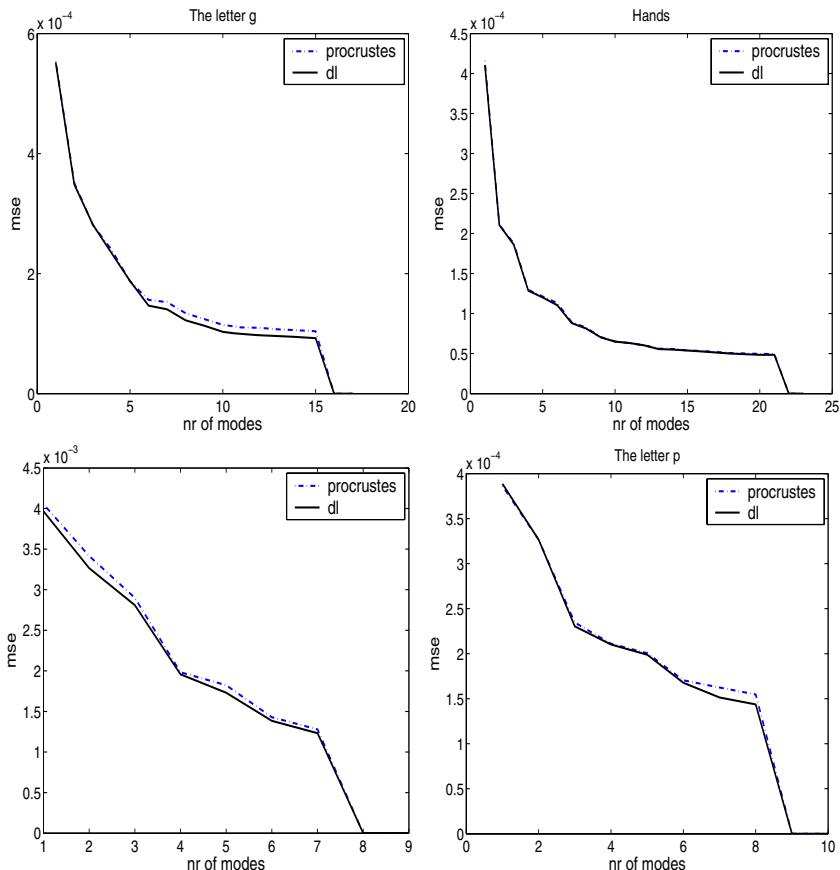
one according to the Euclidian norm. The initialisation for the optimisation for rotation was set to the rotation according to Procrustes. The rotation was then optimised by minimising the Description Length.

In Figure 2 the typical behaviour of the goal function can be seen. Here two rotations ( $x$  and  $y$  axis) have been optimised to align three shapes. In the left figure it can be seen that the minimum is a well defined global minimum. It seems to be several minima, but this is since the range goes from  $-2\pi$  to  $2\pi$  in  $x$  and  $y$ . The right figure zooms in on the origin (the origin corresponds to the Procrustes solution). It can be seen that the minimum is not at the origin. When more shapes are aligned projections of the goal function looks similar to these plots.

We validate our algorithm on five real data sets, see Figure 3.

... 23 contours of a hand segmented out semi-automatically from a video stream. To simplify the segmentation it was filmed on a dark background. The contours were sampled in 64 landmarks using arc-length parameterisation.

... 32 contours of femurs taken from X-rays in the supine projection. The contours were sampled in 64 landmarks using arc-length parameterisation.



**Fig. 4.** The mean squared error of leave one out reconstructions of the g-dataset, the hand dataset, the three dataset and the 90 p shapes

24 contours of metacarpals (a bone in the hand) deduced from standard projection radiographs of the hand in the posterior-anterior projection. The contours were sampled in 57 landmarks.

The silhouette data set consists of 22 contours of silhouettes of digital camera. The silhouettes were then extracted using an edge detector. The contours were sampled in 81 landmarks.

One data set of 17 curves of the letter g. The curves of the letter g are sampled using a device for handwriting recognition. The contours were sampled in 64 landmarks using arc-length parameterisation.

One data set of 90 curves of the letter p. This letter was taken from the MIT database of Latin letters, initially collected by Rob Kassel at MIT

(<ftp://lightning.lcs.mit.edu/pub/handwriting/mit.tar.Z>). The contours were sampled in 128 landmarks using arc-length parameterisation.

The quality of the models was measured as the mean square error in leave-one-out reconstructions. The model is built with all but one example and then fitted to the unseen example. This is shown in Figure 4. The plot shows the mean squared approximation error against the number of modes used. This measures the ability of the model to represent unseen shape instances of the object class.

For all examples we get models that give the same or lower error when using the description length criterion compared to Procrustes alignment. This means that the models generalise better. The improvements are consistent but small. Using this alignment of course the computational cost increases but the model is only built once.

In Figure 1 is an example of when the Procrustes goes visibly wrong. It is a synthetic example with 24 shapes built up by 128 landmarks. For a human it would be natural to align the boxes and let the bump be misaligned. These shapes are built up with a majority of landmarks around the bumps and therefore the Procrustes method will minimise the error between the bumps instead of the boxes. Note that this data only has one shape mode and therefore perfect alignment should give zero mean squared error on the leave one out reconstruction using just one mode. In this example the description length aligned box bump gets almost zero error on the first shape mode.

## 5 Summary and Conclusions

In this paper we present a new way to align shapes. The rotation is located by minimising the description length. We derive the gradient of the description length with respect to the rotation and propose to use Gauss-Newton to minimise the MDL-criterion. We have shown that the objective function is differentiable and can be written explicitly.

We have compared the proposed algorithm to Procrustes alignment and shown that better models can be achieved.

One reason why the description length alignment does not get even better results is that when there are many shapes the path to the minimum is difficult for the optimiser to follow. The derivatives gets numerically unstable close to the minimum.

## Acknowledgements

Pronosco is acknowledged for providing the contours of femurs and metacarpals. We also like to thank Jesper Skjerning (IMM, DTU) for the silhouettes, Hans Henrik Thodberg (IMM, DTU) for the box bumps and Rob Kassel (MIT) for the Latin letters. And finally we thank Hans Bruun Nielsen (IMM, DTU) for the implementation of the Gauss-Newton optimiser. This work has been financed by the SSF sponsored project 'Vision in Cognitive Systems' (VISCOS), and the

project 'Information Management in Dementias' sponsored by the Knowledge Foundation and UMAS in cooperation with CMI at KI.

## References

1. A. Andrew, E. Chu, and P. Lancaster. Derivatives of eigenvalues and eigenvectors of matrix functions. pages 903–926, 1993.
2. F. L. Bookstein. Size and shape spaces for landmark data in two dimensions. *Statistical Science*, 1(2):181–242, 1986.
3. R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. medical imaging*, 21(5):525–537, 2002.
4. Rhodri H. Davies, Tim F. Cootes, John C. Waterton, and Chris J. Taylor. An efficient method for constructing optimal statistical shape models. In *Medical Image Computing and Computer-Assisted Intervention MICCAI'2001*, pages 57–65, 2001.
5. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, Inc., 1998.
6. A. Ericsson. *Automatic Shape Modelling and Applications in Medical Imaging*. PhD thesis, Mathematics LTH, Lund University, Centre for Mathematical Sciences, Box 118, SE-22100, Lund, Sweden, nov 2003.
7. A. Ericsson and K. Åström. Minimizing the description length using steepest descent. In *Proc. British Machine Vision Conference, Norwich, United Kingdom*, volume 2, pages 93–102, 2003.
8. J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–50, 1975.
9. J. Karlsson, A. Ericsson, and K. Åström. Parameterisation invariant statistical shape models. In *Proc. International Conference on Pattern Recognition, Cambridge, UK*, 2004.
10. D.G Kendall, D Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. John Wiley & Sons Ltd., 1999.
11. T. Papadopoulo and M. Lourakis. Estimating the jacobian of the singular value decomposition. In *Proc. European Conf. on Computer Vision, ECCV'00*, pages 555–559, 2000.
12. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
13. C. G. Small. *The Statistical Theory of Shape*. Springer, 1996.
14. H. H. Thodberg. Minimum description length shape and appearance models. In *Image Processing Medical Imaging, IPMI*, 2003.

# Segmentation of Medical Images Using Three-Dimensional Active Shape Models

Klas Josephson, Anders Ericsson, and Johan Karlsson

Centre for Mathematical Sciences,

Lund University, Lund, Sweden

{f00kj, anderse, johank}@maths.lth.se

<http://www.maths.lth.se>

**Abstract.** In this paper a fully automated segmentation system for the femur in the knee in Magnetic Resonance Images and the brain in Single Photon Emission Computed Tomography images is presented. To do this several data sets were first segmented manually. The resulting structures were represented by unorganised point clouds. With level set methods surfaces were fitted to these point clouds. The iterated closest point algorithm was then applied to establish correspondences between the different surfaces. Both surfaces and correspondences were used to build a three dimensional statistical shape model. The resulting model is then used to automatically segment structures in subsequent data sets through three dimensional Active Shape Models. The result of the segmentation is promising, but the quality of the segmentation is dependent on the initial guess.

## 1 Introduction

Hospitals today produce numerous diagnostic images such as Magnetic Resonance Imaging (MRI), Single Photon Emission Computed Tomography (SPECT), Computed Tomography (CT) and digital mammography. These technologies have greatly increased the knowledge of diseases and they are a crucial tool in diagnosis and treatment planning.

Today almost all analysis of images is still done by manual inspection by the doctors even though the images are digitalised from the beginning. Even for an experienced doctor the diagnosis can be hard to state and it is often time consuming, especially in three dimensional images.

Active Shape Models (ASM) is a segmentation algorithm that can handle noisy data. Cootes et al. have in [1] used ASM to segment out cartilage from two dimensional MR images of the knee. But the MR images are produced in three dimensions and it would be of interest to get a three dimensional representation of the structures.

To do this several problems have been solved. First surfaces have been fitted to unorganised point clouds through a level set method. After that corresponding parametrisations are established over the training set with the Iterated Closest

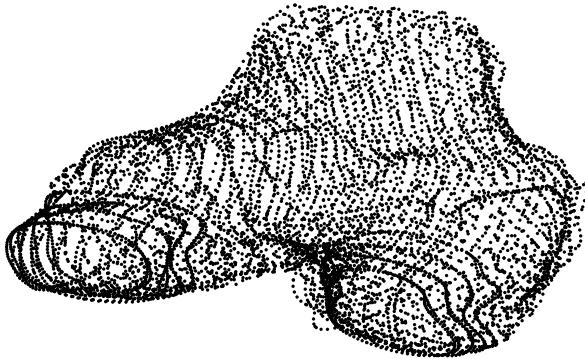
Point (ICP) algorithm. The next step is to build the shape model with Principal Component Analysis (PCA). Finally the model is used to segment new images.

This paper starts in the next section with the theory for the surface fitting algorithm. Section 3 includes the method for finding corresponding points over the training set and Sect. 4 describes how to align those points. In Sect. 5 the method for building the shape model is presented. Sect. 6 holds the algorithms used to segment objects with the aid of the shape model. Later in Sect. 7 the result of the work is showed.

## 2 Shape Reconstruction from Point Clouds

From an unorganised point cloud the aim is to get a triangulation of the surface that the points are located at. To do that first of all the point cloud of the femur has to be constructed. After that a level set approach is used for the surface fitting.

The point cloud is constructed by manual marking in the training set. With cubic splines a denser point cloud than the marked one is achieved. A typical point cloud of the femur can be seen in Fig. 1.



**Fig. 1.** Unorganised point cloud of the femur

### 2.1 Calculating the Triangulation

To handle the noisy representation of the femur a level set approach is used to reconstruct the surface. In [2] Zhao et al. developed a method which reconstructs a surface that is minimal to the distance transform to the data set. This approach has problem when the point clouds are noisy. Later Shi and Karl [3] proposed a data-driven, Partial Differential equation (PDE) based, level set method that handles noisy data.

The idea of the level set method is to represent the surface as an implicit distance function which is zero at the surface and negative inside. The function

is then updated to solve a PDE that is constructed in such a way that it will have a minimum at the true surface. By updating the function iteratively it will fit to the surface. The problem is formulated as follows:

Denote the points in the point cloud  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ . And the distance function  $\phi$  where  $\phi = 0$  is the surface. With aid of the distance function the signed distance from a point to the surface, denoted as  $g(\mathbf{x}_i, \phi)$ , can be written

$$g(\mathbf{x}_i, \phi) = \phi(\mathbf{x}_i) = \int \phi(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}_i) dx, \quad (1)$$

where  $\delta$  is the delta distribution. The distances are then collected in a vector  $\mathbf{g}(\mathbf{X}, \phi)$ . The problem can now be rewritten as an energy minimisation problem. The energy is then

$$E(\phi) = \underbrace{-\log p(\mathbf{g}(\mathbf{X}, \phi) | \phi)}_{E_d} + \mu \underbrace{\int \delta(\phi) |\nabla \phi| dx}_{E_s}, \quad (2)$$

where  $p(\mathbf{g}(\mathbf{X}, \phi) | \phi)$  is the probability of the points  $\mathbf{X}$  given the shape  $\phi$ . Here it is assumed that  $p(\mathbf{g}(\mathbf{X}, \phi) | \phi)$  is a Gaussian distributed:

$$p(\mathbf{g}(\mathbf{X}, \phi) | \phi) \propto e^{-(\mathbf{g} - \mathbf{u})^T \mathbf{W}^{-1} (\mathbf{g} - \mathbf{u}) / 2}, \quad (3)$$

where  $\mathbf{u}$  is the mean distance and  $\mathbf{W}$  is the covariance matrix.

The distance function  $\phi$  also has to fulfil the constraint  $|\nabla \phi| = 1$  to ensure that it is a signed distance function. The  $E_d$  term describes how well the surface is fitted to the points and the  $E_s$  term is for smoothing.

In each iteration of the surface reconstruction a step is taken in the steepest decent direction of the gradient. For the  $E_d$  the gradient is obtained as:

$$\frac{dE_d}{d\phi} = \frac{dE_d^T}{d\mathbf{g}} \frac{d\mathbf{g}}{d\phi}. \quad (4)$$

From (2) we have

$$\frac{dE_d}{d\mathbf{g}} = -\frac{1}{p} \frac{dp}{d\mathbf{g}}$$

and

$$\frac{d\mathbf{g}}{d\phi} = [\delta(\mathbf{x} - \mathbf{x}_1), \dots, \delta(\mathbf{x} - \mathbf{x}_n)]^T.$$

Now approximate  $\delta(\mathbf{x})$  with

$$\delta_\alpha(\mathbf{x}) = \begin{cases} 0, & |\mathbf{x}| > \alpha, \\ \frac{1}{2\alpha} \left[ 1 + \cos \left( \frac{\pi|\mathbf{x}|}{\alpha} \right) \right], & |\mathbf{x}| < \alpha, \end{cases} \quad (5)$$

where  $\alpha \geq 0$ . Finally put the evolution step due to the negative gradient direction so that the step due to the  $E_d$  term is

$$\left[ \frac{d\phi}{dt} \Big|_{\mathbf{x}=\mathbf{x}_0} \right]_d = - \sum_{i=1}^n \frac{dE_d}{d\mathbf{g}(\mathbf{x}_i, \phi)} \delta_\alpha(\mathbf{x}_0 - \mathbf{x}_i). \quad (6)$$

To construct the component of the evolution speed for the function  $\phi$  due to the data term over the whole domain, denoted as  $F_d$ , so that  $\phi$  will remain a signed distance function, the method from [4] is followed. Thus solving of the PDE:

$$\nabla\phi \cdot \nabla F_d = 0 \quad (7)$$

with the boundary condition

$$F_d(\mathbf{x}_0) = \left[ \frac{d\phi}{dt} \Big|_{\mathbf{x}=\mathbf{x}_0} \right]_d \quad (8)$$

obtained from (6) is made. This PDE is derived so that  $|\nabla\phi| = 1$  will remain true. The smooth terms evolution speed is as Zhao et al. showed in [5]

$$\left[ \frac{d\phi}{dt} \right]_s = \left( \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \right) |\nabla\phi|. \quad (9)$$

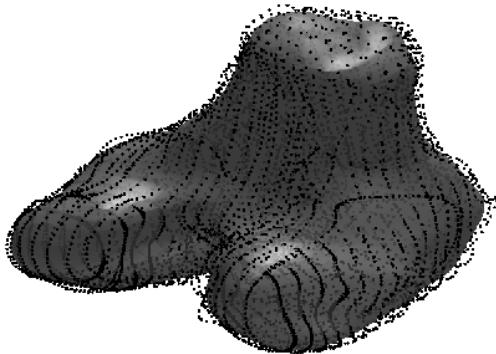
Combining the two terms of energy gives the final evolution speed as

$$\frac{d\phi}{dt} = F_d |\nabla\phi| + \mu \left( \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \right) |\nabla\phi|. \quad (10)$$

Here  $\mu$  is used to make a good balance between the surface fitting and the smoothing.

The use of the approximated delta distribution makes the method robust to local minima, because a point only affects the surface close to the point. Thus it is possible to make an initial guess close to the real surface which reduce the time for the algorithm significantly. The  $\alpha$  parameter in (5) can be varied to make the surface fitting algorithm able to handle data sets with different sparsity.

Between every step of the iteration the fast marching algorithm is used to update the distance function. This is necessary to keep  $|\nabla\phi| = 1$ . For more about the fast marching algorithm see Adalsteinsson and Sethian in [4].



**Fig. 2.** The surface fitted to an unorganised point cloud. Even though the surface is a little bit transparent it is hard to see the points inside the femur, this makes the surface look like it lies inside the points

## 2.2 Result on Real Data

The surface reconstruction has problems when the data is not dense enough. In the femur example in Fig. 1 that does not produce any problems. The result can be seen in Fig. 2.

## 3 Finding Corresponding Points

In shape modeling it is of great importance that during the training a dense correspondence is established over the training set. This part is the most difficult and the most important for a good result of the upcoming segmentation.

### 3.1 Iterative Closest Point

In this paper the correspondence of points over the training set is established by the Iterative Closest Point (ICP) algorithm [6]. With the ICP algorithm a corresponding triangulation of the structures is achieved over the training set.

The ICP algorithm matches two overlapping surfaces. It uses one as source surface and one as target. The triangulation of the source is kept and the aim is to get an optimal corresponding triangulation on the target surface. To do this an iterative process is applied with the steps as follows:

1. For each vertex at the source surface find the closest point at the target surface.
2. Compute the similarity transformations from the source to the new points, located in the previous step, that minimise the mean square error between the two point clouds with translation and rotation.
3. Apply the transformation
4. Return to 1 until the improvement between two iterations is less than a threshold value  $\tau > 0$ .

When the threshold value is reached the closest points on the target surface are calculated one last time and these points give the new vertices on the target surface.

This algorithm gives two surfaces with corresponding triangulation and each point can be looked at as a landmark with a corresponding landmark at the other surface. If the same source surface is always used and the target surface is switched it is possible to find corresponding landmarks in a larger training set.

## 4 Aligning the Training Set Using Procrustes Analysis

When the corresponding landmarks are found the next step is to align the landmarks under similarity transformations. This is done because only the shape should be considered in the shape model and the translation, scale and rotation should be filtered out.

Alignment of two shapes,  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  and  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ , in three dimensions can be calculated explicitly. Umeyama presents a way to do this [7].

In this paper not only two data sets but the whole training set is to be aligned. Therefore an iterative approach proposed by Cootes et al. [1] has been used.

When the corresponding points are aligned it is possible to move forward and calculate a shape model of the knee.

## 5 Building the Shape Model

With  $n$  landmarks,  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$  where  $\mathbf{x}_i$  are  $m$ -dimensional points, at the surface. The segmentation problem is  $nm$  dimensional. It is therefore of great interest to reduce the dimension and in an accurate way be able to decide whether a new shape is reasonable.

The aim is to find a model so that new shapes can be expressed by the linear model  $\mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{b}$ , where  $\mathbf{b}$  is a vector of parameters for the shape modes. With this approach it is possible to constrain the parameters in  $\mathbf{b}$  so the new shape always will be accurate.

To generate the model  $\Phi$  from  $N$  training shapes, Principal Component Analysis (PCA) is applied [1].

### 5.1 Constructing New Shapes from the Model

From the model new shapes can be constructed. Let  $\Phi = [\Phi_1, \dots, \Phi_N]$ , where  $\Phi_i$  are the eigenvectors of the covariance matrix used in the PCA. New shapes can now be calculated as

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} + \Phi \mathbf{b} = \bar{\mathbf{x}} + \sum_{i=1}^N \Phi_i b_i. \quad (11)$$

Cootes et al. propose in [1] a constraint of the  $b_i$  parameters of  $\pm 3\lambda_i$ , where  $\lambda_i$  is the square root of the eigenvalues,  $\sigma_i$ , of the covariance matrix, to ensure that any new shape is similar to the shapes in the training set. This method is used in this paper.

Another way to constrain the parameters in the shape model would be to look at the probability that the new shape is from the training set and constrain the whole shape into a reasonable interval.

It is not necessary to choose all  $\Phi_i$ , but if not all are used those corresponding to the largest eigenvalues are to be chosen. The numbers of shape modes to be used in the shape reconstruction can be chosen to represent a proportion of the variation in the training set. The proportion of variation that  $t$  shape modes cover are given by

$$V_t = \frac{\sum_{i=1}^t \sigma_i}{\sum \sigma_i}. \quad (12)$$

## 6 Segmentation with Active Shape Models

The segmentation with active shape models is based on an iterative approach. After an initial guess the four steps below are iterated.

1. Search in the direction of the normal from every landmark to find a suitable point to place the landmark in.
2. Update the parameters for translation, rotation, scale and shape modes to make the best fit to the new points.
3. Apply constraints on the parameters.
4. Repeat until convergence.

### 6.1 Multi-resolution Approach for Active Shape Models

To improve the robustness and the speed of the algorithm a multi-resolution approach is used. The idea of multi-resolution is to first search in a coarser image and then change to a more high resolution image when the search in the first image is not expected to improve. This improves the robustness because the amount of noise is less at the coarse level and therefore it is easier to find a way to the right object. The high resolution images are then used to find small structures. The speed increases because there is less data to handle at the coarse levels.

### 6.2 Getting the Initial Guess

In order to obtain a fast and robust segmentation it is important to have a good initial estimation of the position and orientation. In the initial guess the shape is assumed to have the mean shape. This makes it necessary to find values of seven parameters to make a suitable initial guess in three dimensions (three for translation, one for scale and three for the rotation). The method to find the initial guess is usually application dependent.

### 6.3 Finding Suitable Points

To find the new point to place a landmark in, while searching in the directions of the normal, models of the variations of appearance for a specific landmark  $l$  is build. Sample points in the normal direction of the surface are evaluated, this gives  $2k + 1$  equidistant points. These values usually have a big variation of intensity over the training set. To minimise this effect the derivative of the intensity is used. The sampled derivatives are put in a vector  $\mathbf{g}_i$ . These values are then normalised by dividing with the sum of absolute values of the vector.

This is repeated for all surfaces in the training set and gives a set of samples  $\{\mathbf{g}_i\}$  for each landmark. These are assumed to be Gaussian distributed and the mean  $\bar{\mathbf{g}}$  and the covariance  $\mathbf{S}_g$  are calculated. This results in a statistical model of the grey level profile at each landmark.

Through the process from manual marking of the interesting parts to building the triangulation with corresponding landmarks of the object small errors in

the surface position will probably be introduced. This will make the modeled surface to not be exactly coincide to the real surface. Thus the profiles will be translated a bit and the benefit of the profile model will be small. To reduce these problems an edge detection in a short distance along the normal to the surface is performed. If the edge detection finds a suitable edge the landmarks are moved to that position.

**Getting New Points.** When a new point is to be located, while searching in the direction of the normal during segmentation, the quality of the fit is measured by the Mahalanobis distances given by

$$f(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T \mathbf{S}_g^{-1} (\mathbf{g}_s - \bar{\mathbf{g}}), \quad (13)$$

where  $\mathbf{g}_s$  is the sample made around the new point candidate. This value is linearly related to the probability that  $\mathbf{g}_s$  is drawn from the model. Thus minimising  $f(\mathbf{g}_s)$  is the same as maximising the probability that  $\mathbf{g}_s$  comes from the distribution and therefore that the point is at the sought-after edge.

To speed up the algorithm only a few of the landmarks are used at the coarse levels. 1/4 of the landmarks were kept for every step to a coarser level.

## 6.4 Updating Parameters

When new landmark positions are located the next step is to update the parameters for translation, scale, rotation and shape modes to best fit the new points. This is done by an iterative process. The aim is to minimise

$$\|\mathbf{Y} - T_{t,s,\theta}(\bar{\mathbf{x}} + \Phi \mathbf{b})\|^2, \quad (14)$$

where  $\mathbf{Y}$  is the new points and  $T$  is a similarity transformation. The iterative approach is as follows the one presented by Cootes et al. [1].

In the segmentation only shapes relatively similar to the shapes in the training set are of interest. Therefore constraints are applied to the  $\mathbf{b}$  parameters. Usually those constraints are  $\pm 3\sqrt{\sigma_i}$  where  $\sigma_i$  is the eigenvalue corresponding to shape mode  $i$ .

## 7 Experiments

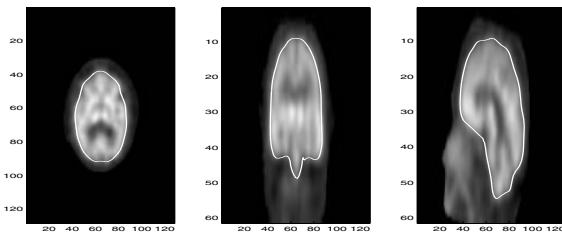
The algorithm was used on two data sets, MR images of the knee and SPECT images of the brain.

### 7.1 Segmentation

The result of the segmentation showed difference between the MR images and the SPECT images. The result was better on the SPECT images.



**Fig. 3.** The result of the segmentation when the model of the gray level structure were used. The segmentation was applied in sagittal images (left) and the result looks better in the sagittal view than in the coronal view (right)



**Fig. 4.** The result of the segmentation on SPECT images of the brain

**Results for MR Images of the Knee.** When the initial guess was not good enough the model was not able to find the way to the femur. Instead other edges were located that were of no interest (often the edge of the image).

If the initial guess was good enough the search algorithm found the right edges almost every time. But in some parts of the images the result was not as good. During the segmentation only the sagittal images were used and if the result was visually examined the result looked better in the sagittal view. In Fig. 3 the result from a segmentation is viewed.

**Results for SPECT Images of the Brain.** When the segmentation was done on the SPECT images a better result was obtained. When the algorithm was used on a number of brains and the result was compared to the points marked on the surface it was hard to tell which were the choice of the computer and which were chosen by the expert. In Fig. 4 the result from a segmentation is viewed.

## 8 Summary and Conclusions

In this paper we present a fully automated way to segment three dimensional medical images with active shape models. The algorithm has been tested at MR images of knees and SPECT images of the brain. The results are promising especially in the SPECT images. In the MR images it is harder to find a

good initial guess which makes the result not so good as in the SPECT images. But if the initial guess is good the segmentation algorithm usually gives a good result.

## Acknowledgments

The authors would like to thank Magnus Tägil at the Department of Orthopedics in Lund for providing the MR images of the knee. For the results of the brain images Simon Ristner and Johan Baldetorp is greatly acknowledged. Lars Edenbrandt is thanked for providing the brain images.

This work has been financed by the SSF sponsored project 'Vision in Cognitive Systems' (VISCOS), and the project 'Information Management in Dementias' sponsored by the Knowledge Foundation and UMAS in cooperation with CMI at KI.

## References

1. Cootes, T., Taylor, C.: Statistical models of appearance for computer vision. [http://www.isbe.man.ac.uk/~bim/Models/app\\_models.pdf](http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf) (2004)
2. Zhao, H.K., Osher, S., Merriman, B., Kang, M.: Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding* **80** (2000) 295–314
3. Shi, Y., Karl, W.: Shape reconstruction from unorganized points with a data-driven level set method. In: *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*. IEEE International Conference on. Volume 3. (2004) 13–16
4. Adalsteinsson, D., Sethian, J.: The fast construction of extension velocities in level set methods. *Journal of Computational Physics* **148** (1999) 2–22
5. Zhao, H.K., T., C., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *Journal of Computational Physics* **127** (1996) 179–195
6. Besl, P., McKay, H.: A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **14** (1992) 239–256
7. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **13** (91) 376–380

# A Novel Algorithm for Fitting 3-D Active Appearance Models: Applications to Cardiac MRI Segmentation

Alexander Andreopoulos and John K. Tsotsos

York University, Dept. of Computer Science and Engineering,  
Centre for Vision Research, Toronto Ontario, M3J 1P3, Canada  
`{alekos, tsotsos}@cs.yorku.ca`

**Abstract.** We present an efficient algorithm for fitting three dimensional (3-D) Active Appearance Models (AAMs). We do so, by introducing a 3-D extension of a recently proposed method that is based on the inverse compositional image alignment algorithm. We demonstrate its applicability for the segmentation of the left ventricle in short axis cardiac MRI. We perform experiments to evaluate the speed and segmentation accuracy of our algorithm on a total of 1473 cardiac MR images acquired from 11 patients. The fitting is around 60 times faster than standard Gauss-Newton optimization, with a segmentation accuracy that is as good as, and often better than Gauss-Newton.

## 1 Introduction

Active Appearance Models (AAMs) provide a promising method for the interpretation of medical images [3], [4]. There is interest in the use of 3-D Active Appearance Models for the segmentation of the left ventricle from short axis cardiac MRI [5], due to AAMs' ability to learn the 3-D structure of the heart and not lead to unlikely segmentations [7].

The algorithms described in the literature for fitting AAMs, are either robust but inefficient gradient descent type algorithms, or efficient but ad-hoc algorithms. Fitting AAMs using standard optimization methods is inefficient, due to the high number of parameters needing to be optimized. This problem is exacerbated with 3-D AAMs since such models can use 50-100 parameters. To deal with this, efficient algorithms for fitting AAMs have been developed [4]. Under such formulations, we look for a constant matrix  $\mathbf{R}$  such that if the current fitting error between the AAM and the image is  $\delta\mathbf{t}$ , the update to the AAM parameters is  $\delta\mathbf{p} = \mathbf{R}\delta\mathbf{t}$ . However, the fitting accuracy and convergence rates of such algorithms are often unsatisfactory. Such methods lack a sound theoretical basis since in general there is no reason why the error measure  $\delta\mathbf{t}$  should uniquely identify the update  $\delta\mathbf{p}$  of the parameters.

Recently, a novel algorithm for fitting 2-D AAMs was introduced in [6]. Its applicability was demonstrated on artificial data and for face tracking. However, as it is cited in [6], there was no known way of extending the algorithm to

higher dimensions since a certain argument used in the paper applied only to 2-D similarity transformations. In this paper we present an extension of the algorithm for the fitting of 3-D AAMs on short axis cardiac MRI. By definition, short axis cardiac MR images are such that the long axis of the heart is perpendicular to the acquisition image plane. This means that during the AAM fitting we need to rotate our model only around the long axis of the heart. We take advantage of this fact to design an efficient fitting algorithm.

We perform experiments comparing our algorithm with Gauss-Newton based optimization, which is generally known as one of the most accurate and reliable optimization algorithms for such problems [1]. We observe a 60 fold improvement in the fitting speed, with a segmentation accuracy that is as good - and in many cases better - as brute force Gauss-Newton optimization. Our algorithm's border positioning errors are significantly smaller than the errors reported for other 3-D AAMs [7] which use the constant matrix approach for the fitting.

## 2 3-D AAMs

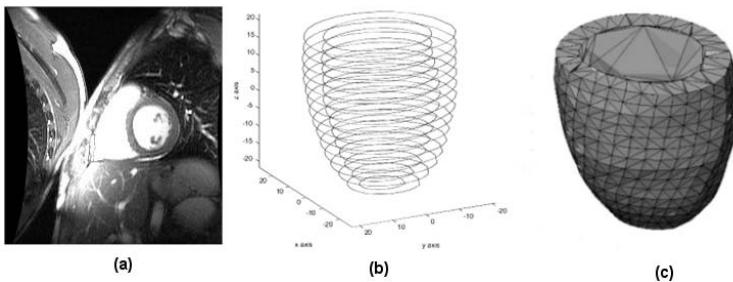
Figure 1(a) shows a short axis cardiac MR image. A stack of such images gives a volumetric representation of the heart. Manual segmentations of the left ventricle provide us contours representing the endocardium and epicardium of the left ventricle. By uniformly sampling each of these contours at  $i_0$  points along their arclength, starting from the closest point to the posterior junction of the left and right ventricles, each contour is represented by a set of landmarks. By stacking the landmarks on top of each other we obtain a 3-D representation of the left ventricle's endocardium and epicardium, as shown in Figure 1(b). However, the number of images intersecting the left ventricle is not the same across every patient. Therefore, we need to interpolate between the contours so that every 3-D model is made up of the same number of slices. If we want to create a slice at height  $z_0$  located between two slices, we can simply do the following: From the line segment joining the  $i^{th}$  landmark in the two slices, we find the location with height  $z_0$ . This gives us the  $i^{th}$  landmark in the new slice. In our implementation, we created 15 contour slices, evenly sampled along the z-axis, located between the apex and basal contours.

If we have a set of  $N$  sample shapes, and each sample consists of  $m$  landmarks, we can represent each shape sample as a  $3m$  dimensional vector. By applying principal component analysis (PCA) on the distribution of the shape vectors, any shape  $\mathbf{s}$  out of the  $N$  shapes can be approximated as

$$\mathbf{s} \approx \mathbf{s}_0 + \sum_{i=1}^n p_i \mathbf{s}_i \quad (1)$$

for some  $\mathbf{p} = (p_1, \dots, p_n) \in \Re^n$ , where  $\mathbf{s}_0$  is the mean shape vector (a.k.a base mesh), and  $\mathbf{s}_i$  indicates the  $i^{th}$  eigenvector. We are summing over  $n$  eigenvectors  $\mathbf{s}_i$  such that they explain around 90% – 95% of the shape variation.

When building a 2-D AAM, we need to make sure that the shapes are aligned with each other so that we remove any difference between two shapes due to a similarity transform. See [3] for more details. This leads to more compact AAMs, which can be described by a smaller number of parameters. In [6] and in our algorithm, this is a necessary step since, by removing the similarity transform from the training set shapes, the  $\mathbf{s}_i$  vectors will be orthogonal to a subspace of the 3-D similarity transforms of  $\mathbf{s}_0$ . In our algorithm, this was accomplished by using an iterative alignment procedure as described in [3], only that in this case we aligned the shapes with respect to translation, scaling and rotation around only the z-axis. We did not align the shapes with respect to x and y axis rotations since we only wanted our model to handle rotations around the z-axis.



**Fig. 1.** (a)Short axis cardiac MRI. (b)Endocardial and epicardial landmarks stacked on each other. Shown as curves for greater clarity. The mean of these landmarks is given by vector  $\mathbf{s}_0$ . (c)Tetrahedrization of  $\mathbf{s}_0$ . Each tetrahedron represents a part of the myocardial muscle or a part of the left ventricle's blood pool

We need to model the appearance variation of the shape. We first manually tetrahedrize  $\mathbf{s}_0$ , as shown in Figure 1(c). This splits the left ventricular volume enclosed by  $\mathbf{s}_0$  into tetrahedra. The same landmark connectivity defining the tetrahedra of  $\mathbf{s}_0$  can be used to define the tetrahedrization of any shape variation resulting from Eq. (1). Then, we use these tetrahedra to sample the appearance enclosed by each training shape [7]. Let the mean appearance we get by averaging the sampled appearances be  $A_0(\mathbf{x})$  and the  $k$  eigenvectors we found by PCA, describing around 90%-95% of the appearance variation, be  $A_1(\mathbf{x}), A_2(\mathbf{x}), \dots, A_k(\mathbf{x})$  (where  $\mathbf{x}$  denotes the appearance coordinate in the base model  $\mathbf{s}_0$  coordinate system). For different  $b_i$  values

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^k b_i A_i(\mathbf{x}) \quad (2)$$

defines the appearance variations the model has learned from our training data.

### 3 The Inverse Compositional Algorithm

There is a wealth of literature on image alignment algorithms and the reader is referred to [1, 2] and the references therein for an overview. The inverse compositional approach has been shown to provide fast and reliable image alignment.

Assume we have a template  $A_0(\mathbf{x})$  that we want to align with an image  $I(\mathbf{x})$ . In the compositional framework for image alignment we compute a warp  $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$  that is composed with the current warp  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  (where  $\mathbf{p}$  are warp parameters and  $\mathbf{x}$  denotes pixel/voxel coordinates), in order to find the update parameters  $\Delta\mathbf{p}$  minimizing

$$\sum_{\mathbf{x} \in \text{Domain}(A_0)} [I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}); \mathbf{p})) - A_0(\mathbf{x})]^2. \quad (3)$$

In the inverse compositional approach we are trying to minimize

$$\sum_{\mathbf{x} \in \text{Domain}(A_0(W))} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}))]^2. \quad (4)$$

The solution to this least squares problem is approximately [1]:

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x} \in \text{Domain}(A_0)} [\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{x})] \quad (5)$$

where

$$\mathbf{H} = \sum_{\mathbf{x} \in \text{Domain}(A_0)} [\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}] \quad (6)$$

and  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$  is evaluated at  $(\mathbf{x}; \mathbf{0})$ . It can be shown that within first order, under the condition of a fairly “smooth” warping function  $\mathbf{W}$ , (4) is equal to

$$\sum_{\mathbf{x} \in \text{Domain}(A_0)} [I(\mathbf{W}(\mathbf{W}^{-1}(\mathbf{x}; \Delta\mathbf{p}); \mathbf{p})) - A_0(\mathbf{x})]^2. \quad (7)$$

This means that the  $\Delta\mathbf{p}$  in (5) can also be used to minimize (7). Notice that (7) is an image alignment error measure of the form (3). So once we have found  $\Delta\mathbf{p}$  we can update the warp by

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}^{-1}(\mathbf{x}; \Delta\mathbf{p}) \quad (8)$$

and go to Eq. (5) to perform another iteration of the image alignment algorithm. This is a very efficient algorithm, known as the inverse compositional algorithm [2]. It is efficient because we can precompute  $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$  and  $\mathbf{H}$ . This algorithm can easily be used for fitting templates  $A(\mathbf{x})$  as described in section 2. In that case, if in Eq. (4) above,  $A_0(\mathbf{x})$  is replaced by  $A(\mathbf{x})$ , with appearance variation as described in section 2, then within first order the error is equal to [2]:

$$\sum_{\mathbf{x} \in \text{Domain}(A_0)} [\text{proj}_{\text{span}(A_i)^\perp} (I(\mathbf{W}(\mathbf{W}^{-1}(\mathbf{x}; \Delta\mathbf{p}); \mathbf{p})) - A_0(\mathbf{x}))]^2 \quad (9)$$

where  $\text{span}(A_i)^\perp$  denotes the space orthogonal to  $A_1, \dots, A_k$ . We minimize expression (9) by using expressions (5), (6), only that now we project each column of  $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$  onto  $\text{span}(A_i)^\perp$  and use this projected matrix instead of  $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ .

## 4 Inverse Compositional Fitting of 3-D AAMs

In this section we show how to fit 3-D AAMs using the inverse compositional approach. In section 4.1 we show how to extend [6] to fit 3-D AAMs with no global similarity transformations. In section 4.2 we show how to fit a 3-D AAM when we allow translations, rotations around only one axis - by convention we will be rotating around the z-axis - and scaling of the coordinate axes.

### 4.1 Fitting Without Global Shape Transform

We now show how the inverse compositional algorithm can be used for fitting 3-D AAMs without taking care of any global shape similarity transformations (translation, scalings and rotations).

We first need to define  $\mathbf{W}(\mathbf{x}; \mathbf{p})$ , where  $\mathbf{p}$  denotes the current landmarks model parameters from Eq. (1), and the  $\mathbf{x} = (x, y, z)^T$  parameter denotes a point in the base mesh  $\mathbf{s}_0$ . Then  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  denotes the warping of  $\mathbf{x}$  under the current warp parameters  $\mathbf{p}$ . As mentioned above, every base mesh voxel  $\mathbf{x}$  lies in a tetrahedron  $\mathbf{T}_0$  defined by the vertices  $(x_i^0, y_i^0, z_i^0)$ ,  $(x_j^0, y_j^0, z_j^0)$ ,  $(x_k^0, y_k^0, z_k^0)$ ,  $(x_l^0, y_l^0, z_l^0)$ . If the current shape parameters of the model are  $\mathbf{p}$ , then let the vertices of the deformed tetrahedron  $\mathbf{T}_1$  be  $(x_i, y_i, z_i)$ ,  $(x_j, y_j, z_j)$ ,  $(x_k, y_k, z_k)$ ,  $(x_l, y_l, z_l)$  which were computed from Eq. (1).  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  computes the affine transformation of  $\mathbf{x}$  from  $\mathbf{T}_0$  to  $\mathbf{T}_1$ . If  $\alpha_i, \alpha_j, \alpha_k, \alpha_l$  denote the barycentric coordinates of  $\mathbf{x}$  in  $\mathbf{T}_0$  given by

$$\begin{pmatrix} \alpha_i \\ \alpha_j \\ \alpha_k \\ \alpha_l \end{pmatrix} = \begin{pmatrix} x_i^0 & x_j^0 & x_k^0 & x_l^0 \\ y_i^0 & y_j^0 & y_k^0 & y_l^0 \\ z_i^0 & z_j^0 & z_k^0 & z_l^0 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (10)$$

(by the definition of barycentric coordinates  $\alpha_i + \alpha_j + \alpha_k + \alpha_l = 1$  and  $0 \leq \alpha_i, \alpha_j, \alpha_k, \alpha_l \leq 1$ ), the affine transformation of  $\mathbf{x}$  is  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  and is given by:

$$\alpha_i(x_i, y_i, z_i)^T + \alpha_j(x_j, y_j, z_j)^T + \alpha_k(x_k, y_k, z_k)^T + \alpha_l(x_l, y_l, z_l)^T \quad (11)$$

To compute  $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$  in Eq. (5) we do the following: For every point  $\mathbf{x}$  in the mean tetrahedrization  $\mathbf{s}_0$  compute  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  and sample image  $I$  at that location by trilinear interpolation. By a straightforward extension of [6] from 2-D to 3-D,

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \sum_{i=1}^m \left[ \frac{\partial \mathbf{W}}{\partial x_i} \frac{\partial x_i}{\partial \mathbf{p}} + \frac{\partial \mathbf{W}}{\partial y_i} \frac{\partial y_i}{\partial \mathbf{p}} + \frac{\partial \mathbf{W}}{\partial z_i} \frac{\partial z_i}{\partial \mathbf{p}} \right] \quad (12)$$

where  $\frac{\partial \mathbf{W}}{\partial x_i} = (\alpha_i, 0, 0)^T \pi(\mathbf{x}, i)$ ,  $\frac{\partial \mathbf{W}}{\partial y_i} = (0, \alpha_i, 0)^T \pi(\mathbf{x}, i)$ ,  $\frac{\partial \mathbf{W}}{\partial z_i} = (0, 0, \alpha_i)^T \pi(\mathbf{x}, i)$  and  $\frac{\partial x_i}{\partial \mathbf{p}} = (\mathbf{s}_1^{x_i}, \mathbf{s}_2^{x_i}, \dots, \mathbf{s}_n^{x_i})$ ,  $\frac{\partial y_i}{\partial \mathbf{p}} = (\mathbf{s}_1^{y_i}, \mathbf{s}_2^{y_i}, \dots, \mathbf{s}_n^{y_i})$ ,  $\frac{\partial z_i}{\partial \mathbf{p}} = (\mathbf{s}_1^{z_i}, \mathbf{s}_2^{z_i}, \dots, \mathbf{s}_n^{z_i})$ .  $\pi(\mathbf{x}, i)$  equals 1 if  $\mathbf{x}$  is in a tetrahedron of  $\mathbf{s}_0$  having landmark  $i$  as its vertex, and is 0 otherwise.  $\mathbf{s}_j^{x_i}, \mathbf{s}_j^{y_i}, \mathbf{s}_j^{z_i}$  denote the element of  $\mathbf{s}_j$  corresponding to  $x_i, y_i$  and  $z_i$  respectively. The summation in (12) is nonzero only for the 4 vertices of the tetrahedron enclosing the point  $\mathbf{x}$  where we are evaluating the jacobian.

By the argument in [6], within first order  $\mathbf{W}^{-1}(\mathbf{x}; \Delta \mathbf{p}) = \mathbf{W}(\mathbf{x}; -\Delta \mathbf{p})$ . From (8) we conclude that we need to find a parameter  $\mathbf{p}'$  such that  $\mathbf{W}(\mathbf{x}; \mathbf{p}') = \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}^{-1}(\mathbf{x}; \Delta \mathbf{p})$ . We can approximate this quantity by finding a  $\mathbf{p}''$  such that  $\mathbf{W}(\mathbf{x}; \mathbf{p}'') = \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; -\Delta \mathbf{p})$ . The problem here is that piecewise affine warping does not form a group under the operation of composition. In other words the composition of two piecewise affine warps cannot necessarily be described by another piecewise affine warp. We compensate for this by estimating a new position for the landmarks  $\mathbf{s}_0$  under the composition of the two warps and once we have done this for all landmarks in  $\mathbf{s}_0$ , we estimate  $\mathbf{p}''$  by finding the closest vector  $\mathbf{p}$  in Eq. (1) satisfying the new landmarks and letting  $\mathbf{p}''=\mathbf{p}$ .

We estimate a new position for landmarks  $\mathbf{s}_0$  by the following method. For every landmark  $\mathbf{a}$  in vector  $\mathbf{s}_0$  we estimate  $\mathbf{W}(\mathbf{a}; -\Delta \mathbf{p})$  by using Eq. (1) with  $-\Delta \mathbf{p}$  as parameter. Then, to compute  $\mathbf{W}(\mathbf{W}(\mathbf{a}; -\Delta \mathbf{p}); \mathbf{p})$  we use the following heuristic procedure, which gives good results for the 2-D case in [6] and our 3-D case. For each one of the tetrahedra in  $\mathbf{s}_0$  having  $\mathbf{a}$  as a vertex, we estimate the destination of  $\mathbf{W}(\mathbf{a}; -\Delta \mathbf{p})$  under that tetrahedron's affine warp and we define the value of  $\mathbf{W}(\mathbf{W}(\mathbf{a}; -\Delta \mathbf{p}); \mathbf{p})$  to be the average value of the destination of  $\mathbf{W}(\mathbf{a}; -\Delta \mathbf{p})$  under the affine warps of those tetrahedra.

## 4.2 Fitting Without X and Y Axes Rotations

Let  $\mathbf{x} = (x, y, z)^T$  and  $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6) = (\frac{a}{c_1}, \frac{b}{c_2}, \frac{c}{c_3}, \frac{t_x}{c_4}, \frac{t_y}{c_5}, \frac{t_z}{c_6})$ , where the  $c_i$  are constants that will be defined later. Assume  $\mathbf{N}(\mathbf{x}; \mathbf{q})$  is a global transform, which does not rotate the shape around the x and y axes, defined as follows:

$$\mathbf{N}(\mathbf{x}; \mathbf{q}) = \begin{pmatrix} 1+a & -b & 0 \\ b & 1+a & 0 \\ 0 & 0 & 1+c \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}. \quad (13)$$

Notice that  $\mathbf{q} = \mathbf{0}$  gives the identity transformation. If we let  $a = k \cos(\theta) - 1$ ,  $b = k \sin(\theta)$  and  $c = s - 1$  then

$$\mathbf{N}(\mathbf{x}; \mathbf{q}) = \begin{pmatrix} k \cos(\theta) & -k \sin(\theta) & 0 \\ k \sin(\theta) & k \cos(\theta) & 0 \\ 0 & 0 & s \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}. \quad (14)$$

This performs a rotation by an angle  $\theta$  around the z-axis followed by a scaling by  $k$  of the x,y coordinates and a scaling by  $s$  of the z-coordinates, followed by a translation by  $(t_x, t_y, t_z)^T$ . If we replace the  $s$  above by  $k$ , then we are performing a three-dimensional similarity transform where we are not rotating around the x,y axes. In other words Eq. (13) above has slightly more expressive power than

a typical similarity transform since it allows to scale the z-coordinate values by a value different than the scaling of the x,y coordinates. Then  $\mathbf{N} \circ \mathbf{W}$  performs both the piecewise affine warp of section 4.1 and the global transform  $\mathbf{N}$ .

Some may argue against scaling along the z axis independently from the x,y axes scalings, since this is not a similarity transform. But from the test cases we performed later on, this does not lead to incorrect model instances, and instead adds more expressive power to our model, making it easier to fit heart models with different z-axes scalings than the ones in our training set.

As noted above, our base mesh is  $\mathbf{s}_0 = (x_1^0, y_1^0, z_1^0, \dots, x_m^0, y_m^0, z_m^0)^T$ . Let  $\mathbf{s}_1^* = c_1(x_1^0, y_1^0, 0, \dots, x_m^0, y_m^0, 0)^T$ ,  $\mathbf{s}_2^* = c_2(-y_1^0, x_1^0, 0, \dots, -y_m^0, x_m^0, 0)^T$ ,  $\mathbf{s}_3^* = c_3(0, 0, z_1^0, \dots, 0, 0, z_m^0)^T$ ,  $\mathbf{s}_4^* = c_4(1, 0, 0, \dots, 1, 0, 0)^T$ ,  $\mathbf{s}_5^* = c_5(0, 1, 0, \dots, 0, 1, 0)^T$ ,  $\mathbf{s}_6^* = c_6(0, 0, 1, \dots, 0, 0, 1)^T$  where  $c_i$  is a constant such that  $\mathbf{s}_i^*$  is of unit length. Then

$$\mathbf{N}(\mathbf{s}_0; \mathbf{q}) = \mathbf{s}_0 + \sum_{i=1}^6 q_i \mathbf{s}_i^* \quad (15)$$

where  $q_1 = \frac{a}{c_1}$ ,  $q_2 = \frac{b}{c_2}$ ,  $q_3 = \frac{c}{c_3}$ ,  $q_4 = \frac{t_x}{c_4}$ ,  $q_5 = \frac{t_y}{c_5}$ ,  $q_6 = \frac{t_z}{c_6}$ . If during the shape alignment we aligned the training data such that their center was at point (0,0,0) then  $S^* = \{\mathbf{s}_1^*, \mathbf{s}_2^*, \mathbf{s}_3^*, \mathbf{s}_4^*, \mathbf{s}_5^*, \mathbf{s}_6^*\}$  is an orthonormal set.

The set  $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$  from Eq. (1) is the set of eigenvectors of the covariance matrix. For reasons which will become obvious later, we must make sure that every vector in  $S^*$  is orthogonal to every vector in  $S$ . Because  $S^*$  is not a similarity transform, and due to various sources of error, the alignment procedure we performed earlier might not make the two sets fully orthogonal. We therefore have to orthogonalize the two sets  $S^*$  and  $S$ . In our test cases we did it as follows. For every vector in  $\mathbf{v} \in S$  we take its projection  $\mathbf{v}'$  onto the space orthogonal to  $S^*$  by

$$\mathbf{v}' = \mathbf{v} - \sum_{\mathbf{v}^* \in S^*} (\mathbf{v}^T \mathbf{v}^*) \mathbf{v}^* \quad (16)$$

We call this new set  $S'$ . Then, by using an orthogonalization algorithm such as the Gram-Schmidt algorithm we can transform  $S'$  into  $S''$ , where now  $S''$  is an orthonormal set. Then every vector in  $S^*$  is orthogonal to every vector in  $S''$ .

We need to show what is the Jacobian of  $\mathbf{N} \circ \mathbf{W}$  which will be used instead of  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$  in Eq. (5)-(6), and how to update the parameters from Eq. (7). From [6] it is seen that the jacobian of  $\mathbf{N} \circ \mathbf{W}$  is  $(\frac{\partial \mathbf{N} \circ \mathbf{W}}{\partial \mathbf{q}}, \frac{\partial \mathbf{N} \circ \mathbf{W}}{\partial \mathbf{p}}) = (\frac{\partial \mathbf{N}}{\partial \mathbf{q}}, \frac{\partial \mathbf{W}}{\partial \mathbf{p}})$ .

In the same way noted in section 4.1, to within first order  $(\mathbf{N} \circ \mathbf{W})^{-1}(\mathbf{x}; \Delta \mathbf{q}, \Delta \mathbf{p}) = \mathbf{N} \circ \mathbf{W}(\mathbf{x}; -\Delta \mathbf{q}, -\Delta \mathbf{p})$ . We use this approximation to define

$$\mathbf{s}^\dagger = \mathbf{N} \circ \mathbf{W}((\mathbf{N} \circ \mathbf{W})^{-1}(\mathbf{s}_0; \Delta \mathbf{q}, \Delta \mathbf{p}); \mathbf{q}, \mathbf{p}) \quad (17)$$

(the new locations of the landmarks  $\mathbf{s}_0$ ) by using the method of section 4.1 for composing two warps. Once we have estimated the new landmark positions  $\mathbf{s}^\dagger$  from Eq. (17), we need to find new values for  $\mathbf{p}$  and  $\mathbf{q}$  such that  $\mathbf{N} \circ \mathbf{W}(\mathbf{s}_0; \mathbf{q}, \mathbf{p}) = \mathbf{s}^\dagger$ . First notice that

$$\mathbf{N} \circ \mathbf{W}(\mathbf{s}_0; \mathbf{q}, \mathbf{p}) = \mathbf{N}(\mathbf{s}_0 + \sum_{i=1}^n p_i \mathbf{s}_i; \mathbf{q}) \quad (18)$$

which can be rewritten as

$$\mathbf{N}(\mathbf{s}_0; \mathbf{q}) + \begin{pmatrix} 1+a & -b & 0 \\ b & 1+a & 0 \\ 0 & 0 & 1+c \end{pmatrix} \sum_{i=1}^n p_i \mathbf{s}_i \quad (19)$$

where the summations above are taking place over all vectors  $\mathbf{s}_i \in S''$ . The matrix multiplication in (19) above with the  $3m$  dimensional vector  $\mathbf{s}_i$ , indicates the result of multiplying every triple of adjacent x,y,z coordinates by the matrix.

By using (19), we see that  $\mathbf{N} \circ \mathbf{W}(\mathbf{s}_0; \mathbf{q}, \mathbf{p}) = \mathbf{s}^\dagger$  can be rewritten as

$$\begin{aligned} \mathbf{s}^\dagger &= \mathbf{s}_0 + \sum_{i=1}^6 q_i \mathbf{s}_i^* + \\ (1+a) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \sum_{i=1}^n p_i \mathbf{s}_i + b \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \sum_{i=1}^n p_i \mathbf{s}_i + (1+c) \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \sum_{i=1}^n p_i \mathbf{s}_i. \end{aligned} \quad (20)$$

The three terms in Eq. (20) above that are multiplied by  $1+a$ ,  $b$  and  $1+c$ , are orthogonal to the vectors in  $S^*$ , since  $S^*$  and  $S''$  are orthogonal. This is most difficult to see for the fourth term in (20) that is multiplied by  $b$ . This term is orthogonal to the vectors in  $S^*$  because for each vector in  $S^*$ , if we switch the values of the  $x$  and  $y$  coordinates and change the sign of one of the two coordinates, the resulting vector still belongs to  $\text{span}(S^*)$ . From (20) we get

$$q_i = \mathbf{s}_i^* \cdot (\mathbf{s}^\dagger - \mathbf{s}_0). \quad (21)$$

which we can use to get

$$p_i = \mathbf{s}_i \cdot (\mathbf{N}^{-1}(\mathbf{s}^\dagger; \mathbf{q}) - \mathbf{s}_0) \quad (22)$$

where

$$\mathbf{N}^{-1}(\mathbf{s}^\dagger; \mathbf{q}) = \begin{pmatrix} 1+a & -b & 0 \\ b & 1+a & 0 \\ 0 & 0 & 1+c \end{pmatrix}^{-1} [\mathbf{s}^\dagger - \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}] \quad (23)$$

and we found the new parameters  $\mathbf{p}$  and  $\mathbf{q}$ . Without the orthonormal set  $S^*$  we introduced, there would be no efficient way of updating the parameters  $\mathbf{p}, \mathbf{q}$ .

## 5 Experimental Results

We trained and fitted the model on a data set of 11 short axis cardiac MR image sequences with a temporal resolution of 20 frames. Each image's resolution was  $256 \times 256$  pixels. For each frame, the number of slices intersecting the left ventricle

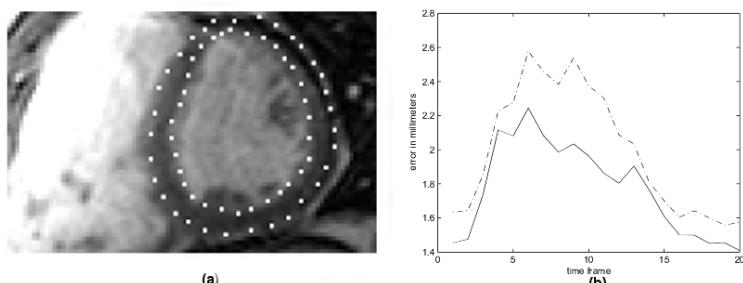
ranged between 6 and 10. All test cases were done on an Intel Xeon 3.06Ghz with 3GB RAM using MATLAB 7.01.

A manual tracing of the left ventricular endocardium and epicardium was acquired from each image where both the left ventricle's endocardium and epicardium was visible. The endocardial contours were drawn behind the papillary muscles and trabeculae.

We trained two models, one using the data from the 10 frames closest to end-diastole and the other model using the remaining 10 frames closest to end-systole, giving us a total of 110 training samples for each of the two models. We used a leave-one-out approach for each model's training: If we wanted to fit an AAM on a certain patient's MR sequence, we trained our two models on the training examples not belonging to that patient, and on each frame we fit the appropriate model. The fitting process stopped if the error change was below a certain threshold, or if the fitting error had stopped decreasing monotonically.

We compared the accuracy of the algorithm described in this paper and of a standard Gauss-Newton optimizer which simultaneously optimized all the model parameters. We compared our algorithm with a Gauss-Newton optimizer for two reasons: Firstly to emphasize the gain in speed that our algorithm can provide compared to brute force approaches. Secondly to investigate our algorithms fitting accuracy against an exhaustively tested algorithm which is known from previous research [6] to outperform the faster (but less reliable) constant matrix approaches described in the introduction.

We manually translated, rotated and scaled  $s_0$  until our model fitted well the median MR slice of the left ventricle, and then ran the algorithms. Once the segmentation was complete, for each of the image slices intersected by our model, we extracted the endocardial and epicardial contours of the model's intersection with the image. A total of 1473 images were segmented by both algorithms. For each one of the landmarks in the extracted contours we estimated the minimum distance to the curve corresponding to the ground truth. The inverse compositional algorithm gave a mean left ventricle endocardial and epicardial error of  $1.6 \pm 1.58\text{mm}$  and  $1.9 \pm 1.7\text{mm}$  and a mean fitting speed of 12.1



**Fig. 2.** (a) Resulting segmentation of an image, shown as white dots. (b) The average error in millimeters for each of the 20 frames in our MR sequences. The solid line is the inverse compositional algorithm's error and the dashed line is the Gauss-Newton error. End-systole is close to frame 10 for the patients in our data set

seconds. The respective measures for Gauss Newton based optimization were  $1.73 \pm 1.61\text{mm}$  and  $2.20 \pm 2.04\text{mm}$  and 688.5 seconds. In Figure 2(b) we show the mean error rates as a function of time frame. The algorithm described in this paper fits the model approximately 60 times faster than a typical brute force Gauss-Newton optimization with a significantly smaller error than standard Gauss-Newton optimization, especially for frames closest to end-systole. We have also experimented with training sets of other sizes, which does lead to significant changes in the error measures we get, but in all cases the inverse compositional algorithm is at least comparable, if not better than Gauss-Newton. We suspect that a reason why Gauss-Newton gives a slightly greater error is because Gauss-Newton has to optimize the appearance parameters also, while the inverse compositional algorithm projects them out. We believe that if both algorithms directly optimized the same number of parameters, Gauss-Newton would be at least as accurate as the inverse compositional algorithm, but would remain much slower. Our models were made up of 30 shape parameters, 20 appearance parameters (that the inverse compositional algorithm projected out) and the 6 global shape parameters  $\mathbf{q}$ , for a total of 56 parameters. The Gauss-Newton based models used 5 global shape parameters (the z-axis scaling was the same as that of the x,y axes) so they used 55 parameters. The inverse compositional algorithm proved to be extremely sensitive to global intensity changes in the images. We handled this by normalizing the left ventricle's intensity in each image to a common mean and standard deviation before training and fitting our models.

## 6 Conclusions

We presented an efficient and robust algorithm for fitting 3-D AAMs on short axis cardiac MRI. It gives rapid segmentation results with a precision that is at least as good as that of previously reported results and brute force Gauss-Newton optimization. In conclusion, the fitting algorithm presented here has given us encouraging results, indicating that it might become the method of choice for fitting 3-D AAMs when rotation around only one axis is needed.

## Acknowledgements

We thank Dr. Paul Babyn and Dr. Shi-Joon Yoo of the Department of Diagnostic Imaging at the Hospital for Sick Children in Toronto for providing us the MRI data. JKT holds a Canada Research Chair in Computational Vision and acknowledges its financial support. AA holds an NSERC PGS-M and acknowledges its financial support. We also thank the reviewers for their comments.

## References

1. S. Baker, R. Goss, and I. Matthews, "Lucas-Kanade 20 Years on: A Unifying Framework," *International Journal of Computer Vision*, Vol. 56, No. 3, pp. 221-255, 2004.
2. S. Baker, and I. Matthews, "Equivalence and Efficiency of Image Alignment Algorithms," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 1090-1097, 2001.
3. T.F. Cootes, "Statistical Models of Appearance for Computer Vision," [Online]. Available: [http://www.isbe.man.ac.uk/~bim/Models/app\\_models.pdf](http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf)
4. T.F. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," In *Proceedings of the European Conference on Computer Vision*, Vol. 2, pp. 484-498. 1998.
5. A.F. Frangi, W.J. Niessen, and M.A. Viergever, "Three-Dimensional Modeling for Functional Analysis of Cardiac Images: A Review," *IEEE Transactions on Medical Imaging*, Vol.20, No. 1, pp. 2-25, 2001.
6. I. Matthews, and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 135-164, 2004.
7. S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "3-D Active Appearance Models: Segmentation of Cardiac MR and Ultrasound Images," *IEEE Transaction on Medical Imaging*, Vol.21, No. 9, pp. 1167-1178, 2002.

# Decision Support System for the Diagnosis of Parkinson's Disease

Anders Ericsson, Markus Nowak Lonsdale, Kalle Astrom,  
Lars Edenbrandt, and Lars Friberg

No Institute Given

**Abstract.** Recently new nuclear medical diagnostic imaging methods have been introduced by which it is possible to diagnose Parkinson's disease, PD, at an early stage. The binding of a specific receptor ligand [ $^{123}\text{I}$ ]-FP-CIT in specific structures of the human brain is visualized by way of single photon emission computerized tomography, SPECT. The interpretation of the SPECT data can be accessed by visual analysis and manual quantification methods. We have developed a computer aided automatic decision support system in attempt to facilitate the diagnosis of Parkinson's disease from the acquired SPECT images. A rigid model was used for the segmentation of the basal ganglia of the human brain. The aim of the study was to develop an automated method for quantification and classification of the images.

The study comprises SPECT scans 89 patients, who underwent a [ $^{123}\text{I}$ ]-FP-CIT examination because of suspected Parkinson's disease. An experienced nuclear medicine physician interpreted the images and diagnosed 65 of the patients as most likely suffering from Parkinson's disease.

The method included the following steps; (i) segmentation of basal ganglia by fitting a rigid 3D model to the image, (ii) extraction of 17 features based on image intensity distribution in the basal ganglia and a reference based on image intensity distribution outside the basal ganglia, (iii) classification using Support Vector Machine (SVM).

The classification based on the automated method showed a true acceptance of 96.9% and a true rejection of 91.6%. The classification based on a manual quantification method gave a true acceptance of 98.5% and a true rejection of 100%. The method proposed here is fully automatic and it makes use of the full 3D data set in contrast to a method that is widely used at hospitals today which only uses a few 2D image slices.

## 1 Introduction

Parkinson's disease (PD) is a neurological movement disorder associated with slow movements, tremor, rigidity and postural instability. The disease can occur before the age of 40, but more commonly it affects people in their 60s. The first symptoms can be very delicate, e.g., slight tremor of one hand. The disease usually progress slowly over years but in the late stages PD patients may become severely affected and wheelchair bound. There is no cure for PD, but patients may benefit from anti-parkinsonian medication.

An early and correct diagnosis of PD is important for the management of patients. Until recently the diagnosis solely was based on clinical assessment and clinical rating scales. The initial diagnoses of PD made by general practitioners have shown to be incorrect in 24% to 35% of the cases [1]. A reliable diagnostic test, which could be used to differentiate between different tremor disorders, would therefore be of great value. PD is caused by degeneration of a part of the brain called substantia nigra. The dopamine producing neurons projects to a nearby structure called the basal ganglia where dopamine is released. The cell membrane of these neurons contain dopamine re-uptake receptor sites (dopamine transporters, DAT). The number of the dopamine producing neurons decrease and the reduction of receptors can be assessed with [<sup>123</sup>I]-FP-CIT, which is a recently introduced diagnostic method [2] [3]. The radiotracer is injected intravenously transported with the blood to the brain where it binds to the pre-synaptic dopamine re-uptake site receptors. After 3 hours when an equilibrium between free and bound receptor ligand is reached 3-dimensional images of the brain are obtained using a gamma camera. The interpretation of the examination is based on both a visual assessment and a quantification of the images in order to assist the physician in the interpretation process. The binding of radiotracers in the basal ganglia and other brain areas can be assessed by manual positioning of regions of interest (ROI). The operator selects from the image a few slices with the highest uptake and the slices are added. Thereafter a set of predefined ROIs is superimposed over the basal ganglia bilaterally and a reference region - usually the visual cortex in the posterior aspect of the brain. The average number of counts of the ROIs are used as an indicator of the number of receptors in different parts of the brain. This manual quantification method has proven effective in more than a 1000 [<sup>123</sup>I]-FP-CIT images that it has been applied to. However, it is time consuming and operator dependent. In addition only a portion of the 3-dimensional volume is considered in the calculations.

The aim of the present study was twofold. First to develop an automated method for quantification and classification of [<sup>123</sup>I]-FP-CIT images. The method should be based on a 3-dimensional analysis. Second to compare the performance of the new method with that of a manual quantification method.

## 2 Material and Method

### 2.1 Material

The material have been [<sup>123</sup>I]-FP-CIT-images obtained at investigations at the Department of Clinical Physiology and Nuclear Medicine at Bispebjerg Hospital. In the database there are images from 89 patients that went through a diagnostic investigation because of suspected PD. The patients had a mean age of 68 years (range 21 - 82). The final diagnosis based on the interpretation by an experienced nuclear medicine physician was PD in 65 cases and non PD in 24 cases.

## 2.2 Method

A dose of 185–225 MBq of [123I]-FP-CIT was injected intravenously and after three hours data acquisition was performed with a three-headed gamma camera (Prism XP3000, Maconi/Philips). Emission data were acquired with a full 360 degree rotation with simultaneously acquisition of transmission data from a 159Gd-source. Image processing was performed using iterative reconstruction with scatter and non-uniform attenuation correction. The iterative algorithm (ordered subsets) OS-EM was used. The number of iterations was 20 for the transmission scans and 4 for the emission data. The chosen number of iterations was based on pilot studies. Resolution in trans-axial plane was 6 mm of the reconstructed slices.

The automated method for quantification and classification of the brain images consists of three main steps: segmentation of the basal ganglia, feature extraction from these regions, and classification of the images. The aim with this work has been to automatically identify images with abnormal receptor radio ligand distribution indicating presences of idiopathic Parkinson's disease or related Parkinson Plus syndromes. In the analysis two main problems was considered: 1) extracting automatically the outline of the basal ganglia and its substructures (caudate nucleus and putamen) and 2) training a classifier on those features.

---

### SEGMENTATION

1. Initiate the model

The first step is to initiate the model of the basal ganglia.

2. Fit the model to data

The model is fitted to data by running an optimisation scheme.

### QUANTIFICATION

3. Measure the intensity distribution

Once the model is fitted, the mean intensity is measured inside four of the regions: left and right putamen and left and right caudatus.

4. Normalisation

To be able to normalise; obtain a reference value by taking the median of the intensity outside the basal ganglia.

5. Retrieve features

Retrieve features from measures.

### CLASSIFICATION

6. Classify using SVM

Classify the features using Support Vector Machines.

Above is a short overview of the method. The main algorithmic development has been done for automatic segmentation of the basal ganglia and automatic feature extraction from those regions. These features are used for the training and classification procedures.

### 2.3 Segmentation

The first step is to segment out the basal ganglia. This is done by optimising the fit of a 3D-model of the basal ganglia.

**(1) Initiate the Model.** Before optimising the model parameters (position and rotation), an initial location of the model must be found. Since the intensity is high in the basal ganglia compared to other brain structure, the centre of mass of the brain corresponds very well to the location of the basal ganglia.

The centre of mass is calculated in the following way: Let  $\mathcal{I}$  be the three dimensional image divided into voxels. Let  $\mathcal{J}$  be the set of indices corresponding to the  $M$  voxels with highest intensity.  $M$  is chosen so that around 20% of the voxels are considered. Then the centre of mass

$\mathbf{C}_m = [x_m, y_m, z_m]$  is retrieved by

$$x_m = \frac{\sum_{i \in \mathcal{J}} \mathcal{I}(i)x(i)}{\sum_{i \in \mathcal{J}} \mathcal{I}(i)}, y_m = \frac{\sum_{i \in \mathcal{J}} \mathcal{I}(i)y(i)}{\sum_{i \in \mathcal{J}} \mathcal{I}(i)}, z_m = \frac{\sum_{i \in \mathcal{J}} \mathcal{I}(i)z(i)}{\sum_{i \in \mathcal{J}} \mathcal{I}(i)},$$

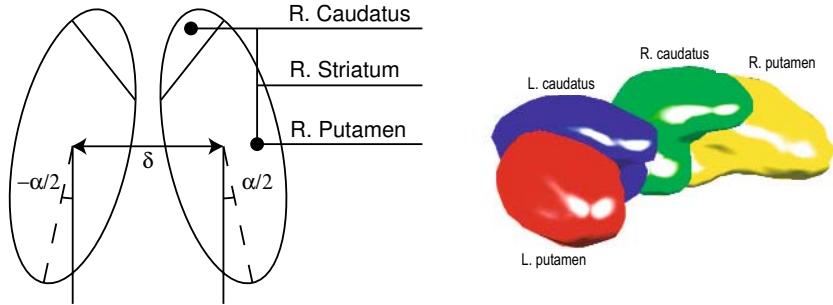
where  $x(i)$  is the x-coordinate of the voxel with index  $i$ .

**(2) Fit the Model to Data.** First a brief explanation of the model and then the optimisation procedure to fit the model to the image.

**The Model.** The model, which consists of two volumes of 3D-voxles, see Figure 1, is rigid, i.e. has no shape parameters. It has eight degrees of freedom. Except for translation and rotation (six degrees) there are two other parameters,  $\alpha$  and  $\delta$ . The distance between the left and the right side is  $\delta$ , and the skewness angle between the left and the right side is  $\alpha$ . The model consists of four regions: left putamen, right putamen, left caudatus and right caudatus, see Figure 1. Each region was defined by a number of 3D-points. These 3D-points were defined from the basal ganglia segmented manually from a 'T1-Single- Subject MR' template distributed with software package SPM'99 (<http://www.fil.ion.ucl.ac.uk/spm>).

Let  $\mathcal{M}_l$  be the set containing the points of the left hand side of the model (left striatum) and let  $\mathcal{M}_r$  be the corresponding right hand side. The operator  $\mathbf{R}_\alpha$  rotates points in a set  $\alpha$  radians clockwise in the z-plane. The operator  $\mathbf{R}_{\theta, \varphi, \psi}$  rotates points in a set and  $\mathbf{t}$  and  $\mathbf{t}_\delta$  are translation vectors. Then the model  $\mathcal{M}(\mathbf{t}, \mathbf{R}, \alpha, \delta)$  is stated

$$\mathcal{M} = \mathbf{R}(\mathbf{R}_{\alpha/2}\mathcal{M}_l - \mathbf{t}_{\delta/2} \cup \mathbf{R}_{-\alpha/2}\mathcal{M}_r + \mathbf{t}_{\delta/2}) + \mathbf{t}. \quad (1)$$



**Fig. 1.** To the left a schematic figure of the model. To the right a 3D-image of the model, where the 3D points are defined from an MR-image

**Optimisation Scheme.** The fitting of the model to data is stated as an optimization problem. Depending on the properties of the isotope (DAT), the intensity is highest at the location of the basal ganglia. The model is therefore fitted to the data by maximizing the intensity contained inside the two volumes. This can be expressed as a constrained-linear maximization problem. The 3D voxel data contains also structures in the lower part of the head below the brain. In severe cases where the intensity of the basal ganglia is relatively low it is important to constrain the search from the lower parts of the data. In some severe cases with very low radioisotope uptake in the basal ganglia the non-specific uptake in the parotid glands becomes very prominent showing a relatively high intensity.

The goal function is chosen to be the integrated intensity inside the two volumes. Let  $\mathcal{M}$  be the model in (1) and  $\mathcal{I}$  is the 3D voxel image, then the goal function  $f$  is defined

$$f(\mathbf{t}, \mathbf{R}, \alpha, \delta) = \sum_{x \in \mathcal{M}(\mathbf{t}, \mathbf{R}, \alpha, \delta)} \mathcal{I}(x) . \quad (2)$$

The optimization problem becomes

$$\max_{(\mathbf{t}, \mathbf{R}, \alpha, \delta)} f(\mathbf{t}, \mathbf{R}, \alpha, \delta) , \quad (3)$$

where

$$\mathbf{t} \geq \mathbf{t}_{min} , \mathbf{t} \leq \mathbf{t}_{max} , \alpha \geq 0 , \alpha \leq \alpha_{max} , \delta \geq \delta_{min} , \delta \leq \delta_{max} .$$

Several different optimization methods could be considered, for example constrained Levenberg-Marquardt. Here steepest descent is used and it turns out to work well. The result is that the model fits to the nearest local maxima from the initialization point. For most cases this is a global maximum. For severe cases of Parkinson the initialization point is important. The optimization might fail for severe cases, but they are easy to interpret, for example by looking at the maximum intensity relative to the mean intensity.

## 2.4 Quantification

**(3) Measure the Mean Intensity.** Once the model has been fitted to data, the mean intensity inside the four regions defined by the model is measured. The sets representing left and right putamina and left and right caudate nuclei are  $\mathcal{M}_{pl}$ ,  $\mathcal{M}_{pr}$ ,  $\mathcal{M}_{cl}$  and  $\mathcal{M}_{cr}$ . The mean intensity inside the left putamen  $p_l$  is retrieved as

$$p_l = \frac{\sum_{x \in \mathcal{M}} \mathcal{I}(x)}{\sum_{x \in \mathcal{M}} 1} . \quad (4)$$

The other regions are treated analogously.

**(4) Normalisation.** The radiotracer binds to the dopamine transporters (re-uptake sites) in the basal ganglia. In addition to this specific binding there is also a non-specific binding in the rest of the brain. In order to determine the specific binding the data for the selected volumes of interest, VOI, should be corrected for non-specific binding. The median count rate outside the basal ganglia, i.e. the rest of the brain, was used as the reference value. This value was used to calculate the specific-to-nonspecific binding ratio.

**(5) Retrieve Features from Measures.** The features that are used are described in Table 1. The mean intensity per voxel for left and right caudate and for left and right putamen is denoted  $c_l$ ,  $c_r$ ,  $p_l$  and  $p_r$  respectively. The reference value is  $m$ .

**Table 1.** Table that explains all features being used

Feature	Expression	Explanation
$x_1$	$c_l - m$	Subtract median radiation
$x_2$	$c_r - m$	"-
$x_3$	$p_l - m$	"-
$x_4$	$p_r - m$	"-
$x_5$	$m$	Median radiation
$x_6$	$x_1/m$	Normalisation
$x_7$	$x_2/m$	"
$x_8$	$x_3/m$	"
$x_9$	$x_4/m$	"
$x_{10}$	$p_r/c_r$	Quotient between right putamen and caudatus
$x_{11}$	$p_l/c_l$	Quotient between left putamen and caudatus
$x_{12}$	$\frac{c_l + p_l - c_r + p_r}{c_l + p_l}$	Quotient between left and right striatum
$x_{13}$	$(x_1 - x_2)/x_1$	Quotient between left and right caudatus
$x_{14}$	$(x_3 - x_4)/x_3$	Quotient between left and right putamen
$x_{15}$		Age
$x_{16}$		Gender

## 2.5 Classification

(6) **Classification Using SVM.** A Support Vector Classifier was chosen to discriminate the groups abnormal images likely to reflect parkinsonism and images from patients less likely to suffer from parkinsonism. SVC is advantageous to use when the underlying distribution is unknown. Ma, Zhao, and Ahalt's Implementation of Support Vector Machines has been used (the OSU-SVM toolbox for MATLAB [4]).

## 2.6 Performance Measure

The Support Vector Machines give for each sample a decision value. The samples are classified to one class or the other depending on the decision value (if it is below or above the threshold). By varying the threshold a receiver operating characteristic (ROC) curve was obtained. The performance of the classification was measured as the area under the ROC curve.

## 3 Results

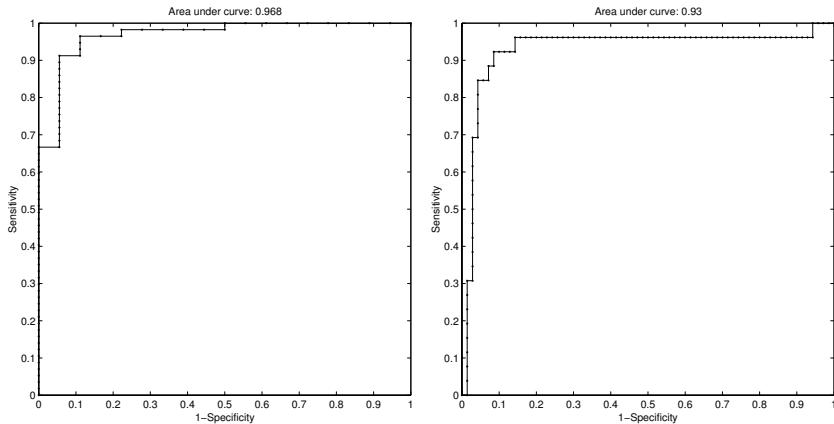
The proposed algorithm has been tested on 89 cases consisting of 65 investigations showing abnormal [<sup>123</sup>I]-FP-CIT distribution (likely parkinsonism) and 24 patients with normal normal [<sup>123</sup>I]- FP-CIT distribution (unlikely parkinsonism). Using the features automatically extracted from these subjects a Support Vector Network has been trained. In Table 2 the results from leave-one-out tests on these 89 cases are given. Also the results from doing the same type of classification using the features of the manual method are presented. Specificity and sensitivity are two terms commonly used in medicine and are other words for true rejection rate and true acceptance rate, respectively.

**Table 2.** The results of leave-one-out tests using different models and under different filtering

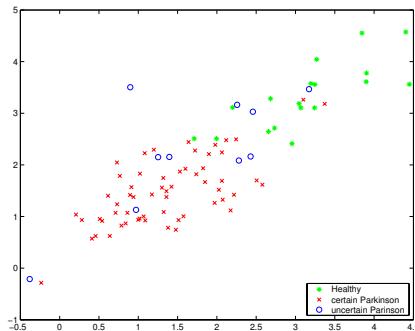
Method	Rate	Specificity	Sensitivity
Automatic method	95.5 %	91.7 %	96.9 %
Semi automatic method	98.9 %	100 %	98.5 %

The performance was also measured from the ROC-curves, where the area under the graph is a measure of performance. In the novel method the area was 0.97 and in the semi-automatic system the area was 0.94, see Figure 2.

To be able to visualize the separation between the two groups, the feature space has been projected onto a two dimensional plane. This is done by selecting the two main axes in the high dimensional feature space. Singular value decomposition (SVD) can be used. If  $F$  is the feature matrix, each column corresponds to one patient. SVD gives,  $F = USV^T$ , where  $SV^T$  is the new coordinates in



**Fig. 2.** The ROC-curve for Algorithm 3 (left) and the ROC-curve for the semi-automatic method (right)



**Fig. 3.** Even when the feature space has been projected into a plane of two dimensions the separation of the groups Parkinson/non Parkinson is visible

the base  $U$ . The first and the second column vector of  $U$  corresponds to the highest and the second highest singular value. Projecting the feature space  $F$  onto the space spanned by  $V$  these two vectors gives the best representation in two dimensions. The coordinates corresponding to these vectors are the two first rows in  $SV^T$ . Even in two dimensions the separation of the groups Parkinson/non Parkinson is visible, see Figure 3.

## 4 Discussion

### 4.1 Segmentation

The segmentation problem was solved by fitting a model by optimizing the sum of the, by the model, integrated intensity. This is generally an appeal-

ing method to segment out regions of interest in SPECT images. It is fast, robust and of low complexity. Since there is little variation in the shape of the basal ganglia among individuals, we believe that a rigid model is sufficient. The shape is, however, interesting and in a future study the information shape can bring will be investigated more thoroughly. The initialization part in the optimization scheme is usually crucial. For most cases the proposed algorithm locates the global maximum. This is due to the fact that the region of the basal ganglia has higher intensity than the rest of the brain. But for severe cases the maximal intensity distribution over the model can be in another part of the brain (normally the lower part) and therefore the initialization is important. The semi-automatic method uses the visual cortex to normalize the measurements. In the proposed method the median of the intensity outside the basal ganglia is used. If there are outliers, the median  $m$  is proved to be more robust than the mean value [5] and motivates our choice of reference value.

## 4.2 Quantification

The features obtained by the proposed automatic method are similar to those obtained by the semi-automatic method. The most important features are the normalized mean intensity of left, right caudate and left, right putamen ( $x_6 - x_9$  in Table 1). Other features of potential interest could include the standard deviation in each region, the minimum and the maximum. The reason for this is that, during the course of PD, a gradual loss of receptors first in the putamen, then the caudate nucleus is observed with considerable anterior-posterior gradient in these regions.

## 4.3 Classification

There are many different classifiers to choose among. Here Support Vector Machine (SVM) [6] was chosen because the underlying distribution is unknown. Another attractive property of a SVM is that it allows many features even if the training material contains few examples. Statistical methods usually require knowledge of the underlying distribution. Other, potentially better classifiers may be available. Different kernels were tested for the SVM. A polynomial kernel of degree 3, a radial based kernel with gamma equal to one, and a linear kernel all gave equally satisfactory results. For the implementations we have used the OSU-SVM toolbox for MATLAB [4]. The SVM gives for each sample a decision value. The samples are classified to one class or the other depending on whether this value is below or above a threshold. By varying this threshold a receiver operating characteristic (ROC) curve was obtained. The performance of the classification was measured as the area under the ROC curve.

## 5 Results and Conclusions

Most parts of the program are implemented in MATLAB code and some in C. Using this non-optimised MATLAB code it takes less than one minute to extract the features and to do the classification (on a Pentium IV 1300 MHz). We have shown the feasibility of an automatic classification of abnormal images from patients likely to suffer from parkinsonism from SPECT data using the [123I]-FP-CIT isotope. To our knowledge this is the first completely automatic system that with a high probability can discriminate between abnormal from abnormal [123I]-FP-CIT SPECT-images. This pilot study of 89 cases also indicates that this fully automatic method performs equally well as the previous semi-automatic system. It has been demonstrated that a completely automated method can be used for the interpretation of [123I]-FP-CIT images.

## Acknowledgements

This work has been financed by the SSF sponsored project 'Vision in Cognitive Systems' (VISCOS).

## References

1. Jankovic, J., A.H., R., M.P., M.: Perl dp. the evolution of diagnosis in early parkinson disease. Parkinson Study Group. *Arch Neurol* **57** (2000) 369–372
2. Booij, J., Tissingh, G., Boer, G., Speelman, J., Stoof, J., Janssen, A.e.a.: 123-i-fpcit spect shows a pronounced decline of striatal dopamine transporter labelling in early and advanced parkinson's disease. *Journal of Neurology, Neurosurgery, and Psychiatry* **62** (1997) 133–140
3. Lokkegaard, A., Werdelin, L., Friberg, L.: Clinical impact of diagnostic spet investigations with a dopamine re-uptake ligand. *Eur.J.Nucl.Med.Mol.Imaging* **29** (2002) 1623–1629
4. Ma, J., Zhao, Y., Ahalt, S.: Osu svm classifier toolbox. (available at [http://eewww.eng.ohio-state.edu/~maj/osu\\_svm](http://eewww.eng.ohio-state.edu/~maj/osu_svm))
5. Rice, J.: Mathematical Statistics and Data Analysis. second edn. Duxbury Press (1994)
6. Cristianini, N., Taylor, J.: An introduction to support vector machines. Cambridge (2000)

# Polygon Mesh Generation of Branching Structures

Jo Skjermo and Ole Christian Eidheim

Norwegian University of Science and Technology,  
Department of Computer and Information Science,  
`Jo.Skjermo@idi.ntnu.no`  
`Ole.Christian.Eidheim@idi.ntnu.no`

**Abstract.** We present a new method for producing locally non-intersecting polygon meshes of naturally branching structures. The generated polygon mesh follows the objects underlying structure as close as possible, while still producing polygon meshes that can be visualized efficiently on commonly available graphic acceleration hardware. A priori knowledge of vascular branching systems is used to derive the polygon mesh generation method. Visualization of the internal liver vessel structures and naturally looking tree stems generated by Lindenmayer-systems is used as examples. The method produce visually convincing polygon meshes that might be used in clinical applications in the future.

## 1 Introduction

Medical imaging through CT, MR, Ultrasound, PET, and other modalities has revolutionized the diagnosis and treatment of numerous diseases. The radiologists and surgeons are presented with images or 3D volumes giving them detailed view of the internal organs of a patient. However, the task of analyzing the data can be time-consuming and error-prone. One such case is liver surgery, where a patient is typically examined using MR or CT scans prior to surgery. In particular, the position of large hepatic vessels must be determined in addition to the relative positions of possible tumors to these vessels.

Surgeons and radiologists will typically base their evaluation on a visual inspection of the 2D slices produced by CT or MR scans. It is difficult, however, to deduce a detailed liver vessel structure from such images. Surgeons at the Intervention Centre at Rikshospitalet in Norway have found 3D renderings of the liver and its internal vessel structure to be a valuable aid in this complex evaluation phase. Currently, these renderings are based on a largely manual segmentation of the liver vessels, so we have explored a way to extract and visualize the liver vessel structure automatically from MR and CT scans.

The developed procedure is graph based. Each node and connection corresponds to a vessel center and a vessel interconnection respectively. This was done in order to apply knowledge based cost functions to improve the vessel tree structure according to anatomical knowledge. The graph is used to pro-

duce a polygonal mesh that can be visualized using commonly available graphic acceleration hardware.

A problem when generating meshes of branching structures in general, is to get a completely closed mesh that does not intersect itself at the branching points. We build on several previous methods for mesh generation of branching structures, including methods from the field of visualization for generation of meshes of tree trunks. The main function of a tree's trunk can be explained as a liquid transportation system. The selected methods for the mesh generation can therefore be extended by using knowledge of the branching angles in natural systems for fluid transportation. This enables us to generate closed and locally non-intersecting polygon meshes of the vascular branching structures in question.

## 2 Previous Work

In the field of medical computer imagery, visualization of internal branching structures have been handled by volume visualization, as the data often was provided by imaging systems that produced volume data. However, visualization of polygon meshes is highly accelerated on modern commonly available hardware, so we seek methods that can utilize this for our visualization of branching vascular transportation structures.

Several previous works have proposed methods for surface mesh generation of trees that handles branching. We can mention the parametric surfaces used in [1], the key-point interpolation in Oppenheimer [14], the "branching ramiforms" [2] (that was further developed by Hart and Baker in [9] to account for "reaction wood"), and the "refinement by intervals" method [11].

In [12], [17], . . . . . from L-systems was introduced. The algorithm used mesh connection templates for adding new parts of a tree to the mesh model, as L-system productions was selected during the generation phase. The mesh connection templates were produced to give a final mesh of a tree, that could serve as a basis mesh for subdivision. This method could only grow the mesh by rules, and could not make a mesh from a finished data set.

The work most similar to our was the . . . . . algorithm presented in [6], [7]. This algorithm was developed for visualization of vascular branching segments in the liver body, for use in a augmented reality aided surgery system. The algorithm produced meshes that could be used with Catmull-Clark subdivision [3] to increase surface smoothness and vertex resolution.

The . . . . . algorithm defined local coordinate axis in a branching point. The average direction of the incoming and outgoing segments was one axis, and an . . . . . vector generated at the root segment was projected along the cross sections to define another axis (to avoid twist). The child closest to the average direction was connected with quads, at a defined distance. The . . . . . vector defined a square cross section, and four directions, at a branching point. All remaining outgoing segments were classified into one of these directions according to their angle compared with the average direction. The child closest by angle in each

direction was connected with the present tile, and this was recursively repeated for any remaining children.

Furthermore, the algorithm did not include any method for automatic adjustment of the mesh with respect to the areas near forking points, and could produce meshes that intersected locally if not manually tuned. Our method automatically generates meshes without local intersection as long as the underlying structures loosely follows natural branching rules.

### 3 Main Algorithm

The proposed algorithm is loosely based on the algorithm. It also uses knowledge of the branching angles in natural systems for fluid transportation as described in Sect. 3.1.

#### 3.1 Natural Branching Rules

Leonardo Da Vinci presented a rule for estimating the diameter of the segments after a furcation in blood vessels, as stated in [15]. The rule states that the cross-section area of a segment is equal to the combined cross section area of the child segments, as seen in the following section.

$$\pi r_0^2 = \pi r_1^2 + \pi r_2^2 + \dots + \pi r_n^2 \quad (1)$$

A generalization, as seen in [2], was presented by Murray in [13]. Here, the rule has been reduced so that the sum of the diameters of the child segments just after a furcation is equal to the diameter of the parent just before the furcating, where  $d_0$ ,  $d_1$ , and  $d_2$  are the diameters of the parent segment and the child segments, respectively.  $\alpha$  was used to produce different branching.  $\alpha$  values between 2 and 3 are generally suggested for different branching types.

$$d_0^\alpha = d_1^\alpha + d_2^\alpha \quad (2)$$

From this Murray could find several equations for the angle between 2 child branches after a furcation. One is shown in [3], where  $x$  and  $y$  are the angles between the direction of the parent and each of two child segments. As seen from the equation, the angles depend on the diameter of each of the child segments. Murray also showed that the segments with the smallest diameter have the largest angle.

$$\cos(x + y) = \frac{d_0^4 - d_1^4 - d_2^4}{2d_1^2 d_2^2} \quad (3)$$

Thus, we assume that the child segment with the largest diameter after a furcation, will have the smallest angle difference from the direction of the parent segment. This forms the basis for our algorithm, and it will therefore produce meshes that do not locally intersect as long as the underlying branching structure mostly follows these rules that are based on observations from nature. We now show how this is used to produce a polygon mesh of a branching structure.

### 3.2 New Algorithm

The algorithm is based on the extended rule to ensure mesh consistency. It uses data ordered in a DAG (Directed Acyclic Graph) where the nodes contains the data for a segment (direction vector, length and diameter). The segments at the same level in the DAG are sorted by their diameters. A vector is used to define the vertices at each segments start and end position. The vertex pointed to by the vector, is set to be a corner of a square cross section of a segment. The sides of this square defines the directions used in sorting any child segments. The sorting results in four sets of child segments (one for each direction of the square), where the child segments in each set are sorted by the largest diameter.

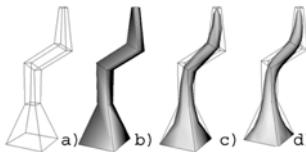
To connect segments, we basically sweep a moving coordinate frame (defined by a projected vector) along a path defined by the segments data. However, at the branching points we must build another type of structure with vertices, so we can add the child segments on to the polygon mesh. This is done by considering the number of child segments, and their diameters and angles compared to the parent segment.

Starting at the root node in the DAG we process more and more child segments onto the polygon mesh recursively. There are four possible methods for building the polygon mesh for any given segment. If there are one or more child segments in the DAG, we must select one of the methods described in Sect. 3.3 (for one child segment), Sect. 3.5 (for more then one child segment), or in Sect. 3.4 (for cases where the child segment with largest diameter has an angle larger then 90 degrees with respect of the parent segment). If there are no child segments, the branch is at its end. The segment is closed with a quad polygon over four vertices generated on a plane defined by the projected vector and the segments diameter at the segments end.

### 3.3 Normal Connection

If there is one child segment (with angle between the present and the child segment less then 90 degrees), we connect the child segment to the present segment, and calculates the vertices, edges and polygons as described in this section. Each segment starts with four vertices on a plane at the segments start position. As the algorithm computes a segment, it finds four vertices at the segment's end. It then closes the sides of the box defined by these eight vertices (not the top and bottom).

The first step is to calculate the average direction between the present segment, and the child segment. This direction is the . Next, the up vector is projected onto the plane defined by the . and the segments end point. A square cross section is then defined on the plane at the segment's end position, oriented to the projected vector to avoid twist. The length of the vector is also changed to compensate for the tilting of this plane compared to the original vector. The corners of the cross section are the four end corner vertices for the present segment. These vertices, along with the four original vertices, defines the box that we close the sides of with quad polygon faces.



**Fig. 1.** Simple mesh production. a) the produced mesh, b) shaded mesh, c) one subdivision, d) two subdivisions

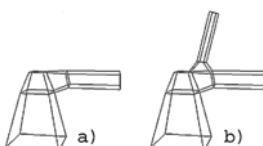
In Fig. 1, the mesh of a stem made of four segments connected in sequence can be seen. After processing the segment, the child segment is set as the present segment.

### 3.4 Connect Backward

When the first child segment direction is larger than 90 degrees compared to the present segments direction, special care has to be taken when producing the mesh (the main part of the structure bends backward). We build the segment out of two parts, where the last part is used to connect the child segments onto the polygon mesh.

The first step is to define two new planes. The  $\nearrow$  is defined along the direction of the segment at the distance that equal to the segments' length, plus the first child's radius (from the segments start position). The  $\nwarrow$  is defined at a distance equal to the diameter of the first child, along the negative of the present segments direction (from the segments end position). Two square cross sections are defined by projecting the  $\nearrow$  vector into the two newly defined planes. The cross section at the segments top can be closed with a quad surface, and the sides between the segments start and the middle cross section can also be closed with polygons. The sides between the middle and the top cross sections that has no child segments in its direction, can also be closed with polygons.

All child segment (even the first one) should be sorted into sets, defined by their direction compared to four direction. The directions are defined by the  $\nearrow$  cross section, and each set should be handled as if they were sets of normal child segments, as described in Sect. 3.5. Vertices from the newly defined  $\nearrow$  and  $\nwarrow$  cross sections are used to define the start cross sections (the four start vertices) for each of the new directions. An example can be seen in Fig. 2.



**Fig. 2.** Mesh production for direction above 90 degrees. a) first child added (direction of 91 degrees), b) next child

### 3.5 Connect Branches

If there is more than one child segment, we start with the first child segment. The first segment has the largest diameter, and should normally have the smallest angle compared to its parent segment.

To connect the first branching child segment onto the mesh, we first use the same method as in section 3.3. We make a square cross section at the end of the present segment, and its sides are closed by polygons. The distance from the segments start to the new cross section can be decreased to get a more accurate polygon mesh (for instance by decreasing the length by half of the calculated  $l + x$  value found later in this section). In the example where vessels in a liver was visualized (Sect. 4.2), the length was decreased as we just described.

A new square cross section is also defined along the . . . . . of the first child, starting at the center of the newly defined cross section. These two new cross sections defines a box, where the sides gives four cross sections (not the top or bottom side). The first child segment (and its children) are recursively added to the top of this box (on the cross section along the . . . . .), while the rest are added to the four sides. Note that the end position of a segment is calculated by vector algebra based on the parent segment's end position, and not on the cross section along the . . . . . This means that the segment's length must be larger than the structure made at the branching point, to add the child.

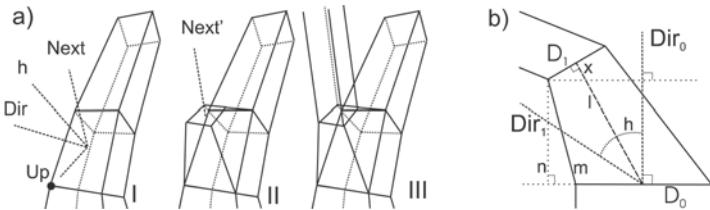
When the recursion returns from adding the main child segment (and its child segments), the remaining child segments are sorted into four sets. The sorting is again done by the segments angle compared to the sides of the cross section around the present segment's end point. One must remember to maintain ordering by diameter while sorting.

The vertices at the corners of the two new cross sections defines a box where the sides can be used as new cross sections for each of the four directions (not the top and bottom sides). For each of the four directions, a new . , vector is defined as the vector between the center of the directions cross section, and a corresponding vertex on the present segment's end cross section. Figure 3 a) shows the . , and . . . . . when adding a child segment in one of the four directions.

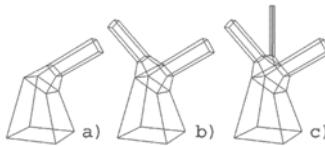
The main problem one must solve is to find the distance along the . . . . . to move before defining the start cross section for the main child segment. This to ensure that there is enough space for any remaining child segments. If the distance moved is too small, the diameter of any remaining child segments will seem to increase significantly after the branching. A too large distance will result in a very large branching structure compared to the segments it connects.

Our main contribution is the automatic estimation of the distance to move, to allow space for any remaining child segments. In Fig. 3 b), we can see the situation for calculating the displacement length . for a given half angle.

The length of . must at least be as large as the root of the  $d_1^2 + d_2^2 \dots + d_n^2$ , where  $d_1, d_2..$  are the diameter of the child segments. This because we know from the . . . . . and murray's findings that every child segment at this point



**Fig. 3.** a) Adding a second child segment. I) new *up*, *direction* and *half direction* vectors for this direction, II) first part of child segment, III) resulting mesh. b) Finding minimum distance  $m$  to move along the half vector to ensure space for any remaining branches (as seen from a right angle)



**Fig. 4.** The mesh production in a branching point. a) First child added, b) next child added, c) third child added

will have equal or smaller diameter than the parent segment (hence the sorting by diameter of child segments). Note that the angle  $h$  will be less or equal to 45 degrees, as any larger angle will lead to the segment being handled as in section 3.4 (as the main angle then will be larger than 90 degrees).

We could find the exact length of  $m$ , but observe that as long as the length of  $n$  is equal to  $d_1$ , we will have enough space along  $m$ . Setting  $n = d_1^2 + d_2^2 + \dots + d_n^2$  gives 4 for calculating the length to move along the segments.

$$l + x = \sqrt{d_1^2 + d_2^2 + \dots + d_n^2} / \cos(h) + \tan(h) * d_1 / 2 \quad (4)$$

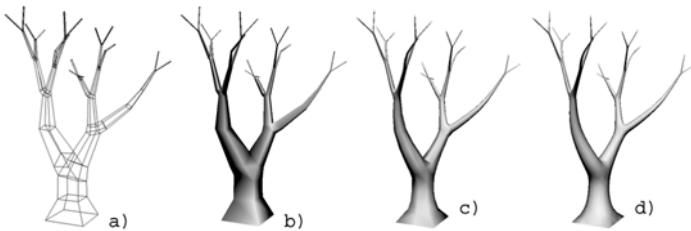
The error added by using  $x + l$  instead of  $m$ , will introduce a small error in the mesh production. However, we observe that setting  $n = d_1$  seems to give adequate results for most cases. An example result from using the algorithm can be seen in Fig. 4.

## 4 The Examples

A preliminary application has been produced in OpenGL to test the algorithm. This section shows this application at work. We have used it to produce polygon meshes for both naturally looking tree stems from a L-system generator, as well as meshes of the derived portal vein from a CT scan of a liver. Normal Catmull-Clark subdivision was used for the subdivision step.

#### 4.1 Lindenmayer Generated Tree Stems

An extension to the application accepted an L-system string, representing a tree stem after a given number of Lindenmayer generation steps, as input. The extension interpreted the L-system string into a DAG that the application used to produce a base polygon mesh from. The application then subdivided this mesh to the level set by the user. An example with a shaded surface can be seen in Fig. 5.



**Fig. 5.** A tree defined by a simple L-system. a) the produced mesh, b) shaded mesh, c) after one subdivision, d) after two subdivisions

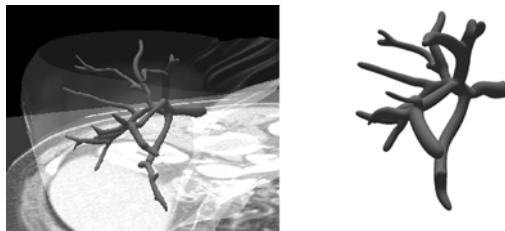
#### 4.2 Delineation of Hepatic Vessels from CT Scans

Several processing steps have to be completed in order to visualize the hepatic vessels from a CT scan. In the preprocessing phase, histogram equalization [8] is first conducted to receive equal contrast on each image in the CT scan. Next, the blood vessels are emphasized using matched filtering [4]. After the preprocessing phase, the blood vessels are segmented using entropy based thresholding [10] and thresholding based on local variance with modifications using mathematical morphology [16]. A prerequisite to our method is that the liver is segmented manually beforehand.

After the vessel segments are found, the vessels' centers and widths must be calculated. These attributes are further used in a graph search to find the most likely vessel structure based on anatomical knowledge. First, the vessel centers are derived from the segmentation result using the segments' skeletons [16]. The vessels' widths are next computed from a modified distance map [8] of the segmented images.

The last step before the vessel graph can be presented is to make connections between the located vessel centers. Centers within segments are interconnected directly. On the other hand, interconnections between adjacent CT slices are not as trivial. Here, as previously mentioned, we use cost functions representing anatomical knowledge in a graph search for the most likely interconnections [5]. The resulting graph is finally visualized using the outlined algorithm in this paper.

A few modification to the existing graph is made in order to make it more visually correct. First, nodes with two or more interconnected neighbors have their heights averaged since the resolution in the y-direction is normally lower



**Fig. 6.** Left: Portal vein visualized from a CT scan of a liver (the CT scan data can be shown at the same time for any part of the liver, for visual comparison). Right: The same structure without the scan data

than that in the image plane. Second, if two interconnected nodes are closer than a predefined limit, the two nodes are replaced by one node positioned between them. Fig. 6 shows the resulting visualization of the derived portal vein from a CT scan of a liver.

## 5 Findings

Our method for automatically calculating the distance for sufficient space for any remaining child segments after the first child segment has been added, seems to produce good results. The preliminary results from our method applied to visualization of hepatic vessels in the liver gives good results when compared with the CT data they are based on, but these results have only been visually verified (however the first feedbacks from the Intervention Centre at Rikshospitalet in Norway has been promising). However, a more throughout comparison with existing methods, and verification against the data set values should be completed before using the method in clinical applications.

The algorithm is fast and simple, and can be used by most modern PC's with a graphic accelerator. The meshing algorithm mostly does its work in real-time, but the subdivision steps and any preprocessing slow things down a bit. Graphic hardware support for subdivision will hopefully be available in the relative near future. When this happens, the subdivision of the branching structures may become a viable approach even for large amounts of trees or blood vessels in real-time computer graphics.

## References

1. Bloomenthal J.: Modeling the mighty maple. Computer Graphics 19, 3 (July 1985) 305–311
2. Bloomenthal J., Wyvill B.: Interactive techniques for implicit modeling. Computer Graphics 24, 2 (Mar. 1990) 109–116
3. Catmull E., Clark J.: Recursively generated b-spline surfaces on arbitrary topological meshes. Computer Aided Design 10, 6 (1978) 350-355

4. Chaudhuri S., Chatterjee S., Katz N., Nelson M., Goldbaum M.: Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imageing* 8, 3 (1989) 263–269
5. Eidheim O. C., Aurdal L., Omholt-Jensen T., Mala T., Edwin B.: Segmentation of liver vessels as seen in mr and ct images. *Computer Assisted Radiology and Surgery* (2004) 201–206
6. Felkel P., Fuhrmann A., Kanitsar A., Wegenkittl R.: Surface reconstruction of the branching vessels for augmented reality aided surgery. *Analysis of Biomedical Signals and Images* 16 (2002) 252-254 (Proc. BIOSIGNAL 2002)
7. Felkel P., Kanitsar A., Fuhrmann A. L., Wegenkittl R.: Surface Models of Tube Trees. *Tech. Rep. TR VRVis 2002 008*, VRVis, 2002.
8. Gonzalez R. C., Woods R. E.: *Digital Image Processing*, second ed. Prentice Hall, 2002.
9. Hart J., Baker B.: Implicit modeling of tree surfaces. *Proc. of Implicit Surfaces 96* (Oct.1996) 143–152
10. Kapur J. N., Sahoo P. K., Wong A. K. C.: A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* 29 (1985) 273–285
11. Lluch J., Vicent M., Fernandez S., Monserrat C., Vivo R.: Modelling of branched structures using a single polygonal mesh. In *Proc. IASTED International Conference on Visualization, Imaging, and Image Processing* (2001).
12. Maierhofer S.: Rule-Based Mesh Growing and Generalized Subdivision Meshes. PhD thesis, Technische Universitaet Wien, Technisch-Naturwissenschaftliche Fakultaet, Institut fuer Computergraphik, 2002.
13. Murray C. D.: A relationship between circumference and weight in trees and its bearing in branching angles. *Journal of General Phylol.* 9 (1927) 725–729
14. Oppenheimer P. E.: Real time design and animation of fractal plants and trees. *Computer Graphics* 20, 4 (Aug. 1986) 55–64
15. Richter J. P.: *The notebooks of Leonardo da Vinci Vol. 1*. Dover Pubns., 1970.
16. Soille P.: *Morphological Image Analysis*. Springer-Verlag, 2003.
17. Tobler R. F., Maierhofer S., Wilkie A.: A multiresolution mesh generation approach for procedural definition of complex geometry. In *Proceedings of the 2002 International Conference on Shape Modelling and Applications (SMI 2002)* (2002) 35–43

# Joint Analysis of Multiple Mammographic Views in CAD Systems for Breast Cancer Detection

Márta Altrichter, Zoltán Ludányi, and Gábor Horváth

Department of Measurement and Information Systems,

Budapest University of Technology and Economics,

P.O.Box 91, H-1521 Budapest, Hungary

[grimma@sch.bme.hu](mailto:grimma@sch.bme.hu), [quad@dpg.hu](mailto:quad@dpg.hu), [horvath@mit.bme.hu](mailto:horvath@mit.bme.hu)

**Abstract.** In screening X-ray mammography two different views are captured and analysed of both breasts. In the four X-ray images some special signs are looked for. First the individual images are analysed independently, but good result can be achieved only if joint analysis of the images is also done. This paper proposes a simple procedure for the joint analysis of the breast's two views. The procedure is based upon the experiences of radiologists: masses and calcifications should emerge on both views, so if no matching is found, the given object is a false positive hit. First a reference system is evolved to find corresponding regions on the two images. Calcification clusters obtained in individual images are matched in "2.5 D" provided by the reference system. Masses detected in individual images are further examined with texture segmentation. The proposed approach can significantly reduce the number of false positive hits both in calcification and in mass detection.<sup>1</sup>

## 1 Introduction

Breast cancer is the most frequent cancer and the leading cause of mortality for women. Evidences show that the chance of survival is significantly increased with early diagnosis and treatment of the disease. X-ray mammography is the only reliable screening method that can be used in masses for early detection. [1] The nationwide screening projects result in enormous amount of mammograms to be analysed in each year. To ease and assist the work of overburdened radiologists computer aided diagnostic (CAD) systems are developed.

During X-ray screening two different views are captured of each breast: a CC (cranio-caudal) from above and a leaning lateral view, the MLO. The two most important symptoms are microcalcifications and masses. Microcalcification clusters have a higher X-ray attenuation than the normal breast tissue and appear as a group of small localized granular bright spots in the mammograms. Masses appear as areas of increased density on mammograms.

<sup>1</sup> This work was supported by National Office for Research and Technology under contract IKTA 102/2001.

Several algorithms searching for pathological abnormalities on a single mammogram were developed within a combined CAD project of Budapest University of Technology and Economics and of Semmelweis Medical University [2], [3]. The main feature of these algorithms is that the positive cases are found with large probability – a hit rate is about 90-95% – but the number of false positive hits per picture is too high. Similar results can be found in the literature like: [4], [5].

A method is needed to decrease the number of false positive hits, which will not or will barely decrease the number of true positive ones. This paper presents a relatively simple new way of this. The method sets off from the fact that the images of calcifications and masses have to appear on both views (MLO and CC). To be more precise they must be on positions of the two views that correspond to each other. In practice a 3-D reconstruction of the breast would be needed. But the full 3-D reconstruction is impossible, because only two views of the breast are available, and because these two views are the “2-D” projections of the differently compressed breast. Therefore instead of a full 3-D reconstruction we suggest a relatively simple procedure which we call “2.5-D” correspondence.

Due to the different characters of the two pathological abnormalities the joint analysis of the two views is done differently for calcifications and masses, although the base of both methods is to restrict the examined picture to a region corresponding to a region on the other view. As tissues with calcifications and normal breast tissues are rather similar in texture for the regions of calcification clusters only the 2.5-D matching method is used. However, for masses texture segmentation could be developed to further refine their assignment.

Boundary detection and image segmentation were necessary preliminary steps of asymmetry analysis and correspondence assay. Although there are many algorithms for detection of edges in images, here – based on its previous successful applications in the field of computer aided diagnostics [6] – Edgeflow algorithm [7] was selected. The results of Edgeflow algorithm were used to find the high intensity gradient of the pectoral muscle (one element of the reference system for the “2.5-D” matching) and at texture segmentation as well.

In Sect. 2. Edgeflow algorithm is briefly described. The building of a reference system is presented in Sect. 3., while the texture segmentation is described in Sect. 5. The use of the reference system on calcification clusters is examined in Sect. 4., while Sect. 6. describes how the reference system and the texture analysis can combine. The performance of the joint analysis is evaluated in Sect. 7., and conclusions are drawn in Sect. 8.

## 2 Edgeflow

Image segmentation is beset with difficulties. In fact edge detection and segmentation are ill-posed problems, since it is usually undefined what we regard as an edge or one segment. The strength of EdgeFlow is that the refinement of the segmentation can be adjusted with a so-called scale parameter ( $\sigma$ ).

The EdgeFlow algorithm is based on a differential filtering when edges are defined as local gradient maxima. The differential filter used here is the derivative of the 2D Gaussian function:  $G_\sigma(x, y)$ .

The filtering results in the edge energy:

$$E(s, \theta) = \left| \frac{\partial}{\partial \vec{n}} \right| = \left| \frac{\partial}{\partial \vec{n}} [I(x, y) * G_\sigma(x, y)] \right| \quad (1)$$

where  $s = (x, y)$  is a pixel,  $I(x, y)$  is the image value at  $(x, y)$  pixel, and  $\vec{n}$  represents the unit vector in the  $\theta$  direction.

The traditional edge detection approach uses a threshold. If the edge energy falls above this value for a given pixel, this pixel is considered as a point of an edge. EdgeFlow also uses thresholding but only after an energy propagation trick with which edge energies are shifted and accumulated during iterative steps. The direction of this propagation is based on probabilities.  $P(s, \theta)$  gives the probability of finding an edge in the direction  $\theta$  within the distance  $d = 4\sigma$ :

$$P(s, \theta) = \frac{Error(s, \theta)}{Error(s, \theta) + Error(s, \theta + \pi)}, \text{ where} \quad (2)$$

$$Error(s, \theta) = |I_\sigma(x, y) * DOOG_{\sigma, \theta}(x, y)| \quad (3)$$

is a kind of predictive coding error in direction  $\theta$  at scale  $\sigma$ , and

$$DOOG_{\sigma, \theta}(x, y) = G_\sigma(x', y') - G_\sigma(x' + d, y') \quad (4)$$

$$x' = x * \cos(\theta) + y * \sin(\theta), \quad y' = -x * \sin(\theta) + y * \cos(\theta) \quad (5)$$

The edge flow vector  $\vec{F}(s)$  is derived from  $E(s)$  and  $\Theta(s)$ . Its magnitude is proportional to the edge energy, and its phase is directed towards the nearest edge on the given scale  $\sigma$ . (Summing up probabilities instead of finding the maximum reduces the effects of noises.)

$$\vec{F}(s) = \sum_{\theta(s) \leq \theta < \theta(s) + \pi} E(s, \theta) * \exp(j\theta) \quad (6)$$

$$\Theta(s) = \arg \max_{\theta} \left\{ \sum_{\theta(s) \leq \theta' < \theta(s) + \pi} P(s, \theta') \right\} \quad (7)$$

Edge flow vectors are propagated with an iterative algorithm proposed by Ma and Manjunath [8].

1.  $n = 0, \vec{F}_0(s) = \vec{F}(s)$
2.  $\vec{F}_{n+1}(s) = 0$
3. for every  $s'$  neighbour of  $s$  if  $\vec{F}_n(s) \cdot \vec{F}_n(s') > 0$  where dot denotes inner product then  $\vec{F}_{n+1}(s') = \vec{F}_n(s') + \vec{F}_n(s)$

The iteration stops when no change occurs. Edge detection ends with the usual thresholding for the magnitudes of the edge flow vectors. According to [6], [7] and our experiments a fixed value can usually be found for this purpose, there is no need for any kind of analysis of the resulting image.

### 3 Reference System

For single view analysis three landmarks are named in publications [6]: the pectoral muscle, the nipple and the boundary of the breast. These landmarks segment the breast to its anatomical regions.

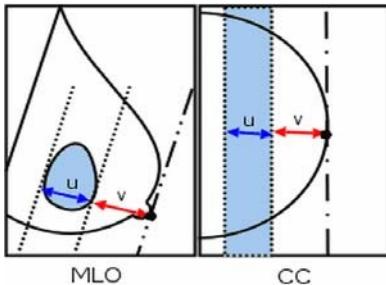
Many complex algorithms tried to establish 3-D reconstruction [8] of breast segments. With 3-D reconstruction the shape of the mass, microcalcification distributions and matching objects can be determined, which help to distinguish between malignant/benign cases and to reduce false positive cases. Because of the difficulties of 3-D reconstruction our main aim was only to build a simple “2.5-D” positioning system, which can find the approximate corresponding region to a region on the other view, thus it is able to help joint analysis of the CC and MLO views. The system works similar in concept to the procedure a radiologist applies at comparing the two pictures.

CC and MLO are two-dimensional projections of the three dimensional object. Therefore a stripe will correspond to a region on the other image. The reference system is to calculate the position of this stripe. The algorithm is founded on three simple hypotheses:

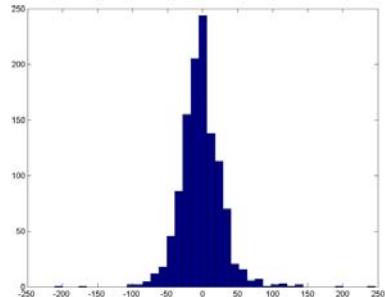
1. The position of the nipple can be estimated by laying a tangent on the breast border parallel with the pectoral muscle.
2. The pectoral muscle on a CC image is assumed to be the vertical axis.
3. The distance covered from the nipple perpendicular to the pectoral muscle on MLO approximately corresponds to the distance measured up on the horizontal axis from the nipple on CC.

The first step of the algorithm is to find the angle enclosed by the pectoral muscle and the horizontal axis on MLO views. With the angle a tangent is laid on the breast border marking the nipple. The distances of the observed region from the tangent ( $u$  and  $v$ ) – are measured. The same distances are measured up on the perpendicular line to the tangent from the nipple of the other view. The two points and the angle of the tangent mark out the stripe. (See Fig. 1.)

The correctness of the reference system was tested by a statistical analysis. Cases with  $400\mu/\text{pixel}$  resolution (600\*400 pixels) from the DDSM database [9] were selected indiscriminately, where these contained only one pathological growth on each views according to the radiologists' assessments. Therefore it could be assumed, that those two masses or calcification clusters correspond to each other on the two views. The pixel corresponding to the centroid of the growth on the MLO was determined, and the deviation of the result from the centroid of the growth on the CC was measured in pixel. The results (See Fig. 2.) show that the assumption of the hypotheses was correct though there is some variance caused by the failures of the algorithm, wrong radiologist assessment or the flaw of the hypotheses (because of breast deformation) for a few cases. To compensate the effect of variance the width of the stripe can be increased by a constant or by a number relative to the width of the stripe to counteract the deviation of the algorithm.



**Fig. 1.** The corresponding stripe on the CC of a selected region on the MLO



**Fig. 2.** Histogram of pixel errors, number of cases 1159

Pectoral muscle is a roughly triangular region with high intensity and is located at the upper corner of the MLO mammogram. It has a higher intensity than the surrounding tissues therefore its border appears as a sharp intensity change, as an edge. Therefore boundary detection - Edgeflow - is the first step of finding the pectoral muscle, then the elimination of weak edges with cutting at a threshold.

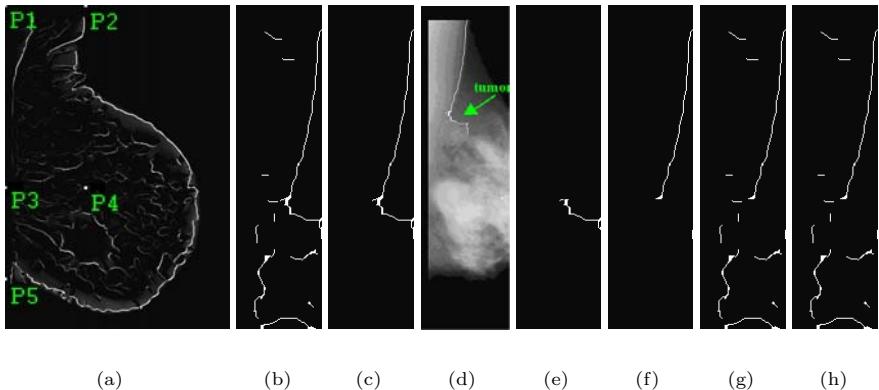
Secondly, a region of interest (ROI) containing the pectoral muscle is obtained according to the commendation of paper [6]. Five control points are used. P1: top-left corner pixel, P2: top-right pixel of the breast boundary, P5: lowest pixel on the left of breast boundary, P3: 2/3 between P1 and P5, P4: completes a rectangle with P1, P2 and P3 and forms the ROI. (See Fig. 3(a).)

As the whole line of the pectoral muscle is not needed, just an approximation of the angle enclosed by the pectoral muscle and the horizontal axis, the iteration processing the lines diverges from paper [6]. Like the deletion of line segments disturbing the angle finding is allowed. The pseudo code of the algorithm is:

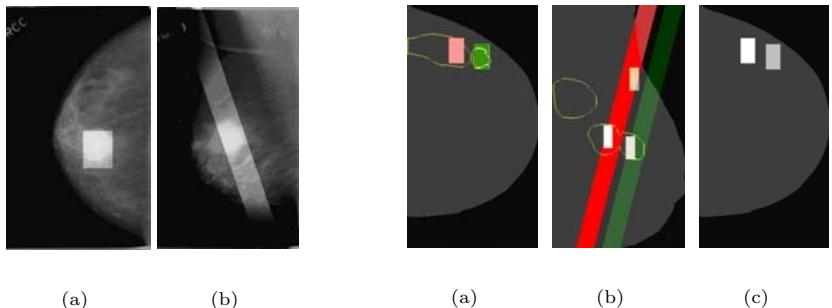
1.  $n = 0, BW_0 = ROI$
2.  $L_n = \text{longest object on } BW_0$
3.  $L_n$  is divided to parts with uniform length along the vertical axis
4.  $L_{bad_n} = \text{objects which enclose } < 40^\circ \text{ or } > 90^\circ \text{ angles with the horizontal axis}$
5.  $L_{good_n} = L_n - L_{bad_n}, BW_{n+1} = BW_n - L_{bad_n}$
6. – if  $BW_{n+1} == BW_n$ 
  - then iteration stops, the pectoral muscle is the object  $L_{good_n}$
  - else  $n = n + 1$  and go to Step 2.

The steps 3, 4 and 5 are used to increase the robustness of the algorithm for cases, where a mass or blood vessel deflects the edge of the pectoral muscle. (See Fig. 3.) With the deletion of segments that cannot be part of the pectoral muscle the angle is better approached.

The nipple is marked out by a tangent parallel to the pectoral muscle laid on the breast border. With the knowledge of the nipple position and the angle of pectoral muscle



**Fig. 3.** (a) ROI selection (b)  $BW_0$ , (c)  $L_0$ , (d)  $L_0 + Picture$ , (e)  $L\_bad_0$ , (f)  $L\_good_1$ , (g)  $BW_1$ , (h)  $L_1 + Picture$



**Fig. 4.** (a) Original image, (b) Strip on other view

**Fig. 5.** Figure showing the process of matching

connection between the two views is provided by simple coordinate transformations. (See. Fig 4.)

#### 4 Application of 2.5-D Reconstruction on Microcalcification Clusters

The probability accompanied to a calcification cluster is modified with the area ratio of the stripe corresponding to it and of other calcification clusters found on the other view. Fig. 5. represents such a matching. 5(a) shows the result of the calcification detection algorithm on a CC view. Two ROIs were found where the brightness (intensity) values of these regions are proportional to the “probability” of finding a calcification cluster. The Fig. 5(b). image shows the corresponding stripes on the MLO view, while Fig. 5(c). shows the result: the “probabilities” are changed according to the matching ratios.

## 5 Texture Segmentation

For masses a more sophisticated joint analysis can be done because they have a distinctive texture that makes it possible to do a texture-based pairing in the stripe. Since the given mass detecting algorithms are fairly characteristic in size and shape of the identified mass, a good segmenting algorithm is needed.

The first question arising when trying to apply EdgeFlow is the selection of the proper scale. After running it for a wide variety of mammographic images and range of scales, scales 1, 2 and 3 seem to be useful. Since the EdgeFlow algorithm itself only detects edges, some further steps are necessary to create a segmentation from its output: line segments should be linked creating continuous borders and closed segments. With some basic morphological operations (removing isolated pixels, dilation, removing disjoined line segments) one can get a practically good segmentation.

However, the result is sometimes too detailed, or may also contain unduly small segments. Computing some texture features and using clustering for the segments based on them can solve these problems. Note that in this case the number of clusters is not equal to the numbers of segments created after merging the members of each cluster, since these members may form more isolated groups on the image. With about 100 isolated areas the resulting segmentation is adequate for our aims. By binary search for the number of clusters needed this number can be approximated in 2-3 steps. (The number of segments on the original segmentation varies from about 80 up to even 300. Small regions are forced to merge even if we have less than 100 segments.)

The texture features used are as follows: mean of intensity, variance of intensity, mean and variance of co-occurrence values, mean and variance of grey-level differences. (Co-occurrence matrix and grey-level differences are image features used with great success for maa detection in the project. Reviewing these features of this article, but one can find descriptions in [9].) For clustering four methods have been tested: single linkage hierarchical, k-means, fuzzy k-means-and subtractive clustering [10], [11], [12]. Reasoned by our experiments the first two ones have been chosen for their simplicity and reliability.

## 6 Matching of Masses

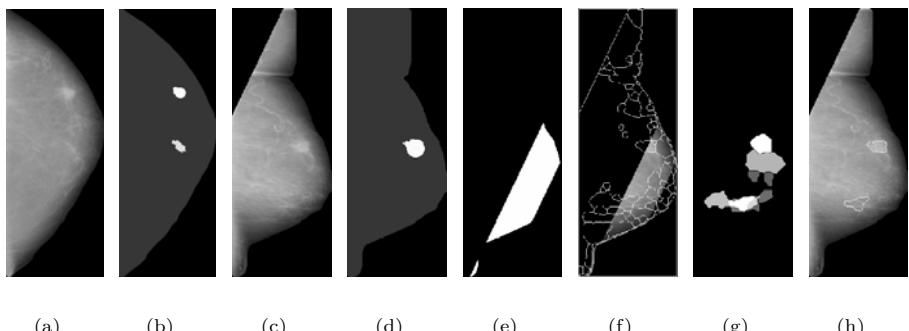
Once a good segmenting algorithm and characteristic texture features are given, the accuracy of mass detecting can be increased matching their results on the different views. If pairs can be found on both pictures of the same side, the identifying probability of that mass should be increased. And finding no pair reduces this probability.

The expression „...“ is used since finding a counterpart to a mass merely says that there is something characteristic „...“ – because it can be seen from both views – but it might be either malignant or benign. Alike – if no counterpart is found, it says the mass supposed to be recognized on one of the images is only virtual, its appearance is the result of some overlaying

tissues. Note also that this correspondence can solely be done for “clear” breasts. For dense ones even experienced radiologists can rarely find the a mass on both images.

Pairing goes by the following steps:

1. In the beginning results of a mass detection algorithm are given – usually a binary mask covering the mass-candidate area with the probability of that hit (see Fig. 6(b), 6(d)). (During the matching mass-candidates of one image called ... are to be paired with mass-candidates of the other one called ...)
2. A mass-candidate – in this case the upper one – is chosen from the source image.
3. Based on the reference system described before, a stripe in the target image is determined. In Fig. 6(f). the segmentation can be seen, with the stripe highlighted.
4. Segments overlapped by a mass-candidate are identified, and for each segment the distance – a measure of similarity – is computed from segments overlapped by the corresponding stripe, creating a non-binary mask where nonzero elements cover the paired segments. For each segment the covering values are the reciprocal of the above detailed distance. Thus on Fig. 6(g). intensity is proportional to similarity. (Areas fairly overlapping with the mask or having radically different size or intensity are omitted.) Since one mass-candidate area may overlap more segments, these masks must be added. In fact as EdgeFlow may be run for more scales and for more clustering algorithms, one may also sum along these dimensions as well. (In our experiments 3 scales and k-means segmentation are used.) At the end this non-binary mask is thresholded. The resulting pairs can be seen on Fig. 6(h).
5. Taking the hits on the source image one by one, we examine if its pairs overlap with any of the mass candidates on the target image. If so, the similarity of this pair is the mean of those nonzero elements on the non-



**Fig. 6.** Steps of pairing for masses based on area and texture matching

- binary mask (Fig. 6(h).) that are covered by the given mask pair on the target image.
6. This pairing is done in reverse direction as well. Mass-candidates that are not paired are dropped.

## 7 Performance

The calcification matching was analysed over 188 cases (376 pairs of mammographic images). 66 of these cases contained malinous calcifications. The original calcification searching algorithm solved 63 of these cases but only 1 (in 122) of the negative ones. As it was previously emphasised the number of false positive hits are enormous. With the combined matching 61 malignant cases and 15 normal cases are solved.

- 12.4% of false positives hits were dropped
- 96.8% of true positive hits were kept
- 3.2% of true positive hits were dropped

The reasons for the loss of positive markers are summarised in the conclusion.

In the case of mass detection the above detailed algorithm has been tested on 363 pairs of mammographic images – with 256 masses – from the DDSM database.

- 90% of true positive hits were kept
- 18% of false positives were dropped

There are 4 main reasons for dropping true positive hits: (i) a mass can be seen on both views but only one of them is marked as mass-candidate (3/18), (ii) the mass can be seen only on one of the images (2/18), (iii) the reference system is not accurate enough (7/18), (iv) miscellaneous errors of the pairing algorithm (2/18). The results of mass-pairing are worse than that of calcification-pairing since the number of mass-candidates per image is higher than the number of calcification-candidates per image.

## 8 Conclusions

The paper proposed a relatively simple way of combining the results of mass and microcalcification detection algorithms applied for individual X-ray breast images. The joint analysis follows the way applied by skilled radiologists: if a suspicious area can be found in one view, usually its corresponding pair should be detected in the other view of the same breast. The first results – based on a few hundred of cases - show that using this approach the number of false positive detections can be reduced significantly while the decrease of true positive hits is relatively small. The loss of a few true positive cases comes from three problems: (i) the variance of the corresponding distances (Fig. 2), (ii) the lack of detected

microcalcification cluster or mass in one of the corresponding views, (iii) the lack of microcalcification cluster or mass in one of the corresponding views. The variance can be decreased if the different deformation caused by breast compression is taken into consideration. The reason of the second problem is that although there are signs of tumour in both views, the primal algorithms can detect them only in one of the views, in these cases the primal algorithms should be improved. The third problem cannot be solved as in these cases the signs cannot be seen in one of the images even by a skilled radiologist. This means that in such cases other modality like ultrasound should also be used. The proposed joint analysis system is under testing: the whole 2.600 cases of the DDSM data base will be analysed in the near future.

## References

1. L. Tabár: Diagnosis and In-Depth Differential Diagnosis of Breast Diseases. Breast Imaging and Interventional Procedures, ESDIR, Turku, Finland, 1996.
2. G. Horváth, J. Valyon, Gy. Strausz, B. Pataki, L. Sragner, L. Lasztovicza, N. Székely: Intelligent Advisory System for Screening Mammography. Proc. of the IMTC 2004 - Instrumentation and Measurement Technology Conference, Como, Italy, 2004, Vol. pp.
3. N. Székely, N. Tóth, B. Pataki: A Hybrid System for Detecting Masses in Mammographic Images. Proc. of the IMTC 2004 - Instrumentation and Measurement Technology Conference, Como, Italy, 2004. Vol. pp.
4. Songyang Yu, Ling Guan: A CAD System for the Automatic Detection of Clustered Microcalcifications in Digitized Mammogram Films. IEEE Trans. on Medical Imaging, Vol. 19, No. 2, February 2000.
5. B. Verma and J. Zakos: A Computer-Aided Diagnosis System for Digital Mammograms Based on Fuzzy-Neural Feature Extraction Techniques. IEEE Trans. on Information Technology in Biomedicine, vol. 5. No. 1. pp. 46-54. 2001.
6. R.J. Ferrari, R. M. Rangayyan, J. E. L. Desautels, R. A. Borges, A. F. Frre: Automatic Identification of the Pectoral Muscle in Mammograms. IEEE Trans. on Image Processing, Vol. 23, No 2, pp. 232-245, February 2004.
7. Wei-Ying Ma, B. S. Manjunath: EdgeFlow: A Technique for Boundary Detection and Image Segmentation. IEEE Trans. on Image Processing, Vol. 9, No 8, pp. 1375-1388, August 2000.
8. M. Yam, M. Brady, R. Highnam, Ch. Behrenbruch, R. English and Y. Kita: Three-Dimensional Reconstruction of Microcalcification Clusters from Two Mammographic Views. IEEE Trans. on Image Processing, Vol. 20, No. 6, June 2001.
9. M. Heath, K. Bowyer, D. Kopans, R. Moore, K. Chang, S. Munishkumaran and P. Kegelmeyer: Current Status of the Digital Database for Screening Mammography. In: Digital Mammography, N. Karssemeier, M. Thijssen, J. Hendriks and L. van Erning (eds.) Proc. of the 4th International Workshop on Digital Mammography, Nijmegen, The Netherlands, 1998. Kluwer Academic, pp. 457-460.
10. I. Pitas: Digital Image Processing and Algorithms and Applications. John Wiley & Sons, New York, 2000.
11. D. E. Gustafson and W. C. Kessel: Fuzzy Clustering with a Fuzzy Covariance Matrix. Proc. IEEE-CDC, Vol. 2, pp. 761-766, 1979.
12. R.O. Duda, P.E. Hart: Pattern Classification and Scene Analysis. John Wiley & Sons, 1973.

# Approximated Classification in Interactive Facial Image Retrieval

Zhirong Yang and Jorma Laaksonen

Laboratory of Computer and Information Science,  
Helsinki University of Technology,  
P.O. Box 5400, FI-02015 HUT, Espoo, Finland  
`{zhirong.yang, jorma.laaksonen}@hut.fi`

**Abstract.** For databases of facial images, where each subject is depicted in only one or a few images, the query precision of interactive retrieval suffers from the problem of extremely small class sizes. A potential way to address this problem is to employ automatic even though imperfect classification on the images according to some high level concepts. In this paper we point out that significant improvement in terms of the occurrence of the first subject hit is feasible only when the classifiers are of sufficient accuracy. In this work Support Vector Machines (SVMs) are incorporated in order to obtain high accuracy for classifying the imbalanced data. We also propose an automatic method to choose the penalty factor of training error and the width parameter of the radial basis function used in training the SVM classifiers. More significant improvement in the speed of retrieval is feasible with small classes than with larger ones. The results of our experiments suggest that the first subject hit can be obtained two to five times faster for semantic classes such as “black persons” or “eyeglass-wearing persons”.

## 1 Introduction

Most existing face recognition systems require the user to provide a starting image. This however is not practical in some situations, e.g., when searching a previously seen image via the user’s recalling. To address this problem, some interactive facial image retrieval systems such as [9] have been proposed, which are mainly based on learning the relevance feedback from the user.

Unlike content-based image retrieval (CBIR) systems on general images, the query precision on facial images suffers from the problem of extremely small class sizes [9]. In a popular collection, FERET [7], most subjects possess only one or two frontal images. Making the  $\dots \dots \dots \dots$  appear as early as possible is critical for the success of interactive facial image retrieval. If only images that depict the correct person are regarded as relevant, many zero pages (i.e. the images in these rounds are all non-relevant) will be displayed until the first relevant image emerges. This is because the negative responses from the user in early rounds provide only little semantic information and – as a result – the iteration progresses in a nearly random manner.

The above problem can be relieved by allowing the user to submit partial knowledge, e.g. gender or race, on the query target. With this kind of ... or ... there are far less image candidates than the entire collection and the first subject hit will undoubtedly appear much sooner. However, this requires labeling of the images according to the semantic criteria and manual work is not feasible for a large database. Thus approximating the semantic annotation by automatic classification is desired.

Classifiers constructed by machine learning are generally not perfect. If the correct target happened to be misclassified then it would never be displayed due to the filtering. In this paper we suppose the user would not give up an unsuccessful query after some number of endurable rounds – instead he or she would remove the restriction and continue the search by using the entire database. This assumption allows us to compute the mean position of the first subject hit and assess the advantage obtainable with approximated classification.

In this paper we point out that only classifiers with very high accuracy can be significantly beneficial to the retrieval. This basic assumption is verified by experiments in Section 2. Support Vector Machines are used for automatic classification. We review SVM's principles and discuss how to choose its parameters with radial basis function kernels in Section 3. Experiments are presented in Section 4, and finally are the conclusions and future work in Section 5.

## 2 Approximated Classification

Restricting the image candidates by some ... is a natural idea to improve query performance. However in CBIR the true semantic classes are usually not available and we have to approximate them by ... which can be defined by some automatic classifiers. If a classifier constructed by machine learning has only a small misclassification error rate, the first relevant image can be shown earlier on the average. In this section we will present a set of preliminary experiments to sustain the idea.

### 2.1 First Subject Hit Advantage Performance Measure

The position of the first relevant hit is critical to the success of CBIR. For example, if there is no relevant image displayed in the first twenty rounds, the user would probably deem the system useless and give up the query. In contrast, if the first relevant hit appears within the user's tolerance, say the first five or ten rounds, the query will probably proceed and further relevant images found.

Suppose  $N$  and  $R$  are the number of all images and relevant images in the database, respectively. Denote by  $j$  the random variable for position of the first subject hit using random retrieval. It is not difficult to prove that the mean of  $j$  is  $E\{j\} = (N - R)/(R + 1)$ . Thus the improvement compared with the random retrieval can be quantified by the following ... (FSHA) measurement:

$$\text{FSHA}(i; N, R) = \frac{E\{j\}}{i} = \frac{N - R}{i \cdot (R + 1)}, \quad (1)$$

where  $i \in \{0, 1, \dots, N - R\}$  is the position of the first subject hit using the improved retrieval. FSHA equals one when the retrieval is done in a random manner and increases when the retrieval is able to return the first relevant image earlier. For example, it equals two when the first subject hit occurs in the position whose index is half of the expected index in random retrieval.

## 2.2 Simulated Classification Study

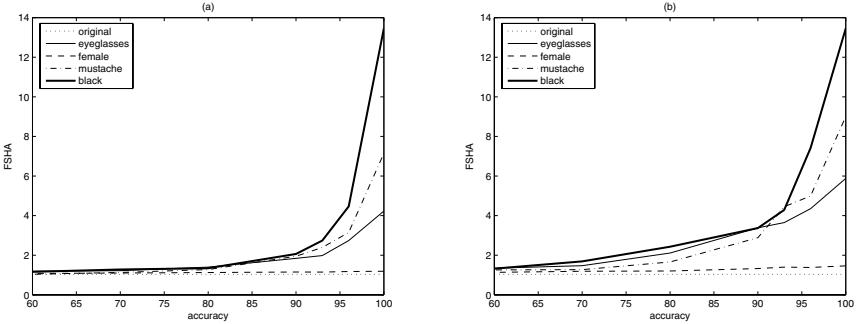
In order to test the advantage attainable by queries with approximated classification, our experiments were carried out as follows. For a set of semantic classes  $\{C_i\}$ , we simulated approximated classification by random sampling with varying percentages of correct and incorrect decisions according to the criterion  $C_i$ . For each simulated class and classification accuracy we then ran the PicSOM CBIR system [5] to calculate the attained average FSHA. We looped over all subjects  $\{S_t\}$  in the class  $C_i$  and at each loop, the retrieval goal was to search all images depicting the current subject  $S_t$ . 20 images were “displayed” per round and the first set of images was randomly selected from  $C_i$ . In the automated evaluation the sole criterion for relevance of an image was whether it depicted the current subject  $S_t$  or not. If no subject hit appeared within a predefined number of rounds,  $T$ , the target was deemed to have been misclassified. The test program then removed the restriction and resorted to using the entire database until the first subject hit occurred.

In the experiments we used the FERET database of readily segmented facial images collected under the FERET program [7]. 2409 frontal facial images (pose mark “fa” or “fb”) of 867 subjects were stored in the database for the experiments. Table 1 shows the specification of four tested true semantic classes.

We used two different  $T$  values, 10 and 20, to study the query performance. The results are shown in Figure 1. The FSAs using the entire database are shown as the dotted baseline at the bottom of the plots. Due to the extremely small subject classes the retrieval without restriction is nearly random before the first subject hit, and its FSA is very close to unity. When the restriction is applied in the early  $T$  rounds of the query, the FSAs increase to different degrees, depending on the class type, the accuracy of the classifier and the cutting round  $T$ . The small classes,  $\text{eyeglasses}$ ,  $\text{female}$ ,  $\text{mustache}$ , and  $\text{black}$  have more significant improvement than the large one,  $\text{whole database}$ . In addition, the improvement for all classes is very slight when the classification accuracies are lower than 80%. That is, we get the benefits from the approximated classification only with very

**Table 1.** Tested true semantic classes

class name	images	subjects a priori	
eyeglasses	262	126	15%
female	914	366	42%
mustache	256	81	9%
black	199	72	8%
whole database	2409	867	



**Fig. 1.** FSHA with approximated classification of different accuracies. The predefined number of rounds before removing the restriction is  $T = 10$  in (a) and  $T = 20$  in (b)

accurate classifiers. This phenomenon becomes more evident when  $T = 10$ , i.e. the user's tolerance is smaller and the approximated classification is given up earlier and the whole database used instead.

### 3 Support Vector Machines

To obtain accurate classifiers especially for highly imbalanced data is not a trivial task. We adopt Support Vector Machines (SVMs) [8], which have shown good generalization performance in a number of diverse applications. In this section we give a brief introduction of SVM and describe how we have chosen its parameters.

#### 3.1 Principles of SVM

Given a training set of instance-label pairs  $(\mathbf{x}_i, y_i)$ ,  $i = 1, \dots, l$  where  $\mathbf{x}_i \in R^n$  and  $y_i \in \{1, -1\}$ , the Support Vector Machines require the solution of the following optimization problem:

$$\begin{aligned} & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C_+ \sum_{i: y_i=1} \xi_i + C_- \sum_{i: y_i=-1} \xi_i \\ & \text{subject to } y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \quad \xi_i \geq 0, i = 1, \dots, l. \end{aligned} \quad (2)$$

Here the training vectors  $\mathbf{x}_i$  are implicitly mapped into a higher dimensional space by the function  $\phi$ .  $C_+$  and  $C_-$  are positive penalty parameters of the error terms. The above problem is usually solved by introducing a set of Lagrange multipliers  $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_l\}$ :

$$\begin{aligned} & \max_{\boldsymbol{\alpha}} \mathbf{e}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} \\ & \text{subject to } 0 \leq \alpha_i \leq C_+ \text{ if } y_i = 1, \\ & \quad 0 \leq \alpha_i \leq C_- \text{ if } y_i = -1, \\ & \quad \mathbf{y}^T \boldsymbol{\alpha} = 0, \end{aligned} \quad (3)$$

where  $\mathbf{e}$  is the vector of all ones,  $\mathbf{Q}$  is an  $l \times l$  positive semidefinite matrix given by  $Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ , where  $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$  is called the kernel function. Then  $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i)$  and

$$\text{sgn}(\mathbf{w}^T \phi(\mathbf{x}) + b) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (4)$$

is the decision function. For the kernel function  $K(\cdot, \cdot)$ , we have chosen the radial basis function (RBF) with common variance:

$$K_{RBF}(\mathbf{x}, \mathbf{z}; \gamma) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \quad (5)$$

because it has good classification power and only one parameter needs to be determined. We unify  $C_+$  and  $C_-$  into a single parameter  $C$  with weights according to the inverse of their prior probability estimates, i.e.  $C_+ = C$  and  $C_- = C \cdot N^+/N^-$ , where  $N^+$  and  $N^-$  are the numbers of the positive and negative labels, respectively.

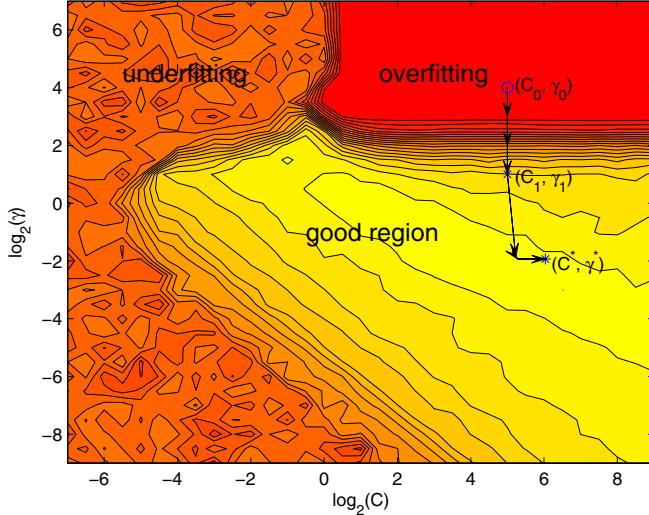
### 3.2 Choosing SVM Parameters

The SVM experiments in this paper were implemented based on the `LIBSVM` library [1]. In the experiments it was noticed that the parameter settings have great impact on the performance of the resulting classifiers. There exist some parameter selection methods (e.g. [2]) which were reported to find good values for these parameters automatically, but it has also been shown that they are unstable in practice [3]. The gradient-based optimization algorithms adopted by these methods require a smoothing surface and a good starting point, which is albeit unknown beforehand. In addition, the penalty parameter  $C$  is incorporated into the kernel matrix, which is valid only when the SVMs are  $L_2$  norm, but such SVMs for imbalanced data are not supported by most current SVM toolkits.

One can make use of some geomorphologic knowledge about the accuracy surface and then apply stochastic optimization to obtain a much more efficient parameter selection method. We first need a combined accuracy estimate which is proper for both a class and its complement. Given a true semantic class  $C_M$  and its complement  $\bar{C}_M$ , which are approximated by the restriction class  $C_R$  and  $\bar{C}_R$ , respectively, we adopt the minimum of the true positive accuracy and the true negative one, i.e.

$$\text{accu} = \min\left(\frac{|C_R \cap C_M|}{|C_M|}, \frac{|\bar{C}_R \cap \bar{C}_M|}{|\bar{C}_M|}\right). \quad (6)$$

Figure 2 illustrates an example of such accuracy measure's contour plot on the  $(C, \gamma)$ -plane. The example comes from 20-fold classification of the eyeglasses class using feature calculated from the left eye of each subject (see Section 4.1). The grid search used to draw this figure is not a part of the searching procedure, but helps us better understand the distribution of good values for  $C$  and  $\gamma$ .



**Fig. 2.** Accuracy contour of eyeglasses in the  $(C, \gamma)$ -space with 20-fold cross-validation

Similar contour shapes have been also observed on a number of other real-world datasets (e.g. [1, 6]).

Based on the above understanding, we propose a path searching algorithm as follows: (1) Choose a large  $C$  and a large  $\gamma$ , i.e. a point on the overfitting plateau, for example,  $C_0 = 2^5$  and  $\gamma_0 = 2^4$ . Then apply a line search downwards by decreasing  $\gamma$  until  $\text{accu} > 0.5$ . (2) Suppose the resulting point of step 1 is  $(C_1, \gamma_1)$ , and for convenience, we write  $\boldsymbol{\theta}_t = (C_t, \gamma_t)$ . Denote  $\mathbf{g}(t)$  the gradient of the accuracy surface and given  $\Delta\boldsymbol{\theta}_1 = d_1\mathbf{g}(1)$ , iteratively apply the conjugate gradient optimization procedure:

$$\Delta\boldsymbol{\theta}_t = \beta(t)\Delta\boldsymbol{\theta}_{t-1} + \mathbf{g}(t), \quad (7)$$

$$\beta(t) = \frac{\|\mathbf{g}^T(t)\|^2}{\|\mathbf{g}^T(t-1)\|^2}. \quad (8)$$

Step 1 locates a good starting point  $\boldsymbol{\theta}_1$  for step 2.  $\boldsymbol{\theta}_1$  is probably on the upper hill side of the good region mountain. The gradient  $\mathbf{g}(t)$  at a point  $\boldsymbol{\theta}_t$  is approximated by a one-sided finite difference where the change of accuracy is measured separately in the  $C$  and  $\gamma$  directions with difference magnitude  $h_k$ . A common form for the sequence  $h_k$  is  $h_k = h/k^m$ , where  $h$  and  $m$  are predefined positive constants.  $d_1$  is the initial learning rate at  $(C_1, \gamma_1)$ . If  $\text{accu}(\boldsymbol{\theta}_t) > \text{accu}(\boldsymbol{\theta}_{t-1})$  then record  $\boldsymbol{\theta}_t$  and  $\mathbf{g}(t)$ ,  $t \leftarrow t + 1$ ,  $k \leftarrow k + 1$ . Otherwise, just shrink  $h_k$  by  $k \leftarrow k + 1$ . This way we can obtain a path with only increasing accuracies. Note that the conjugate gradient method only works in low stochastic level. Therefore 20-fold cross-validation was used instead of the popular 5-fold setting.

The searching path for the eyeglasses class using the left eye feature is shown by arrows in Figure 2. Step 1 began with the initial point  $(C_0, \gamma_0) = (32, 16)$

and the resulting point was  $(C_1, \gamma_1) = (32, 2)$ , from which the conjugate gradient optimization started with the setting  $d_1 = 50$ ,  $h = 1$ , and  $m = 0.1$ . After 18 calls of the cross-validation procedure the searching algorithm returned the final point  $(C^*, \gamma^*) = (65.7, 0.262)$  with accuracy 90.45%. The application of the procedure was the same for all other features and classes.

## 4 Experiments

The testbed we used in the experiments is our CBIR system named PicSOM [5], which utilizes the Self-Organizing Maps (SOMs) as the underlying indexing and relevance feedback processing technique. Some images are shown in each round of a query and the user is supposed to mark zero or more of them as relevant to the current retrieval task. The rest images in that round are treated as non-relevant. This relevance feedback is then used to form relevance score values in the best-matching map units (BMUs) corresponding to the shown images on each participating SOM. The effect of the hits is spread to the neighboring SOM units by low-pass filtering over the SOM surface.

More than one feature can be involved simultaneously and the PicSOM system has a separate trained SOM for each. The convolution provides implicit feature weighting because features that fail to coincide with the user's conceptions mix positive and negative user responses in the same or nearby map units. Such SOMs will consequently produce lower scores than those SOMs that match the user's expectations and impression of image similarity and thus produce areas or clusters of high positive response. The total scores for the candidate images are then obtained by simply summing up the mapwise values in their BMUs. Finally, a number of unseen images with the highest total scores are displayed to the user in the next round.

### 4.1 Data

In the FERET collection [7] the coordinates of the facial parts (eyes, nose and mouth) were obtained from the ground truth data, with which we calibrated the head rotation so that all faces were upright. All face boxes were normalized to the same size of  $46 \times 56$  pixels, with fixed locations for left eye (31,24) and right eye (16,24) in accordance to the MPEG-7 standard [4]. The box sizes of the face and the facial parts are shown in the second column of Table 2.

After extracting the raw features within the boxes mentioned above, we applied Singular Value Decomposition (SVD) to obtain lower-dimensional eigen-features of the face and its parts. The numbers of the principle components preserved are shown in the third column of Table 2.

### 4.2 Single Classifier Results

The resulting 20-fold cross-validation accuracies and respective parameters for all tested classes using individual SVM classifiers are shown in Table 3. The

**Table 2.** Specification of the used features

feature name	normalized size	eigenfeature dimensions
face	$46 \times 56$	150
left eye	$24 \times 16$	30
right eye	$24 \times 16$	30
nose	$21 \times 21$	30
mouth	$36 \times 18$	50

**Table 3.** True positive and true negative accuracies for individual classifiers with 20-fold cross-validation

class	face	left eye	right eye	nose	mouth
eyeglasses	77.10%, 88.22% (181.02, 0.0030)	90.45%, 97.90% (65.67, 0.2617)	90.84%, 97.62% (63.29, 0.2196)	88.55%, 91.01% (0.54, 3.4822)	— —
female	87.09%, 90.84% (1351.2, 0.0073)	82.17%, 82.81% (32.00, 0.1768)	78.77%, 82.34% (32.00, 0.1768)	68.93%, 71.04% (18.38, 0.1015)	81.84%, 81.80% (62.18, 0.1075)
mustache	78.52%, 78.17% (2.00, 0.0313)	— —	— —	70.31%, 71.06% (16.46, 0.1328)	84.38%, 87.46% (0.66, 0.0291)
black	79.90%, 85.48% (90.51, 0.0032)	79.90%, 82.24% (0.66, 2974)	77.89%, 79.91% (0.23, 0.6156)	71.86%, 75.88% (0.81, 0.0670)	80.40%, 84.71% (0.25, 0.1768)

first percentage in each cell is the accuracy for the true positive and the second for the true negative. The number pair under the accuracy percentages is the respective  $C$  and  $\gamma$ . It can be seen that the best accuracy for the eyeglasses class was obtained with the eye features, for the gender with the face feature, and for the mustache with the mouth feature. These results are consistent with our everyday experience. The case of the black race is not so obvious and all other features but the nose seem to perform equally well, but worse than for the three other classes.

### 4.3 Combining Individual Classifiers

Although the features used in the experiments are not fully uncorrelated, it is still beneficial to combine some of the individual classifiers to a stronger one. This can be done by performing majority voting weighted by their accuracies. For a specific class category, denote  $L(f, I)$  the label to which an image  $I$  is classified by using the feature  $f$  and  $\text{accu}(f)$  the respective accuracy of that classifier. Assign  $j$  to  $I$  if

$$j = \operatorname{argmax}_i \left\{ \sum_{L(f, I)=i} [\text{accu}(f) - 0.5] \right\}. \quad (9)$$

The subtractive term 0.5 is used here to give the best-performing classifiers extra reward compared to the worst-performing ones. Table 4 shows the accuracies after combination and the respective features used. It can be seen that for the classes of female and black the accuracies can be significantly improved

**Table 4.** Leave-one-subject-out true positive and true negative accuracies for combined classifiers

class	accuracy	features used
eyeglasses	95.91%, 96.88%	face, left eye, right eye, nose
female	90.62%, 94.58%	face, left eye, right eye, nose, mouth
mustache	84.11%, 87.78%	face, nose, mouth
black	85.70%, 91.04%	face, left eye, right eye, nose, mouth

by combining individual SVM classifiers. The combination also enhanced true positive accuracy for the eyeglasses class. By contrast, the accuracies of the mustache class after the combination remained at the same level as with the mouth feature only.

#### 4.4 Obtainable FSHA Values

We obtained estimates of the FSHA with the combined classifiers by averaging the accuracies of the true positive and the true negative. This mean accuracy was then used when interpolating the FSHA values from the results of Section 2.2. The results shown in Table 5 indicate that the retrieval performance in terms of the first subject hit can be improved to different extent depending on the semantic criterion upon which the approximated classification is based. Also the number of rounds where the restriction is applied is a significant factor. The FSHA values for the eyeglasses class show clear improvement whereas the improvement for the female class alone is quite modest.

**Table 5.** FSHA estimates with the combined classifiers

	eyeglasses	female	mustache	black
T=10	4.2	1.2	1.6	2.0
T=20	5.9	1.3	2.3	3.4

## 5 Conclusions and Future Work

The possibility of incorporating auto-classification into interactive facial image retrieval was probed in this paper. We found that highly accurate classifiers are required to achieve significant advantage in terms of the first subject hit. Support Vector Machines were introduced into this task and we also proposed an automatic method to select good parameter values for training the SVM classifiers. The desired high accuracy can be achieved for a number of class categories by combining individual classifiers created with different low-level facial features. According to our results, we can speed up the occurrence of the first relevant hit by a factor up to nearly six in the case that the person we are searching for is wearing eyeglasses. With the other semantic classes tested, like

the black race or mustache, a bit lower level of improvement can be obtained. Note that even though the improvement by filtering the gender alone is not significant, it is in some cases possible to combine this highly accurate classifier with others to generate more specific semantic subclasses.

Due to limited data for the highly imbalanced classes, we had to use all frontal facial images in the FERET database for training the classifiers and their validation. The experiments to obtain the true values of FSHAs can be easily implemented after more independent external data is acquired, as we now have the operative classifiers available.

There is still plenty of space for further improvement. One of the major questions in the future will be how to handle the class categories with soft boundaries such as hairstyles. Furthermore, so far the class categories which satisfy the accuracy requirement are still limited because we only used five quite general low-level visual features. With the advance of feature extraction techniques we will obtain better classifiers and as a result, more semantic categories can be supported.

## References

1. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
2. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.
3. C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *A Practical Guide to Support Vector Classification*. Document available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
4. ISO/IEC. Information technology - Multimedia content description interface - Part 3: Visual. 15938-3:2002(E).
5. J. Laaksonen, M. Koskela, and E. Oja. PicSOM—self-organizing image retrieval with MPEG-7 content descriptors. *IEEE Transactions on Neural Network*, 13(4):841–853, 2002.
6. J.-H. Lee. Model selection of the bounded SVM formulation using the RBF kernel. Master's thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2001.
7. P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306, 1998.
8. V. Vapnik. *Statistical Learning Theory*. NY: Wiley, New York, 1998.
9. Z. Yang and J. Laaksonen. Interactive retrieval in facial image database using Self-Organizing Maps. In *Proc. of IAPR Conference on Machine Vision Applications (MVA2005)*, Tsukuba Science City, Japan, May 2005.

# Eye-Movements as a Biometric

Roman Bednarik, Tomi Kinnunen, Andrei Mihaila, and Pasi Fränti

Department of Computer Science,

University of Joensuu,

P.O.Box 111, FI-80101 Joensuu, Finland

{bednarik,tkinnu,amihaila,franti}@cs.joensuu.fi

**Abstract.** We propose the use of eye-movements as a biometric. A case study investigating potentials of eye-movement data for biometric purposes was conducted. Twelve participants' eye-movements were measured during still and moving objects viewing. The measured data includes pupil sizes and their dynamics, gaze velocities and distances of infrared reflections of the eyes. For still object viewing of 1 second duration, identification rate of 60 % can be obtained by using dynamics of pupil diameters. We suggest an integration of the eye-movement-based identification into general video-based biometric systems.

## 1 Introduction

*Biometric person authentication* [1] refers to identifying or verifying persons' identity based on their physical and/or behavioral (learned) characteristics. Examples of physical biometrics are *fingerprints* and *facial images*, and examples of behavioral biometrics include *voice* and *signature*. There are numerous applications of biometrics, including forensics, access control to physical facilities, border control, identity verification in e-commerce, and personalizing user profiles in a mobile device.

A perfect biometric should be unique, universal, and permanent over time, easy to measure, cheap in costs, and have high user acceptance. No single biometric fulfills all these requirements simultaneously. For instance, fingerprints and retina are known to be highly unique, but they require dedicated sensors and are not user friendly. On the other hand, voice and facial geometry are not as unique, but they require only a cheap microphone or a camera as a sensor, and they are unobtrusive. Numerous studies have demonstrated that the combination of several complementary biometrics can provide higher recognition accuracy than any individual biometric alone [2,8].

Biometric authentication tasks can be subdivided into *identification* and *verification* tasks. In the identification task, an unknown biometric sample is compared to whole database of known individuals, and the best matching template is selected ( $1:N$  matching). On the other hand, the verification task, or *1:1 matching*, consists of verifying whether the provider of the biometric sample (*claimant*) is the one who (s)he claims to be. In both cases, a “no decision” option is also possible. Verification systems are well-suited for applications having high security requirements, but poorly suited for cases when user friendliness has higher priority. In verification, the user is required to give the identity claim as well as a biometric sample. In identification, the authentication process can be ubiquitous and the user does not even know that he is being authenticated.

Person's eyes provide several useful biometric features both for the high security and user convenient applications. First, iris and retinal patterns are known to be among the most accurate biometrics. Second, eyes are often used as anchor points in geometrical approaches for face detection [10], and the geometric differences (e.g. distance between eyes) can be utilized directly as features in face recognition [3].

In this paper, we propose to use *eye movements* as an additional biometric that can be integrated with other biometrics. To our knowledge, eye-tracking systems have not been considered as a possible solution for a biometric system. The main goal of the present paper is therefore to investigate the potential of eye-tracking as a biometric, as we had no preconceived hypothesis about whether and how the features of eye-movement signal are discriminative or not. This paper reports on a case study conducted by using Tobii ET-1750 eye-tracker in laboratory conditions.

The rest of this paper is organized as follows. In Section 2, we briefly review the current technology for the eye-movement tracking, and discuss its usefulness for biometric application. In Section 3, we consider different alternatives for potential features extracted by the eye-tracker, and describe the feature extraction and classification methods chosen. Experiments are carried out in Section 4, and results reported in Section 5. Discussion is in Section 6, and conclusions are drawn in Section 7.

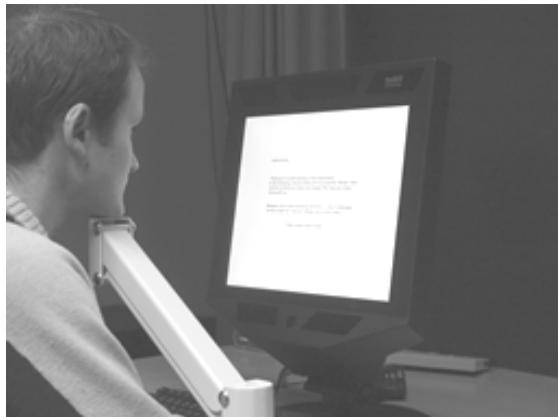
## 2 Eye-Movement Tracking

Humans move their eyes in order to bring an image of inspected object onto fovea, a small and high-resolution area of the retina. Once the image of the object is stabilized on the retina, the information can be extracted. *Eye-tracker* is a device that records these movements. Most of the current eye-trackers use infrared light emitters and video image analysis of the pupil center and reflections from cornea to estimate the direction of gaze, see Figure 1. The infrared reflections are seen as the points with high intensity inside subject's pupil.



**Fig. 1.** Example image from eye-tracker's camera. The bright spots in the middle of the eyes are infrared illumination used by the eye-tracker device

The accuracy of current commercially available eye trackers ranges around 1 degree, while the data is sampled at rates of 50–500Hz. Modern eye-trackers are relatively cheap and able to reliably and unobtrusively collect the gaze data. The usability of eye-tracking is high in controlled laboratory conditions. Their application in real-life situations, however, is still limited due to the need for calibration of each of the users before the recording. An example of such system is shown in Figure 2.



**Fig. 2.** Eye-tracker used in the experiments

Previous research has established a solid evidence about the relation between eye movements, visual attention and underlying cognitive processes [6,7]. Knowing which objects have been visually inspected and in which order and context, one can attempt to infer what cognitive processes were involved to perform a task related to these objects. However, to our knowledge, no attempts were made in order to distinguish individuals based on the properties of eye-movements seen as time-signals.

Eye is never perfectly still. Similarly, pupil diameter is never constant, but oscillates around certain value. This fact is used by the iris biometric systems to enforce the measurement of true authentic eyes. It could be hypothesized, that the mass of the eye-ball, the muscles responsible for movements of the eyes, and the muscles controlling the dilatation of the pupil are anatomically individual. In other words, such a complex system can exhibit high degree of uniqueness. Although the eye-tracker does not provide the direct measures of the muscular systems, the overt movements of eyes and especially the involuntary movements could be thought to reflect the underlying anatomical organization.

Current eye-trackers satisfy most of the desired features for an unobtrusive biometric. In the present eye-tracking systems, the detection of the position of eyes is based on video-imaging, the same technology used in face recognition. Therefore, there is a possibility to join such biometric systems into one, eye-movement based and face-feature based. With a high-resolution sensor the systems could be further combined with iris recognition systems.

The current eye-movement trackers are still too costly, but as the price of the technology becomes lower, wider inclusion into current computer and video-systems is expected. It will, therefore, be possible to simultaneously use all eye-movement measures in combination with other biometrics.

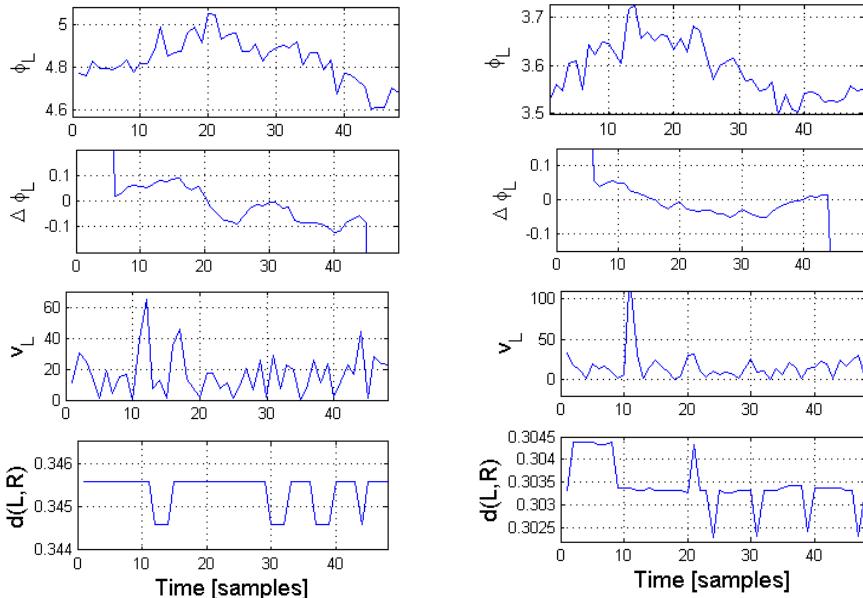
### 3 Person Identification Using Eye-Movements

The eye-tracker employed in the present study provides several measurements for both eyes. First, the eye-tracker outputs normalized coordinates within the camera's

view, as well as estimated gaze-direction measurements. The latter ones are computed through an interpolation between calibration points, whereas the former ones are the coordinates of the infrared light reflections on the cornea of an eye. The device provides also pupil diameters and the distance of the subject from the monitor.

We consider the following features as the candidates for a biometric cue:

- *Pupil diameters ( $\phi_L, \phi_R$ )*
- *Distance between the reflections  $d(L,R)$*
- *Velocities of gaze ( $v_L, v_R$ )*
- *Delta pupil diameters ( $\Delta\phi_L, \Delta\phi_R$ )*



**Fig. 3.** Examples of (top to bottom) left pupil diameters, delta pupil diameters, velocity of the left eye and distance between the reflections, for subjects 3 (left) and 9 (right) in task C1

Pupil diameters are directly provided by the eye-tracker, while the distance, the velocities and the delta pupil diameters are computed from the raw measurements. Distance between the infrared reflections  $d(L,R)$  (hereafter referred to as distance) is computed as the Euclidean distance between the two coordinates  $(x_L, y_L)$  and  $(x_R, y_R)$  within eye-tracker's camera view. The velocities are calculated by estimating the first time derivative of the Euclidean distance between two consecutive points. The delta pupil diameters are computed using linear regression. Examples of eye-movement data collected in the present study are shown in Figure 3.

### 3.1 Feature Extraction

We consider the time track of a single measurement as a vector  $x = (x_1, x_2, \dots, x_T)$  of dimensionality  $T$ . This vector is reduced to a smaller dimensional space using the following approaches:

- *Fourier spectrum* (FFT) [5]
- *Principal components analysis* (PCA)[4]
- FFT followed by PCA (FFT+PCA)

We utilize a fast Fourier transform for a Hamming-windowed sequence, where the purpose of windowing is to suppress the spectral artifacts arising from the finite length effects. We retain the magnitude spectrum for further processing. As an example, the magnitude spectrum applied to pupil diameter data captures slow and fast variations of the pupil size that are encoded in the lower and higher frequencies, respectively.

The Principal component analysis (PCA) is a widely used dimensionality reduction method. It maps the data onto the directions that maximize the total scatter across all classes. In this study, we apply PCA directly to the raw data as well as to its Fourier spectrum. For PCA, we retain the directions of the largest variance as measured by the leading eigenvalues.

### 3.2 Classification and Fusion

In this study, we limit our experiments to the identification task for simplicity. We consider the different features both individually and in combination, and apply the  $k$ -nearest neighbor classification using Euclidean distance. For fusion we combine the individual distances by their weighted sum. We use *leave-one-out cross-validation* to estimate the error rates [4].

## 4 Experimental Setup

For the experiments, we collected a database of 12 volunteering participants (1 female, 11 male) recruited from research and teaching staff from the authors' department. All participants had normal or corrected-to-normal vision. Three of the participants were wearing eyeglasses.

We used the remote, binocular Tobii ET-1750 eye-tracker (Fig. 2), sampling at the rate of 50Hz. The eye-tracker is built into a 17'' TFT panel so no moving or otherwise disturbing parts can be seen or heard. In a pilot experiment, we noticed that the accuracy was degraded when participants moved their head. These caused some inaccuracies and additional reflections, for instance from eye-glasses. Therefore, we constructed a stand where the subjects rested their chin. The chin-rest was mounted approximately 80 centimeters from the screen, centered in the middle. The accuracy of the eye-tracker was greatly improved by fixing the head position.

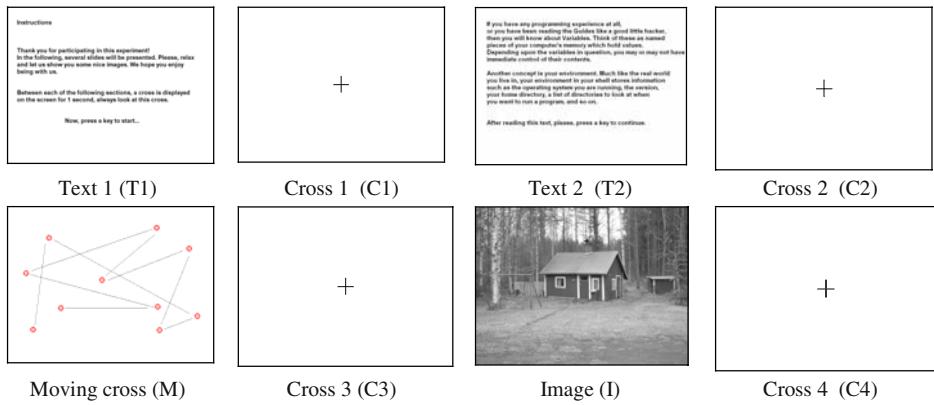
### 4.1 Procedure

Experiments were conducted in a quiet usability laboratory with constant light conditions. The subjects were not informed that the data will be used for a biometric study. Instead, they were told that the data is needed for calibrating a new eye-tracker. Before a recording, a required automatic calibration procedure had to be conducted. If needed, the calibration was repeated to achieve the highest possible accuracy.

After a successful calibration, the participants were instructed to follow the instructions appearing on the screen. The test consisted of several different tasks (see Fig. 4), including text reading (T1, T2), tracking of a moving red cross (M) for 15.9 seconds, and watching a static gray-scaled image (I). After each of these stimuli, a cross was displayed on the middle of the screen (C1, C2, C3, C4) for 1 second. Competitions of the reading tasks were signaled by participants pressing a key, the time to view the static image was not restricted. Completion of the whole task took less than 5 minutes per participant.

#### 4.2 Data Preprocessing

To preprocess the data, erroneous measurements were first removed. These consisted mostly from blinking, and in few cases from unintentional head-movements. After a preliminary experimentation, we realized that the gaze measurements were much noisier than the raw camera data, so we decided to keep only the camera-based measurements. An explanation of this might lie in inaccuracies created during the calibration procedures.



**Fig. 4.** Tasks and their order

### 5 Results

We restricted our experiments to the static crosses tasks (C1-C4) for two reasons. First, as we were interested in studying the inter-person variations, we should factor out the possible influence of the task. Second, this task contained approximately the same amount of data per person, which made the data compatible. For the leave-one-out cross-validation procedure, one of the four tasks was classified at a time using a 3-nearest neighbor classifier; the rest three samples were acting as the representatives for the class. Table 1 reports the number of data vectors recorded for each of the participants during all of the tasks in this experiment.

**Table 1.** The number of data vectors for each of the tasks

Subject: Task ID:	Text		Static cross				Moving cross	Image	
	T1	T2	C1	C2	C3	C4	M	I	Mean
1	643	1201	41	41	28	43	752	1057	423
2	384	936	44	44	50	42	744	549	311
3	546	1146	50	50	49	50	775	455	347
4	534	794	50	48	49	36	748	783	338
5	561	1253	48	49	49	51	701	436	350
6	626	1414	50	50	50	50	787	624	406
7	623	1292	45	41	35	38	779	149	334
8	843	1480	50	49	50	50	771	726	447
9	997	1777	49	50	50	49	790	501	475
10	532	1219	47	44	44	44	741	258	327
11	577	1090	44	44	49	50	782	451	344
12	537	878	50	50	49	50	754	375	306
Mean	617	1207	47	47	46	46	760	530	

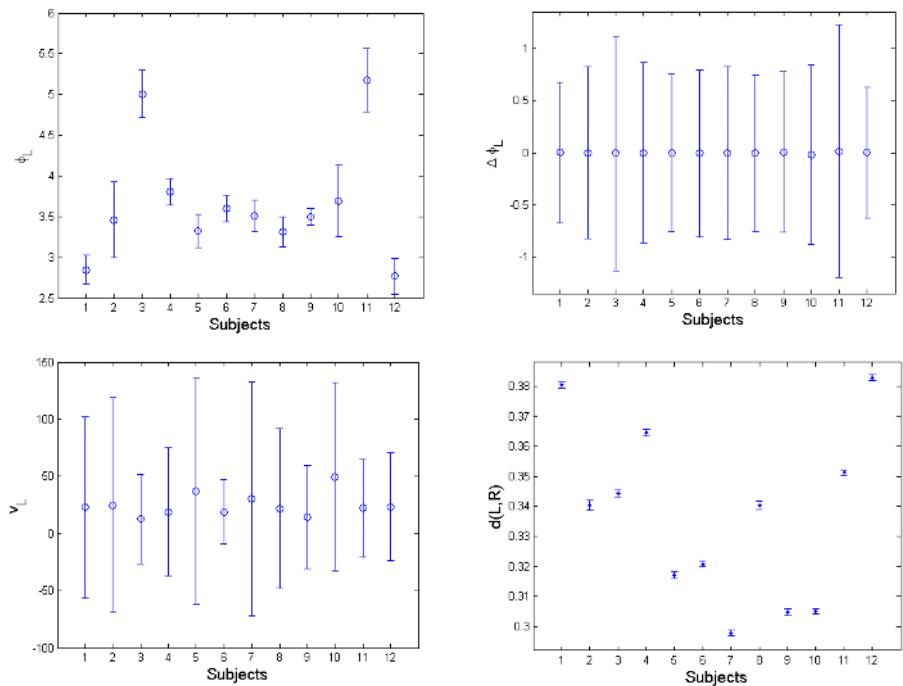
## 5.1 Individual Features

As we were interested to investigate whether there were any individual differences in eye-movement dynamics, we had to remove the static properties from the signal. Thus, for a comparison, we created a *static* user template by taking the time averages for each subject. As long-term statistics, these were expected to carry the information about the physiological properties of the subject's eyes. Figure 5 shows the time-averaged features with their 95 % confidence intervals.

The *dynamic* user templates were formed by considering the time signal as a feature vector as explained in Section 3.1. The identification rates for both the static and dynamic cases are reported in Table 2. For the dynamic features, we studied both the original and mean-removed signals. By removing the long-term mean, we expected the features to discriminate mainly based on their dynamics.

**Table 2.** Identification rates (%) for single category of features

Method	Eye	Mean	Static	Dynamic				
			FFT	PCA		FFT + PCA		
			Original	Mean removed	Original	Mean removed	Original	Mean removed
Pupil	L	31	38	8	32	15	38	8
	R	33	33	13	38	8	33	13
	L+R	38	38	19	46	15	38	17
Delta Pupil	L	4	42	56	44	48	42	56
	R	4	46	56	50	50	46	56
	L+R	8	50	54	50	60	50	54
Velocity	L	17	19	10	15	13	19	10
	R	25	10	6	8	10	10	6
	L+R	21	13	6	13	15	19	8
Distance	-	83	90	6	83	4	90	8



**Fig. 5.** Time-averaged features with their 95 % confidence intervals

## 5.2 Fusion of the Features

By combining complementary features, we expected to improve the accuracy further. After preliminary experiments, we decided to select the combinations as shown in Table 3. The fusion weights were set with an error-and-trial procedure by taking into account the individual performances.

**Table 3.** Identification rates (%) for fusion

Method Features	Fusion Weights	FFT	PCA	FFT + PCA
Pupil + Velocity	0.90 / 0.10	42	42	42
Delta Pupil + Velocity	0.90 / 0.10	46	44	46
Pupil + Distance	0.04 / 0.96	90	90	90
Velocity + Distance	0.10 / 0.90	83	83	83
Pupil + Velocity + Distance	0.10 / 0.05 / 0.85	92	88	92

## 6 Discussion

From the static features, the distance between the eyes shows high discrimination (90%) as expected. The distance can be accurately measured by an eye-tracker and used as a feature for biometric. However, similar measurement could be also obtained from a picture taken by a regular camera as well, as it does not include any dynamics of the eye-movement.

Static features and the original data (without mean removal) perform better compared to the eye dynamics. Nevertheless, the mean-removed features are above the chance level (8.35 %) in most of the cases. From the dynamic features, the delta pupil shows the best performance of 50-60 % and the velocity of the eyes 6-15 %. A possible explanation can be that the stimuli were static and displayed only for 1 second. Considering the feature extraction, FFT and PCA performed equally well and their combination did not improve accuracy further.

Considering the fusion of the features, no further improvement was achieved. The distance provided already an accuracy of 90 %, and dominated the results. The identification rate of the fusion for the dynamic features was around 40-50%. Given that the number of subjects was low, no statistically significant weight selection could be done.

## 7 Conclusions

This paper presents a first step towards using eye-movements as a biometric. We have conducted a case study for investigating the potential of the eye-tracking signal. The distance between eyes turned out to be the most discriminative and stable measurement, yielding identification rate of 90 % for twelve subjects. However, this feature could be measured without an eye-tracking device, and it does not truly reflect the behavioral properties of the eyes. The best dynamic feature was the delta pupil size (60 %), which corresponds to the variation of the pupil size in time. Interestingly, the pupil size itself provides rather weaker discrimination (40 %). Combination of different features gave only marginal improve in accuracy of identification.

In summary, the results indicate that there is discriminatory information in the eye-movements. Considering that both the training and test signals had the duration of 1 second only, the recognition accuracy of 40-90 % can be considered high, especially taking into account the low sampling rate (50 Hz). We expect improvements of the proposed system by having longer training and/or testing data, as well as a higher sampling rate.

As the eye-tracking systems are expected to become more widely available, we expect that they can be integrated into general video-based systems. In our future studies, we intend to create a larger database of recordings and tasks, and to study the inter-session variability of eye-movements as biometric features, i.e. the stability of the features over time. Other important future considerations are the dependence of the recognition accuracy on the task and the proper weight selection for fusion. Ultimately, we should have a feature that is consistent over different tasks.

## References

1. R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to Biometrics*, Springer, 2004.
2. R. Brunelli, D. Falavigna, Person identification using multiple cues, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17 (10): 955-966, October 1995.
3. R. Brunelli, T. Poggio, Face recognition: feature versus templates, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15 (10): 1042-1052, October 1993.
4. R. Duda, P. Hart and D. Stork, *Pattern Classification*, 2<sup>nd</sup> edition, Wiley Interscience, New York, 2000.
5. E.C. Ifeachor and B.W. Lewis, *Digital Signal Processing - a Practical Approach* (2<sup>nd</sup> edition), Pearson Education Limited, Edinburgh Gate, 2002.
6. M.A. Just and P.A. Carpenter. Eye fixations and cognitive processes, *Cognitive Psychology*, 8: 441-480, 1976.
7. K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124: 372-422, 1998.
8. A. Ross, A. Jain, Information fusion in biometrics, *Pattern Recognition Letters* 24, 2115-2125, 2003.
9. P. Verlinde, G. Chollet, M. Achery, Multi-modal identity verification using expert fusion, *Information Fusion* 1 (1): 17-33, Ed. Elsevier, 2000.
10. M. Yang, D. J. Kriegman, N. Ahuja, Detecting faces in images: a survey, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24 (1): 34-58, January 2002.

# Inverse Global Illumination Rendering for Dense Estimation of Surface Reflectance Properties

Takashi Machida<sup>1</sup>, Naokazu Yokoya<sup>2</sup>, and Haruo Takemura<sup>1</sup>

<sup>1</sup> Osaka University, 1-32 Machikaneyama,  
Toyonaka, Osaka 560-0043, Japan

{machida, takemura}@ime.cmc.osaka-u.ac.jp,  
<http://www.lab.ime.cmc.osaka-u.ac.jp/>

<sup>2</sup> Nara Institute of Science and Technology,  
8916-5 Takayama, Ikoma, Nara 630-0192, Japan

**Abstract.** This paper investigates the problem of object surface reflectance modeling, which is sometimes referred to as *inverse reflectometry*, for photorealistic rendering and effective multimedia applications. A number of methods have been developed for estimating object surface reflectance properties in order to render real objects under arbitrary illumination conditions. However, it is still difficult to densely estimate surface reflectance properties faithfully for complex objects with interreflections. This paper describes a new method for densely estimating the non-uniform surface reflectance properties of real objects constructed of convex and concave surfaces. Specifically, we use registered range and surface color texture images obtained by a laser rangefinder. Then, we determine the positions of light sources in order to capture color images to be used in discriminating diffuse and specular reflection components of surface reflection. The proposed method can densely estimate the reflectance parameters of objects with diffuse and specular interreflections based on an inverse global illumination rendering. Experiments are conducted in order to demonstrate the usefulness and the advantage of the proposed methods through comparative study.

## 1 Introduction

*Inverse rendering* is an effective technique to produce a photorealistic image, object geometry, reflectance properties and lighting effect in a scene. In the fields of computer vision and graphics, a number of methods have been developed to estimate reflectance properties from images [1, 2, 3, 4, 5, 6, 7, 8, 9]. These approaches, which are sometimes referred to as *inverse reflectometry*, reproduce the object shape and surface reflectance properties. If the object surface reflectance properties are estimated at once, the virtualized object can be rendered appropriately under arbitrary illumination conditions. In this paper, estimation of object surface reflection in *inverse rendering* framework is focused.

Especially, it is necessary to estimate the incident radiances of surfaces in the scene because the light that any particular surface receives may arrive not only from the light sources but also from the rest of the environment through indirect illumination. The estimation of incident radiances allow to estimate the reflectance properties of the surfaces

in the scene via an iterative optimization procedure, which allows to re-estimate the incident radiances. That is called *Inverse Global Rendering*. Loscos et al. [7] and Drettakis et al. [4] have attempted to estimate object surface reflectance properties based on radiosity equations. Yu et al. [9] have estimated surface reflectance properties of a room from color and geometry data considering both diffuse and specular interreflections based on the inverse global illumination rendering. Boivin et al. [3] have also attempted to estimate surface reflectance properties with considering diffuse interreflections. These methods assume that the surface of interest has uniform reflectance properties. Therefore their algorithms cannot be applied to a non-uniform surface reflectance object.

We have also conducted the study on reflectance estimation based on the radiosity rendering for considering interreflections, which is called *inverse radiosity rendering* [2]. The radiosity algorithm is quite efficient in computing the lighting distribution for a simple model with diffuse materials. However, because the radiosity rendering method considers only diffuse interreflections, when object reflectance properties are estimated it is impossible to eliminate the influence of specular interreflections. In addition, it becomes very costly for complex models and non-diffuse materials. The high computational cost for complex models is due to the fact that the radiosity algorithm computes values for all the patches in the model. Furthermore, there is a problem that the finite mesh representation can be very inaccurate if the mesh is not carefully constructed. Therefore, in the case of using the radiosity for estimating surface reflectance properties, we must have accurate object geometry and efficiently tessellate it.

In this paper, we take notice of the photon mapping [10] that is a global illumination rendering method. The photon mapping can represent diffuse and specular interreflections based on computing emission of photons from a light. Using the photon mapping, we propose a new method for estimating non-uniform reflectance properties of objects with both diffuse and specular interreflections. Additionally, the proposed method can densely estimate reflectance parameters by densely observing both reflection components.

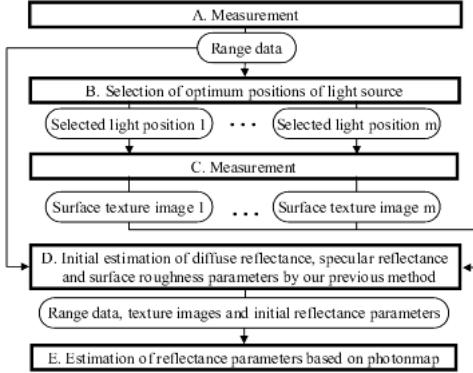
## 2 Reflectance Modeling from Range and Color Images

Figure 1 shows a flow diagram of estimating surface reflectance properties. Our process consists of five parts: measurement of an object (A, C), selection of light source (B), initial estimation of reflectance parameters using our previous method [2] (D), and reflectance estimation using photon mapping (E).

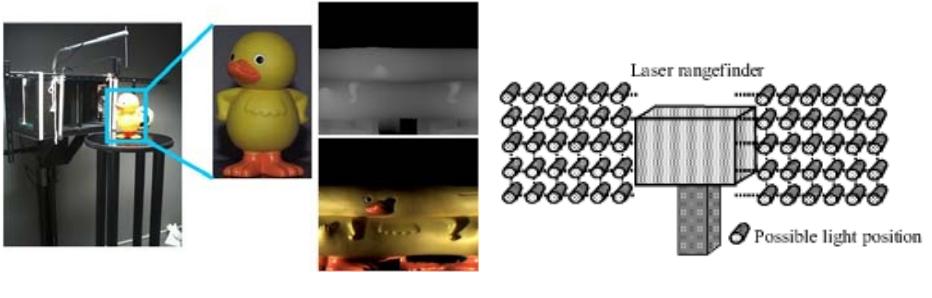
### 2.1 Measurement and Selection of Positions of Light Source

We use a laser rangefinder (Cyberware 3030RGB) with known positions of point light sources and a camera for acquiring surface color images, as shown in Figure 2(a). This system can obtain registered range and surface color texture images at the same time by rotating the rangefinder and the camera around an object.

In the present experimental setup, multiple positions of a light are determined among 60 possible positions prepared around the laser rangefinder and these are two-



**Fig. 1.** Flow diagram of estimating surface reflectance properties



(a) Appearance of 3D-Digitizer and acquired data      (b) Multiple possible light source positions

**Fig. 2.** Experimental setup

dimensionally arranged at the interval of 5 cm as shown in Figure 2(b). The positions of a camera and a light source are calibrated in advance. After optimum light positions are selected, a single light is attached at the selected positions in turn so that the calibration of brightness among multiple lights is not needed. We can also ignore the influence of environmental light by measuring the object in a dark room.

Here, we employ the Torrance-Sparrow model [11] to represent object reflectance properties physically. The Torrance-Sparrow model is given as:

$$i_x = \frac{Y}{D^2} \left\{ P_{dx} \cos \theta_{dx} + \frac{P_{sx}}{\cos \theta_{vx}} \exp\left(-\frac{\theta_{rx}^2}{2\sigma_x^2}\right) \right\}, \quad (1)$$

where  $i_x$  represents an observed intensity corresponding to the surface point  $x$ ,  $i_{dx}$  and  $i_{sx}$  denote the diffuse and specular reflection components, respectively,  $C$  is an attenuation coefficient related to the distance between a point light source and an object surface point, and  $Y$  represents the strength of a light source.  $P_{dx}$ ,  $P_{sx}$  and  $\sigma_x$  are the diffuse reflectance parameter, specular reflectance parameter, and surface roughness parameter, respectively.  $\theta_{dx}$  is the angle between light source vector and surface normal vector,  $\theta_{vx}$  is the angle between viewing vector and surface normal vector, and  $\theta_{rx}$  is the

angle between surface normal vector and half vector. Note that half vector is the vector located halfway between light vector and viewing vector. All vectors are unit vectors.

Dense and independent estimation for non-uniform surface reflectance parameters requires observation of each surface point  $x$  under at least three different lighting conditions: one lighting condition for determining the unknown parameter  $P_{dx}$ , and the other two lighting conditions for acquiring the remaining two unknown parameters  $P_{sx}$  and  $\sigma_x$ . The selection of the optimum positions of the light source in Figure 1(B) is repeated until almost all of the pixels satisfy the three different lighting conditions [1]. As a result of this process, a certain number of light positions, say  $m$ , are selected in order to densely observe both diffuse and specular reflection components.

A texture image is obtained with a selected light position  $p$  ( $p = 1, \dots, m$ ) and consists of  $\gamma$  pixels ( $i_{p1}, \dots, i_{p\gamma}$ ), where  $i_{px}$  means a color intensity of a surface point  $x$ . Each pixel is classified into three types  $T_{diff}$ ,  $T_{spec}$  and  $T_{none}$ .  $T_{diff}$  means a pixel containing only the diffuse reflection component and  $T_{spec}$  means a pixel containing strong specular reflection component.  $T_{none}$  means a pixel which is classified into neither  $T_{diff}$  nor  $T_{spec}$ .

## 2.2 Inverse Photon Mapping for Estimation of Reflectance Parameters

In this paper, we employ the *photon mapping* rendering method for resolving the problem with the inverse radiosity rendering method [2]. By using the inverse photon mapping method, both diffuse and specular interreflections on the object surface can be taken into account. The photon mapping rendering method also does not require tessellated patches due to its pixel base calculations, and thus even if the object has a complicated shape, the photon mapping can render efficiently compared with the radiosity rendering method.

**Photon Mapping.** In the photon mapping rendering method [10], an outgoing radiance  $L$  from a surface point  $x$  is calculated in order to decide the surface color. The following equations form the rendering equations in the photon mapping method.

$$L(x, \vec{\omega}) = L^e(x, \vec{\omega}) + L^r(x, \vec{\omega}), \quad (2)$$

$$L^r(x, \vec{\omega}) = \int_{\Omega} f(x, \vec{\omega}', \vec{\omega}) L^o(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}', \quad (3)$$

$x$  : Surface point

$\vec{n}$  : Unit vector of surface normal at  $x$

$\vec{\omega}$  : Direction from outgoing radiance

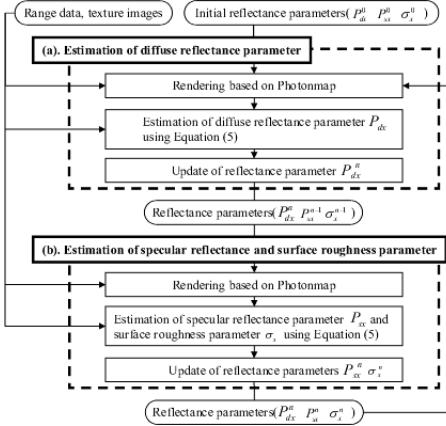
$\vec{\omega}'$  : Direction of incoming radiance

$d\vec{\omega}$  : Differential solid angle

$\Omega$  : Hemisphere of directions

where  $L^e$ ,  $L^r$ ,  $L^o$  and  $f$  are the emitted radiance, the reflected radiance, the incoming radiance, and a BRDF (i.e. the Torrance-Sparrow model), respectively.

Here, the outgoing radiance  $L$  in Equation (2) is equivalent to the reflected radiance  $L^r$  due to the assumption that the underlying objects have no emissions. Equations (2) and (3) are theoretical models. Using Equation (1), the color  $\hat{i}_x$  at surface point  $x$



**Fig. 3.** Flow diagram of estimating surface reflectance properties based on inverse photon mapping

is represented by the following equation, which is referred to as the Ward reflectance model [12]:

$$\begin{aligned} \hat{i}_x &= I_x \left\{ \frac{P_{dx}}{\pi} + P_{sx} \frac{\exp(-\tan^2 \theta_{rx}/\sigma_x^2)}{4\pi\sigma_x^2} \right\} \\ &= I_x \left\{ \frac{P_{dx}}{\pi} + P_{sx} K(\theta_{vx}, \theta_{rx}, \sigma_x) \right\}, \end{aligned} \quad (4)$$

where  $I_x$  is the incoming radiance.  $K(\theta_{vx}, \theta_{rx}, \sigma_x)$  denotes the specular term in Equation (1) and is a non-linear equation, and the other parameters are the same as in Equation (1). In practice, the Ward model described above has five parameters for representing anisotropic object surface reflectance properties. Because the object is assumed to have isotropic reflectance properties in this study, there are three unknown parameters: the diffuse reflectance, the specular reflectance and the surface roughness parameters.  $I_x$  is decided by counting the number of photons that arrive at the point  $x$ . The photon is specifically traced using a Monte Carlo ray tracing method [13]. In this case, the photon is reflected or absorbed according to the reflectance properties, and only the photons that are reflected are traced iteratively.

**Iterative Estimation of Reflectance Parameters.** Figure 3 shows the flow diagram of the present method. As an initial estimation, the reflectance parameters are obtained by our previous method based on *inverse radiosity rendering* [2] in the process (D) in Figure 1. In the initial preprocessing, the diffuse reflectance parameter is estimated based on the inverse radiosity to consider diffuse interreflections. The specular reflectance and the surface roughness parameters are estimated based on the inverse local rendering (i.e. Torrance-Sparrow model) with no consideration of specular interreflections. Here, let  $P_{dx}^{init}$ ,  $P_{sx}^{init}$  and  $\sigma_x^{init}$  be the reflectance parameters obtained in this process. These parameters are used as initial parameters in the next process (E) in Figure 1. In this process, our approach uses the photon mapping method and it is called *inverse photon mapping*. In the following, the detail of reflectance parameter estimation is described.

The reflectance parameter estimation method based on inverse photon mapping is separated into two processes, (a) and (b), as shown in Figure 3. The first process is for the diffuse reflectance parameter estimation ((a) in Figure 3), and the second process is for the estimation of the specular reflectance and surface roughness parameters ((b) in Figure 3). These processes are performed iteratively. In each process, the following equation, which is derived from Equation (4), is minimized at each pixel in the texture image:

$$\begin{aligned} E(P_{dx}, P_{sx}, \sigma_x) &= \sum_{p=1}^q (i_{px} - \widehat{i}_{px})^2 \\ &= \sum_{p=1}^q (L_x - \frac{P_{dx}}{\pi} I_x - P_{sx} K(\theta_{vx}, \theta_{rx}, \sigma_x) I_x)^2, \end{aligned} \quad (5)$$

where  $i_{px}$  is the measured radiance (color intensity) at surface point  $x$  with light source position  $p$ ,  $\widehat{i}_{px}$  is the irradiance that is computed from Equation (4) at surface point  $x$  with light source position  $p$ , and  $q$  denotes the number of selected light positions at surface point  $x$  in categories  $T_{diff}$ ,  $T_{spec}$  and  $T_{none}$  among the selected light positions  $m$ .

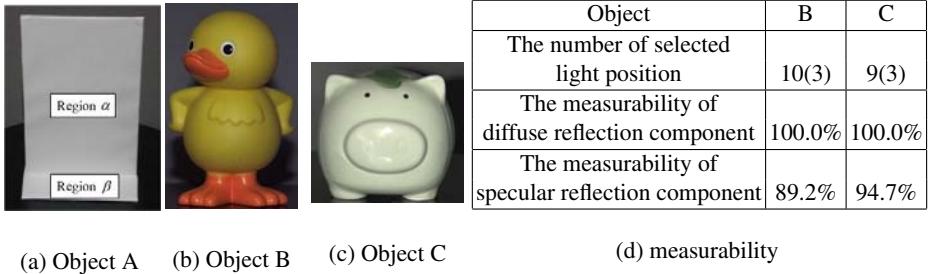
In process (a), the diffuse reflectance parameter  $P_{dx}$  is estimated using a pixel that is categorized as  $T_{diff}$ .  $P_{dx}^{init}$ ,  $P_{sx}^{init}$  and  $\sigma_x^{init}$  are used to compute  $\widehat{i}_{px}$  only at the first iteration. Here, the specular reflection term in Equation (4),  $I_x P_{sx} K(\theta_{vx}, \theta_{rx}, \sigma_x)$ , is set to be 0 because the specular reflection cannot be observed.

In process (b), the specular reflectance  $P_{sx}$  and the surface roughness  $\sigma_x$  parameters are estimated using only pixels that are categorized as  $T_{spec}$  or  $T_{none}$ .  $P_{sx}^{init}$  and  $\sigma_x^{init}$  are again used to compute  $\widehat{i}_{px}$  only at the first iteration. The  $P_{dx}$  estimated above is used in Equation (4). When  $P_{sx}$  and  $\sigma_x$  are estimated, the value of each reflectance parameter is updated, and processes (a) and (b) are iterated  $th$  times. The reflectance parameter is selected when differences between the real and synthetic images is the minimum value among  $th$  samples. Because this is a non-linear equation with unknown parameters, the photon mapping rendering and estimation of surface reflectance parameters is performed iteratively, and the difference between the real image and the synthesized image is minimized (Equation (5)). A number of methods can be used to minimize this equation, and the downhill simplex method is selected for this problem [3].

After the estimation process is complete, the specular reflectance and the surface roughness parameters may not be correct, if the specular reflection component is exceedingly small. Such parameters are interpolated linearly by scanning the texture image horizontally.

### 3 Experimental Results

In experiments, we compare the proposed method with our previous two methods [1, 2] (hereinafter referred to as Method I and Method II). It should be noted that Method I does not consider interreflections at all and Method II considers only diffuse interreflections. See Figure 4 for the test objects used in the experiments. In the following, we first show the result of comparing reflectance parameters estimated by the proposed method (hereafter referred to as Method III) with Methods I and II using Object A. We



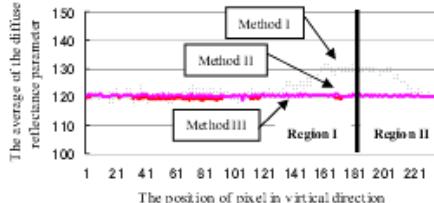
**Fig. 4.** Three objects used in experiment and measurability of both reflection components

then examine the effect of considering interreflections in surface reflectance parameter estimation using Objects B and C with uniform and non-uniform surface properties. Finally, we show rendered images based on reflectance parameters estimated by using Method III. Figure 4(d) shows the number of selected light positions and the measurability of both diffuse and specular reflection components for Objects B and C by using our previous method [1]. The parentheses means the number of selected light positions to observe diffuse reflection component of the whole object.

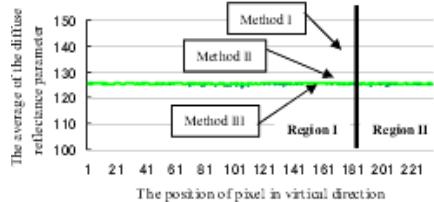
A standard PC (Pentium 4, 3.06 GHz, memory: 2 GB) is used in the following experiments. The number of photons is 2 million, and the proposed algorithm requires approximately four hours to estimate the reflectance parameters of each object. The threshold is fixed at  $th = 50$ .

### 3.1 Removal of an Influence of Both Diffuse and Specular Interreflections

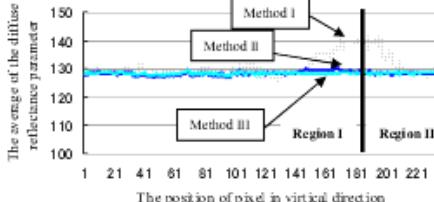
The performance of the proposed method was demonstrated in preliminary experiments using a simple object (Object A). In particular, the present method (Method III) was compared with Methods I and II. Object A consists of two plates (Regions I and II) situated at a 90-degree angle with respect to each other. We have conducted two setups. One is that the same white paper with a uniform diffuse reflectance surface is pasted up on both regions (Setup 1). The other is that the same glossy paper with a uniform reflectance surface is pasted up on both regions (Setup 2). In both setups, the Object A is put on the table obliquely, so that the influence of interreflections can be observed. It is expected that, if interreflections occur, reflectance parameters estimated by the inverse local rendering method must exhibit incorrect values in that part. The results are shown in Figure 5 for Setup 1 and in Figure 6 for Setup 2. Each graph represents the RGB channels of the diffuse reflectance parameter estimated by the three methods. The horizontal axis represents the position of the pixel along the vertical direction of the object, and the vertical axis represents the average diffuse reflectance parameter along the horizontal direction of the object. In Methods I and II, the value of the diffuse reflectance parameter is large around the boundary between Regions I and II due to the influence of interreflections. However, in Method III, the estimated parameter is more stable, indicating that Method III can eliminate the influence of both diffuse and specular interreflections.



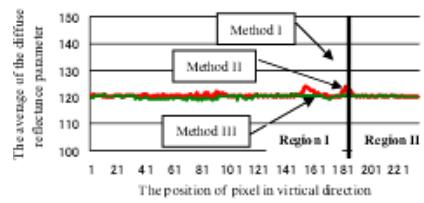
(a) R channel



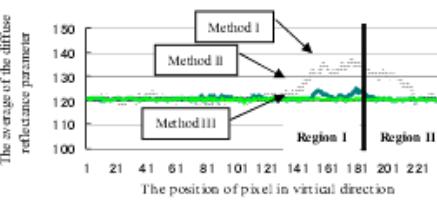
(b) G channel



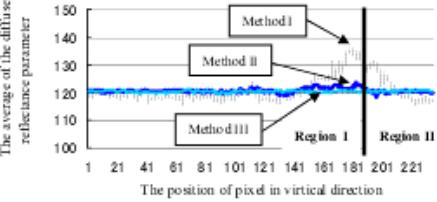
(c) B channel

**Fig. 5.** A comparison among three methods for Object A in Setup 1

(a) R channel



(b) G channel

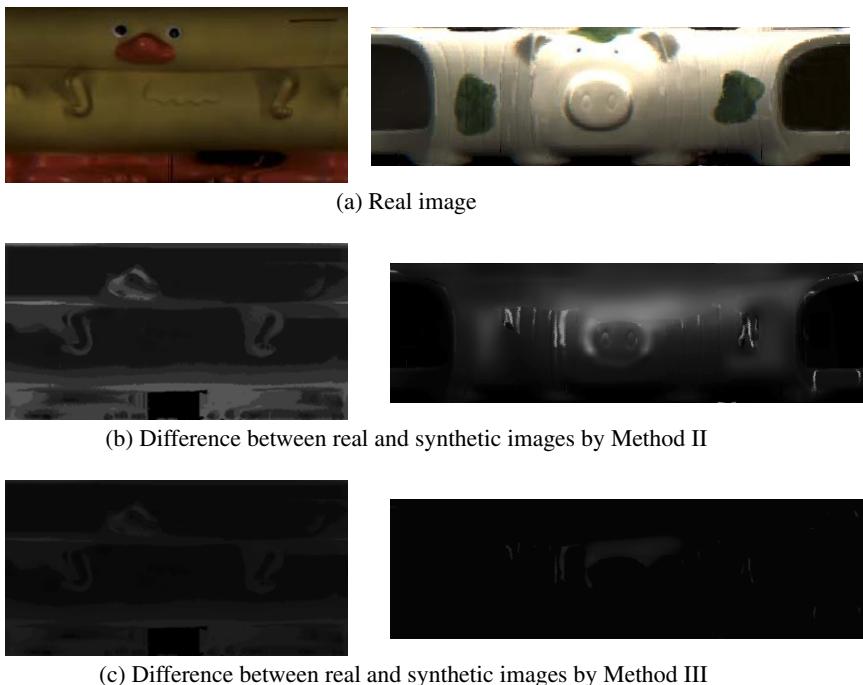


(c) B channel

**Fig. 6.** A comparison among three methods for Object A in Setup 2

### 3.2 Comparison of Rendering Results with Conventional Methods

In the next experiment, we use Objects B and C shown in Figures 4. These objects have non-uniform or uniform diffuse and specular reflectance properties. Figure 7 shows the cylindrical images of real objects and difference images between real and synthetic images (rendered by photon mapping) in Method II and Method III for each test object (Object B and C), respectively. The light position locates at the above on the rangefinder. Synthetic images are rendered using estimated reflectance parameters under the same illumination condition as in the real images. Note that linear interpolation is conducted when the specular reflectance and the surface roughness parameters can not be estimated due to small value of the specular reflection. In Method II, the error due to the influence of specular interreflections is confirmed. Especially, Objects B and C exhibit large errors at the part of inequalities (i.e. duck' legs and pigs' nose). The present method (Method III) does not have such an influence. Additionally, Table 1 shows the average and the variance of differences between real and synthetic images, and the computational time. Method III has much smaller variances than Method II for all the objects. These results show Method III can accurately estimate each reflectance parameter even if diffuse and specular interreflections occur. It is also clear that the *inverse photon mapping rendering* takes more time to estimate the surface reflectance



**Fig. 7.** A comparison of differences between real and synthetic images for Object B and C

**Table 1.** Comparison of the differences between real and synthetic images and computational costs of three methods

Object	Method	Average of differences between real and synthetic images	Variance of differences between real and synthetic images	Computational time [h:m]
B	Method I	28.6	894.7	0:23
	Method II	17.7	501.9	3:51
	Method III	1.11	6.8	4:39
C	Method I	14.2	671.0	0:23
	Method II	8.7	493.3	3:11
	Method III	0.51	3.2	4:43

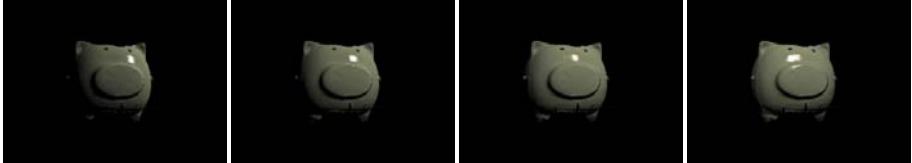
properties than the previous two methods. This is because it takes too much time to render the object image based on estimated reflectance parameters.

### 3.3 Rendering Results with the Photon Mapping Method

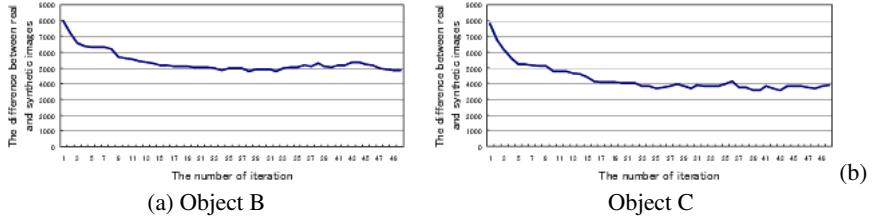
Figure 8 shows rendered images of Objects B and C based on reflectance parameters estimated by the *inverse photon mapping rendering*. It is clear that these images are



(a) Rendering results of Object B



(b) Rendering results of Object C

**Fig. 8.** Rendering of Objects B and C under varying illumination conditions**Fig. 9.** Residuals in minimization with respect to the number of iterations for Objects B and C

photorealistically rendered. However, there are some errors with respect to the geometry. For example, some parts of duck's legs are not rendered and there are spike noises at pig's nose. These errors are due to noise in range images. To solve this problem, it is necessary to interpolate the range data using the data around these parts.

### 3.4 Discussion

Figure 9 shows the relationship between the iterated process and the differences between real and synthetic images by Method III. The vertical axis indicates the sum of differences between the real and synthetic images, while the horizontal axis indicates the number of iterations. Each graph shows that the iterated estimation process decreases the difference between real and synthetic images. However, the minimum difference may not be the global minimum because the proposed iteration method ends when the number of iterations reaches 50. Each graph exhibits the pulsation of residuals and implies that if the number of iterations is more than 50, the residuals may become lower. In this experiment, the reflectance parameters of each object is determined when the number of iterations are 34 and 35, respectively.

Inverse photon mapping rendering has estimated the object surface reflectance properties with both diffuse and specular interreflections. However, the method has a prob-

lem with respect to computational time. This problem is due to the time consumed by the photon mapping rendering. Simply implementing and calculating the photon mapping rendering algorithm on fast graphics hardware (GPU) [14] can solve this problem.

## 4 Conclusions

In this paper, we have proposed a new method for densely estimating non-uniform reflectance properties of real objects. In our approach, we employ the photon mapping method which can represent all of the lighting effects in the real world. This method is independent of complexity of the object geometry unlike the radiosity rendering method. Therefore, the inverse photon mapping method can be applied to objects with various reflectance properties. Experiments have shown that the proposed method exhibits the best performance compared with conventional methods. The advantage of the inverse photon mapping method is that not only diffuse interreflections but also specular interreflections are considered in reflectometry estimation. As a result, estimated parameters can be available without the influence of both interreflections. In future work, we will synthesize a virtualized object into a real world to construct a mixed environment. We will also be concerned with rapid estimation by implementing the algorithm on GPU. Real time estimation has grateful usefulness in computer vision and graphics.

## References

1. Machida, T., Takemura, H., Yokoya, N.: Dense Estimation of Surface Reflectance Properties for Merging Virtualized Objects into Real Images. Proc. 5th Asian Conf. on Computer Vision (ACCV2001) (2002) 688–693
2. Machida, T., Yokoya, N., Takemura, H.: Surface Reflectance Modeling of Real Objects with Interreflections. Proc. 9th IEEE Int. Conf. on Computer Vision (ICCV2003) (2003) 170–177
3. Boivin, S., Gagalowicz, A.: Image-Based Rendering of Diffuse, Specular and Glossy Surfaces from a Single Image. Proc. ACM SIGGRAPH '01 (2001) 107–116
4. Drettakis, G., Robert, L., Bougnoux, S.: Interactive Common Illumination for Computer Augmented reality. Proc. Eurographics Rendering Workshop 1997 (1997) 45–56
5. Lin, S., Lee, S.W.: A Representation of Specular Appearance. Proc. 7th IEEE Int. Conf. on Computer Vision **Vol. 2** (1999) 849–854
6. Lin, S., Lee, S.W.: Estimation of Diffuse and Specular Appearance. Proc. 7th IEEE Int. Conf. on Computer Vision **Vol. 2** (1999) 855–860
7. Loscos, C., Drettakis, G., Robert, L.: Interactive Virtual Relighting of Real Scenes. IEEE Trans. Visualization and Computer Graphics **Vol. 6** (2000) 289–305
8. Sato, Y., Wheeler, M.D., Ikeuchi, K.: Object Shape and Reflectance Modeling from Observation. Proc. ACM SIGGRAPH '97 (1997) 379–387
9. Yu, Y., Debevec, P.E., Malik, J., Hawkins, T.: Inverse Global Illumination: Recovering Reflectance Models of Real Scenes from Photographs. Proc. ACM SIGGRAPH '99 (1999) 215–227
10. Jensen, H.W.: Realistic Image Synthesis Using Photon Mapping. 1st edn. A K Peters, Ltd (2001)

11. Torrance, K.E., Sparrow, E.M.: Theory for Off-specular Reflection from Roughened Surfaces. *Journal of the Optical Society of America* **Vol. 57** (1967) 1105–1114
12. Ward, G.J.: Measuring and Modeling Anisotropic Reflection. *Proc. ACM SIGGRAPH '92* (1992) 265–272
13. Kajiya, J.T.: The Rendering Equation. *Proc. ACM SIGGRAPH '86* (1986) 143–150
14. Purcell, T.J., Donner, C., Cammarano, M., Jensen, H.W., Hanrahan, P.: Photon Mapping on Programmable Graphics Hardware. *Proc. Graphics Hardware* (2003) 265–272

# Multimodal Automatic Indexing for Broadcast Soccer Video

Naoki Uegaki, Masao Izumi, and Kunio Fukunaga

Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

[izumi@cs.osakafu-u.ac.jp](mailto:izumi@cs.osakafu-u.ac.jp),

<http://www.com.cs.osakafu-u.ac.jp/>

**Abstract.** In this paper, we propose a novel method for estimating major soccer scenes from cameraworks and players trajectories based on probabilistic inference, and annotating scene indexes to broadcast soccer videos automatically. In our method, we define relations between cameraworks and scenes, and between players trajectories and scenes by conditional probabilities. Moreover defining temporal relations of scenes by transition probabilities, we represent those relations as dynamic bayesian networks (DBNs). And those probabilities are evaluated by learning parameters of the networks. After extracting the cameraworks and the players trajectories, we compute the posterior probability distribution of scenes, and give the computed results to the soccer video as the scene index. Finally, we discuss the extendibility of the proposal indexing technique in the case of adding ball trajectories and audios.

## 1 Introduction

In recent years, the spread of cable TV, DVD recorders, etc, enabled individuals to record a lot of TV programs easily. But, it needs immense time and efforts to search scenes that you want to watch in large amount of videos. Then, the technology which gives an effective index automatically will be more indispensable from now on.

However, it is difficult to create a general technique of indexing for all kinds of video. On the other hand, domain knowledges which specialized in the specific kind of video are effective information for indexing. Using domain knowledges, there are many researches about scene estimation, event estimation, and indexing. Leonardi et al. [1] estimate major soccer scenes using cameraworks peculiar to soccer videos, for example it pans and zooms rapidly on shoot scenes and corner kick scenes. Xinghua . . . [2] estimate goal events using textures and score boards peculiar to soccer videos. If you use domain knowledges, it becomes easy to indexing. Moreover, it takes into consideration that a video is generally multiple streams of media information, such as audios, texts, and images, it will be thought that the performance of indexing can be raised more by unifying multimodal information. However, unifying multimodal information, we must be careful of there being merit and demerit in each information. For example

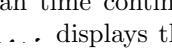
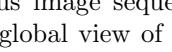
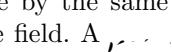
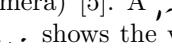
about audios, if words pronounced by the announcer are extracted, it will become an effective information for scene estimation, but word extraction has a high dependence to the announcer. About texts, although it will also become an effective information on scene estimation, having a high dependence to the text creator, and about cameraworks, although a camera performs specific work in each scene, it depends the cameraman. On the other hand, about players and ball trajectories, which are difficult to extract, they don't depend on the announcer and little depend on the cameraman.

In this paper, we propose a novel method for estimating major soccer scenes from cameraworks and players trajectories, and annotating scene indexes to soccer videos automatically. In proposal method, to make the most of the merit of cameraworks and players trajectories, we unify these two informations using bayesian networks. Moreover, for the purpose of considering of temporal transition of scenes, we use dynamic bayesian networks (DBNs) [3],[4] to estimate scenes.

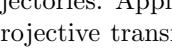
This paper is organized as five sections, the section 2 shows our approach overview for automatic indexing, we discuss scene indexing and retrieval using DBNs in the section 3, the section 4 shows indexing performance to a broadcast soccer video, and the section 5 gives the conclusions.

## 2 Our Approach Overview

### 2.1 Shot Classification

A soccer video can be mainly classified to the following three shots (a shot is an time continuous image sequence by the same camera) [5]. A  displays the global view of the field. A  shows the view of one or a few persons. And an  is denoted as out of the field, mostly audience seats. Applying the method proposed by Uegaki  [6], we classify the soccer video to these three shots. Below this process, since the  includes many important scenes, we focus only on the .

### 2.2 Estimation of Projective Transformation Matrix

It needs projective transformation matrix, which transforms the image coordinate system  $\mathbf{M} = (X \ Y)^T$  to the field coordinate system  $\mathbf{m} = (x \ y)^T$  and vice versa, to extract cameraworks and players trajectories. Applying the method proposed by Nakatsuji  [7], we estimate projective transformation matrix  $P$  (which is  $3 \times 3$  matrix) automatically. Then we can transform the image coordinate system and the field coordinate system each other by the equation (1).

$$\lambda \tilde{\mathbf{M}} = P \tilde{\mathbf{m}} \quad (1)$$

Here,  $\tilde{\mathbf{M}} = (X \ Y \ 1)^T$ ,  $\tilde{\mathbf{m}} = (x \ y \ 1)^T$ , and  $\lambda$  is a scalar.

### 2.3 Extraction of Cameraworks

Cameraworks in  $\dots$  are often performed as the ball position is the center of image, so that the trajectory of the center-of-image projected on the field coordinate system is an important key to estimate scenes. For example, we can observe the trajectory of the center-of-image which goes to the goal area rapidly on shoot scenes. Moreover, the area of the soccer court on the image is an important key, too. For example, we can observe that the area of the soccer court becomes much smaller with fast zoom-in on a corner-kick scene. Accordingly we extract the trajectory of the center-of-image and the area of the soccer court as cameraworks. So we define cameraworks as following vector at frame  $f$ .

$$\mathbf{c}_f = (x_c \ y_c \ v_{x_c} \ v_{y_c} \ a_c \ v_{a_c}) \quad (2)$$

Here,  $\mathbf{m}_c = (x_c \ y_c)$  and  $\mathbf{v}_{\mathbf{m}_c} = (v_{x_c} \ v_{y_c})$  are the position and velocity of the center-of-image on the field coordinate system.  $a_c$  and  $v_{a_c}$  are the area of a soccer court and the velocity of it. This vector is extracted at each frame.

### 2.4 Extraction of Players Trajectories

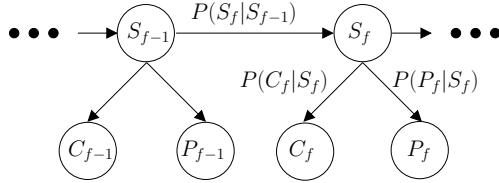
Applying the method proposed by Uegaki  $\dots$  [8], the trajectories of each player on the field coordinate system are extracted. But it is very difficult to get exact trajectory of each player because of frequent overlap of multiple players in a soccer video. On the other hand, it doesn't always need exact trajectory of each player to estimate scenes. For example on shoot scenes, the trajectory of the mean position of players usually goes to the goal area. Accordingly we extract the trajectory of the mean position of players, the variance of the players position, and the number of players as players trajectories. So we define players trajectories as following vector at frame  $f$ .

$$\mathbf{p}_f = (x_p \ y_p \ v_{x_p} \ v_{y_p} \ V_p \ v_{V_p} \ N_p) \quad (3)$$

Here,  $\mathbf{m}_p = (x_p \ y_p)$  and  $\mathbf{v}_{\mathbf{m}_p} = (v_{x_p} \ v_{y_p})$  are the mean position of players and the velocity of it on the field coordinate system.  $V_p$  and  $v_{V_p}$  are the variance of the players position and the velocity of it, and  $N_p$  is the number of players. This vector is extracted at each frame.

### 2.5 Scene Indexing and Retrieval

After extracting the camerawork vector  $\mathbf{c}_f$  and the players trajectory vector  $\mathbf{p}_f$  at each frame, we compute the posterior probability distribution of scenes based on the DBNs (shown in Figure 1), and annotate computed results to the soccer video. Then we retrieve a desirable scene from the indexed video. Next section gives a detailed explanation of it.

**Fig. 1.** The DBNs for scene estimation

### 3 Scene Indexing and Retrieval

Let us estimate six scenes, those are shoot, corner kick, free kick, throw-in, goal kick, and others. This section gives how to estimate these scenes and retrieve them from the indexed video. Figure 1 shows the DBNs for the purpose of estimating these scenes. Node  $S$  called hidden node represents random variables of these scenes, and Table 1 shows the values which node  $S$  can take. Node  $C$  represents random variables of camerawork vectors, and node  $P$  represents random variables of players trajectory vectors. Node  $C$  and  $P$  are called observation node. After observing (extracting) the camerawork vector  $c_f$  and the players trajectory vector  $p_f$  at each frame, we set observation nodes with these observation values, then we compute the posterior probability distribution of scenes and annotate the computed results to the video.

**Table 1.** Target scenes to estimate

Scene name	$S$
shoot	$s^1$
corner kick	$s^2$
free kick	$s^3$
throw-in	$s^4$
goal kick	$s^5$
others	$s^6$

#### 3.1 Clustering Observation Vectors

Extracted camerawork vectors and players trajectory vectors tend to be sparse, because several elements of these vectors are plotted on the soccer field and this field is very vast for these vector elements. Then it will be very difficult to learn observation probability parameters of DBNs in the case of using these vectors directly. In order to reduce the number of observation nodes, we use k-means clustering technique to classify these observation vectors to several classes. If the number of clusters for camerawork vectors is  $K_C$ , camerawork vectors are categorized to  $K_C$  classes  $\{\tilde{c}^1, \tilde{c}^2, \dots, \tilde{c}^{K_C}\}$ , and if the number of clusters for players trajectory vectors is  $K_P$ , players trajectory vectors are categorized to

$K_P$  classes  $\{\tilde{p}^1, \tilde{p}^2, \dots, \tilde{p}^{K_P}\}$ . After clustering these vectors, we use these classes instead of vectors themselves to learn observation probability parameters.

### 3.2 Scene Indexing

Let us focus on a sequence of frames (from frame  $f_1$  to  $f_2$ ). Now the observation sequence from frame  $f_1$  to  $f_2$  ( $o_{f_1:f_2} = \{\tilde{c}_{f_1:f_2}, \tilde{p}_{f_1:f_2}\}$ ) was set to observation nodes, we are able to compute the posterior probability distribution of scenes  $P(S_f|o_{f_1:f_2})$  by the following equation (4) which is called

[4], provided that transition probabilities of scenes  $P(S_f|S_{f-1})$ , observation probabilities of cameraworks  $P(C_f|S_f)$  and observation probabilities of players trajectories  $P(P_f|S_f)$  are respectively defined. These probabilities are defined by parameter learning.

$$\begin{aligned} P(S_f|o_{f_1:f_2}) &= P(S_f|o_{f_1:f}, o_{f+1:f_2}) \\ &= \alpha P(S_f|o_{f_1:f})P(o_{f+1:f_2}|S_f) \end{aligned} \quad (4)$$

Here,  $\alpha$  is a normalizing constant, and

$$\begin{aligned} P(S_f|o_{f_1:f}) &= P(o_f|S_f) \sum_{S_{f-1}} P(S_f|S_{f-1})P(S_{f-1}|o_{f_1:f-1}) \end{aligned}$$

$$\begin{aligned} P(o_{f+1:f_2}|S_f) &= \sum_{S_{f+1}} P(o_{f+1}|S_{f+1})P(o_{f+2:f_2}|S_{f+1})P(S_{f+1}|S_f) \end{aligned}$$

Here,  $P(o_f|S_f)$  is denoted as follows.

$$P(o_f|S_f) = P(\tilde{c}_f|S_f)P(\tilde{p}_f|S_f)$$

Finally, we annotate the computed result to the video.

### 3.3 Parameter Learning

If there are enough training data, we assume that the goal of learning in this case is to find the maximum likelihood estimates (MLEs) of the parameters ( $P(S_f|S_{f-1})$ ,  $P(C_f|S_f)$  and  $P(P_f|S_f)$ ). The log-likelihood of the training set  $D = \{D_1, \dots, D_m\}$  is a sum of terms, one for each node [4]:

$$\begin{aligned} L &= \sum_{i=1}^n \sum_{m=1}^M \log P(X_i|\mathbf{Pa}(X_i), D_m) \\ &= \sum_{ijk} N_{ijk} \log \theta_{ijk} \end{aligned} \quad (5)$$

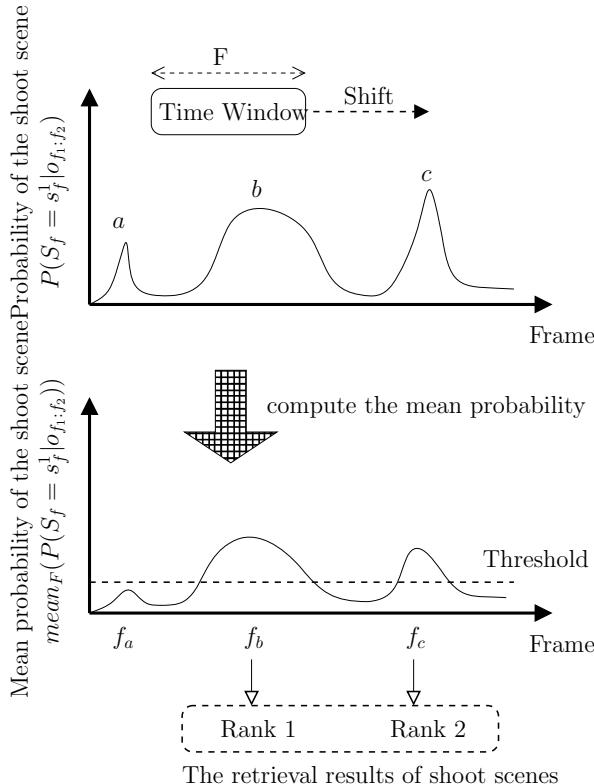
where  $\mathbf{Pa}(X_i)$  are the parents of  $X_i$ ,  $N_{ijk}$  is the number of the events ( $X_i = k, \mathbf{Pa}(X_i) = j$ ) which are seen in the training set, and  $\theta_{ijk}$  is defined as  $\theta_{ijk} = P(X_i = k | \mathbf{Pa}(X_i) = j)$ . The MLEs which maximize the equation (5) is shown in the equation (6).

$$\hat{\theta}_{ijk}^{ML} = \frac{N_{ijk}}{\sum_{k'} N_{ijk'}} \quad (6)$$

However if there are a small number of training cases compared to the number of parameters, the reliability of MLEs decreases. In this case, assuming that any event occurs more than one time, we use the maximum a posteriori (MAP) estimates instead of MLEs. The MAP estimates is shown in the equation (7).

$$\hat{\theta}_{ijk}^{MAP} = \frac{N_{ijk} + 1}{\sum_{k'} N_{ijk'} + Z} \quad (7)$$

where  $Z$  is the amount of values which  $X_i$  can take.



**Fig. 2.** The method of scene retrieval

### 3.4 Scene Retrieval

Let us think the method of retrieving scenes from a video. After computing the posterior probability distribution of scenes  $P(S_f|o_{f_1:f_2})$ , we can get the video with probabilities of scenes. But it has some difficulty to retrieve scenes from this video. For example, suppose that there is a video with probability of the shoot scene as shown in Figure 2. Then the scene around  $b$  should be firstly retrieved as a shoot scene because the probability of the shoot scene around  $b$  rises most stably compared to around  $a$  and  $c$ . To achieve this, we compute the mean probability  $mean_F(P(S_f = s_f^1|o_{f_1:f_2}))$  by shifting the time window which length is  $F$  to  $P(S_f = s_f^1|o_{f_1:f_2})$  and computing the mean probability in the time window as shown in Figure 2. Next we search maximums which are larger than the threshold in  $mean_F(P(S_f = s_f^1|o_{f_1:f_2}))$ , and if  $f_b$  and  $f_c$  are found, since the mean probability at  $f_b$  is larger than at  $f_c$ , then we present the scene around  $f_b$  firstly (Rank 1) and the scene around  $f_c$  secondly (Rank 2) as the retrieval result of shoot scenes. The other scenes are retrieved in a similar way.

## 4 Experiments and Discussions

We applied the proposal method to a broadcast soccer video (resolution is  $320 \times 240$ , frame rate is  $30/s$ ). In this experiment, we prepared a training data of about 18000 frames (about 10 minutes), and after performing the shot classification mentioned in 2.1, there were 5 shoot scenes, 1 corner kick scene, 0 free kick scene, 6 throw-in scenes, and 2 goal kick scenes in  $\dots, \dots, \dots, \dots, \dots, \dots$ . Since there are a small number of training data, observation probabilities were learned using the MAP estimates mentioned in 3.3. On the other hand, transition probabilities were defined manually because the amount of transition probabilities is small (which is  $6 \times 6 = 36$ ). Table 2 shows the transition probabilities, but the observation probabilities are too large to describe in this paper. This time, projective transformation matrix was estimated manually.

**Table 2.** The transition probabilities  $P(S_f|S_{f-1})$

	$s_{f-1}^1$	$s_{f-1}^2$	$s_{f-1}^3$	$s_{f-1}^4$	$s_{f-1}^5$	$s_{f-1}^6$
$s_f^1$	0.90	0.00	0.00	0.00	0.00	0.02
$s_f^2$	0.00	0.90	0.00	0.00	0.00	0.02
$s_f^3$	0.00	0.00	0.90	0.00	0.00	0.02
$s_f^4$	0.00	0.00	0.00	0.90	0.00	0.02
$s_f^5$	0.00	0.00	0.00	0.00	0.90	0.02
$s_f^6$	0.10	0.10	0.10	0.10	0.10	0.90

Table 3 shows the retrieval results of scenes against this trainig data. The average of precision rate of the entire retrieval results was 98%, and the recall rate was 100%.

**Table 3.** Evaluations of the scene retrieval performance for training data

Scene name	Precision	Recall
shoot	0.92	1.00
corner kick	1.00	1.00
free kick	1.00	1.00
throw-in	1.00	1.00
goal kick	1.00	1.00
average	0.98	1.00

## 5 Conclusions

In this paper, we present a novel framework to indexing from multiple informations. This time, we implemented this framework with cameraworks and players trajectories, and confirmed that major soccer scenes can be estimated moderately for the training data only from these two informations. If ball trajectories or audios are extracted, our framework can be easily extended by adding a ball trajectories node or an audios node to the DBNs mentioned in 3, and we expect more effective results of scene indexing and retrieval. Our future works are to introduce ball trajectories and audios, and to prepare more training and testing data.

## References

1. R. Leonardi, P. Migliorati, "Semantic Indexing of Multimedia Documents," *IEEE Multimedia*, pp.44–51, April-June 2002.
2. S. Xinghua, J. Guoying, H. Mei, and X. Guangyou, "Bayesian network based soccer video event detection and retrieval," *Proceedings of SPIE - The International Society for Optical Engineering* 5286, pp. 71-76, 2003.
3. S. Russell, and P. Norvig, "Artificial Intelligence: A Modern Approach(Second Edition)," *Prentice Hall*, Chapter 15, pp.537–581. 2002.
4. Kevin P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," *PhD thesis*, University of California, 2002.
5. J. Assfalg, M. Bertini, C. Colombo, and A. Del Bimbo, "Semantic Annotation of Sports Videos," *IEEE Multimedia*, pp.52–60, April-June 2002.
6. N. Uegaki, K. Nakatsuji, M. Izumi, K. Fukunaga, "Automatic indexing for broadcast soccer video using multiple information," *MIRU2004*, pp.II-329–334, July 2004. (in Japanese)
7. K. Nakatsuji, N. Uegaki, M. Izumi, K. Fukunaga, "Estimation of Players' Position from Image Sequences of Soccer Game TV Program," *IEICE Technical Report*, PRMU2003-214, pp.95-100, Jan. 2004. (in Japanese)
8. N. Uegaki, K. Nakatsuji, M. Izumi, K. Fukunaga, "Tracking of Multiple Players from Soccer Game TV Programs," *IEICE General Conference*, no.D-12-112, p.273, Mar. 2003. (in Japanese)

# Evaluation of the Effect of Input Stimuli on the Quality of Orientation Maps Produced Through Self Organization

A. Ravishankar Rao, Guillermo Cecchi, Charles Peck, and James Kozloski

IBM T.J. Watson Research Center,  
Yorktown Heights, NY 10598, USA  
[ravirao@us.ibm.com](mailto:ravirao@us.ibm.com)

**Abstract.** Self-organized maps have been proposed as a model for the formation of sensory maps in the cerebral cortex. The role of inputs is critical in this process of self-organization. This paper presents a systematic approach to analyzing the relationship between the input ensemble and the quality of self-organization achieved.

We present a method for generating an input stimulus set consisting of images of curved lines. The advantage of this approach is that it allows the user the ability to precisely control the statistics of the input stimuli to visual processing algorithms. Since there is considerable scientific interest in the processing of information in the human visual stream, we specifically address the problem of self-organization of cortical visual areas V1 and V2.

We show that the statistics of the curves generated with our algorithm match the statistics of natural images. We develop a measure of self-organization based on the oriented energy contained in the afferent weights to each cortical unit in the map. We show that as the curvature of the generated lines increases, this measure of self-organization decreases. Furthermore, self-organization using curved lines as stimuli is achieved much more rapidly, as the curve images do not contain as much higher order structure as natural images do.

## 1 Introduction

Several algorithms for the self-organizing formation of orientation columns in the visual cortex have been proposed [3]. The visual cortical area V1 receives retinal input via the thalamus and projects to a higher-level area V2. The role of inputs is critical in the process of self-organization. Hubel . . [5] showed that rather than being genetically predetermined, the structure of cortical visual area V1 undergoes changes depending on the animal's visual experience, especially during the critical period of development. Sharma . . [9] showed that rewiring the retinal output to the auditory cortex instead of the visual cortex resulted in the formation of orientation-selective columns in the auditory cortex. It is thus likely that the same self-organization process is taking place in different areas

of the cortex. The nature of the cortical maps then becomes a function of the inputs received.

Thus, a study of the input space plays an important role in furthering our understanding of cortical maps. In this paper we investigate the presentation of the proper visual inputs that are necessary to achieve the type of self-organization observed in the visual cortex, including areas V1 and V2 [3].

The choice of input stimuli is related to the desired modelling task. Certain classes of inputs are sufficient to model V1. For instance, Bednar [1] used input stimuli consisting of elongated Gaussian blobs. However, if the goal is to model higher order cortical areas such as V2, this class of stimuli is not appropriate. One of the problems with using Gaussian blobs is that they do not provide sufficient information for the self-organization of higher-order maps such as V2, whose elements have wider receptive fields than the elements of V1.

Other researchers have used natural images [6] as inputs to self-organizing algorithms. However, the bulk of such efforts are aimed at modeling area V1, and no higher.

Since the self-organization of cortical maps is critically dependent on the input ensemble, a thorough and systematic study of the dependence of the quality of self-organization with respect to the inputs is desirable. The goal of this paper is to present algorithms for the generation of images of curved lines, which are then used for testing methods for visual processing, such as cortical simulations of the visual stream, including the organization of V1 and V2. The advantage of this approach is that it allows the user the ability to precisely control the statistics of the input stimuli to visual processing algorithms. We show that the statistics of the curves generated with our algorithm are similar to the statistics of natural images as determined by Sigman et al [10]. This way, the self-organizing maps trained by images of artificially generated curved lines should be comparable to the maps generated by training on natural images, as they are both capturing similar input statistical variations [11].

## 2 Background

Our broader research effort is aimed at modeling interactions between V1 and V2 [8], and in general, the self-organization of higher-order maps in the visual stream. The psychophysical observations of Field . . . [4] serve as a testing ground for our model. Field showed that the human visual system is capable of responding to contours of smooth dashed lines in a field of distractors. We are developing a model which provides a neurobiologically grounded explanation for Fields's psychophysical experiment, which involves dashed curved contours. A model that seeks to explain this experiment will only be successful if the appropriate stimuli, namely those containing curved lines, are presented.

Though natural images can be used as input, they must be properly selected to ensure that the right distribution of curved stimuli exists in these images. Furthermore, it is hard to carry out analytics with natural images, as it is difficult to create a mathematical model that precisely describes a natural image. By

using computer graphics-generated stimuli, we have ready access to a precise mathematical model. A systematic study comparing the properties of computer graphics-generated stimuli and natural images for the purposes of training self-organization algorithms has not been done previously. This paper addresses this issue.

A second problem that has received little attention in the literature is the characterization the sensitivity of the self-organization algorithms to the statistics of the inputs they are trained on. We show that self-organization that parallels cortical area V1 in terms of orientation columns occurs only for certain ranges of curvature in the curved line stimuli. This type of analysis will prove useful to researchers in self-organized maps. Though different researchers have tried several approaches to self-organization with different inputs [3], we propose a systematic approach to explore the dependence of self-organization on the inputs.

### 3 Methods

The requirements for the curve-generation algorithm we seek are (1) Generation of realistic curves in terms of their geometry. We use the curvature as a measure of the curve's geometry. (2) The realistic rendering of the curves in terms of their appearance as grayscale images. This implies the use of proper anti-aliasing techniques. (3) The ability to control the spatial distribution of the curves. For instance, it is desirable to ensure that each unit in the cortex has been stimulated an equal number of times by lines of different orientation, such that the distribution of orientations is uniform.

We note that the visual cortex receives retinal inputs via the lateral geniculate nucleus (LGN), whose units perform a center-surround filtering which is similar to edge detection. Thus, the generation of thin contours should suffice to model the output of the LGN to the cortex. The following algorithm is designed to meet the above requirements.

#### 3.1 Generation of Curve Geometry

We used an algorithm that manipulates the curvature explicitly, and can be thought of as a random walk in curvature space. The curvature is defined as  $\kappa = ds/d\theta$  where  $s$  is the arc length and  $\theta$  is the tangent angle. The arc length is integrated as we proceed along the curve. The algorithm starts at an initial point  $(x, y)$  with tangent angle  $\theta$ . The curvature is perturbed by a random value,  $\eta$ , drawn from a uniform distribution in the range  $[-0.5, 0.5]$ . In order to prevent the curvature from changing abruptly, the curve possesses an inertia consisting of the past curvature values. Thus, if the curvature at a time step  $t$  is  $\kappa(t)$ , we maintain the values  $\kappa(t - 1)$  and  $\kappa(t - 2)$ . The curvature update equation is in the form of a moving-average filter as follows.

$$\kappa(t + 1) = 0.5\kappa(t) + 0.3\kappa(t - 1) + 0.2\kappa(t - 2) + \gamma\eta \quad (1)$$

where  $\gamma$  is a gain term that controls the amount of noise added. We used  $\gamma = 0.75$ . We would also like to avoid curves with high curvature, as this would result in a spiral behavior of the curve, resulting in rapid termination. This is achieved by imposing a hard limit on the curvature as follows

$$\kappa(t) = \begin{cases} T & \text{if } \kappa(t) > T \\ -T & \text{if } \kappa(t) < -T \end{cases} \quad (2)$$

The curvature threshold  $T$  is determined to be  $T = 4/W$  where  $W$  is a measure of the size of the image given by the average of its height and width. We make  $T$  a function of  $W$  so that similar looking curves can be generated for any desired image size.

Next, we compute the new slope,  $\mu$  as follows.

$$\mu(t+1) = \mu(t) + s\kappa(t+1) \quad (3)$$

where  $s$  is the arc-length. For the sake of simplicity, we set  $s = 1$ .

In order to give the appearance of a slowly varying curve, the curvature is updated only every  $N$  iterations, say  $N = 4$ .

### 3.2 Rendering of the Curve

Once the geometry of the curve is specified as above, we need to render it in the form of a digital grayscale image. This requires the use of anti-aliasing techniques. We adapt a technique proposed by Wu [12] to render anti-aliased straight lines. Given a point  $(x, y)$  that the curve passes through, with a tangent angle  $\theta$ , we determine the next digital grid point it must pass through. Let us assume the next point in positive direction of  $\theta$  has an  $x$  coordinate of  $x(t+1) = x(t) + \Delta x$ , without loss of generality. Here,  $\Delta x$  is the increment in the  $x$  direction and is set to 1 for simplicity. (An analogous procedure is followed to draw the curve in the negative direction of  $\theta$ ).  $y$  is computed from the slope  $\mu$  obtained from equation 3 as

$$y(t+1) = \mu\Delta x + y(t) \quad (4)$$

Assuming a grayscale value in the range  $[0..1]$  for the image, the next point on the curve, with  $x$  coordinate of  $x(t) + \Delta x$  is rendered as follows. Let  $y_l$  be the floor of  $y(t+1)$  and  $y_u$  be the ceiling of  $y(t+1)$ . The grayscale value  $I$  assigned to point  $(x + \Delta x, y_l)$  is

$$I((x + \Delta x, y_l)) = y_u - y(t+1) \quad (5)$$

and to point  $(x + \Delta x, y_u)$  is

$$I((x + \Delta x, y_u)) = y(t+1) - y_l \quad (6)$$

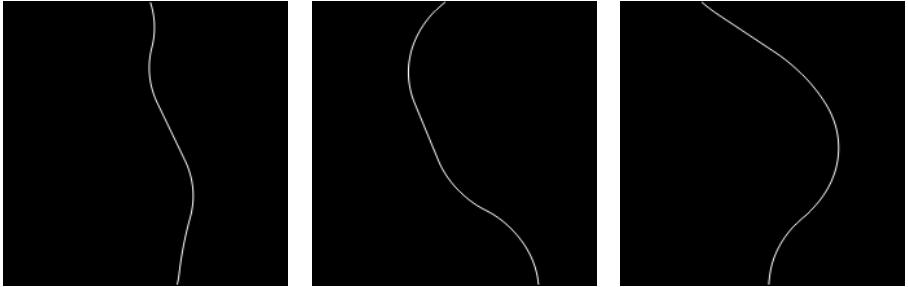
This results in a thin, anti-aliased curved line, which is an appropriate stimulus to use when testing cortical self-organization algorithms.

We can specify the initial condition that the curve passes through a point  $(x_0, y_0)$  and has orientation  $\theta_0$  at this point. The rendering of the curve is performed according to equation 4 in the directions of  $+\theta_0$  and  $-\theta_0$ . This initial condition is important, as it can be used to ensure that every point in an input image has been visited and every orientation about this point has also been covered.

## 4 Experimental Results

The results of applying the above algorithm for curve generation are shown in Fig. 1. The images are of size 200x200. As can be observed, the resulting curves possess a gradual change in curvature, giving them a snake-like appearance.

The ability of the user to specify the initial conditions for the curve are advantageous in that one can ensure that all points in the image have been visited, and all orientations about each point have been rendered.



**Fig. 1.** Sample curves generated by the algorithm

One can vary the rendering of the curve to create dashed lines by using equations 5 and 6 only every  $N$  steps.

We now examine the effect of varying the parametrization of the curves on the self-organization process.

### 4.1 Effect of the Curve Parameters on Self-organization

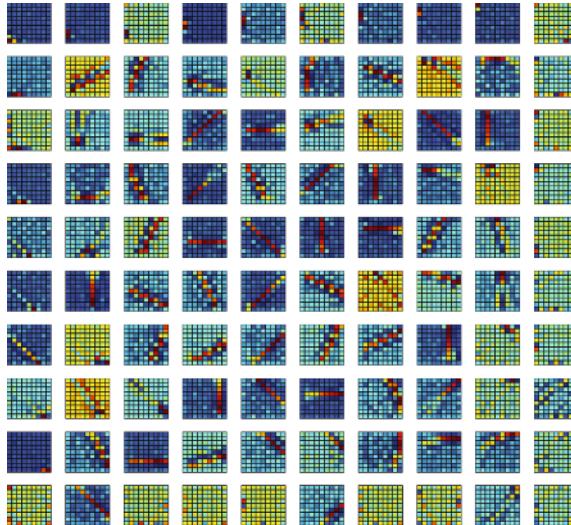
We used the curve generation algorithm described above as input to a self-organization algorithm based on the infomax criterion as developed by Bell and Sejnowski [2]. Consider a network with an input vector  $\mathbf{x}$ , a weight matrix  $\mathbf{W}$ , a bias vector  $\mathbf{w}_0$ , and a squashing function  $g$  which creates the output  $\mathbf{y}$ , where  $\mathbf{y} = g(\mathbf{W}\mathbf{x} + \mathbf{w}_0)$ . Let  $g$  be a sigmoidal function, with  $g(u) = (1 + e^{-u})^{-1}$ . With this form of  $g$ , the learning rules to achieve mutual information maximization between  $\mathbf{x}$  and  $\mathbf{y}$  are

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + (1 - 2\mathbf{y})\mathbf{x}^T \quad (7)$$

$$\Delta\mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y} \quad (8)$$

We apply the above learning rules to a network with a 10x10 input layer and a 10x10 output layer, such that each unit in the input layer is connected to every input in the output layer. Let  $j$  denote an index into the input layer, and  $i$  into the output layer, such that  $i, j \in [1, 100]$ . The weight matrix  $\mathbf{W}$  is thus of size 100x100, where the element  $W_{ij}$  describes the connectivity from

unit  $j$  in the input layer to unit  $i$  in the output layer. The result of training can be visualized through Fig. 2. The weights that connect to the  $i^{th}$  unit can be arranged in a matrix  $\tilde{\mathbf{W}}_i$ , where the entry  $W_i(m, n)$  is the weight from an input at location  $(m, n)$  to the  $i^{th}$  output unit. In our case,  $\tilde{\mathbf{W}}_i$  is a 10x10 matrix. Fig. 2 displays the matrices  $\tilde{\mathbf{W}}_i$  as graylevel images. Note that several of the weight matrices shown have an oriented structure, showing that they are detecting lines of different orientation, at different phases. There are also some weight matrices that do not show such organization.



**Fig. 2.** Result of training using Bell and Sejnowski's algorithm. Each sub-image represents a 10x10 weight matrix for that location in the output layer

In an ideal case, all the weight matrices  $\tilde{\mathbf{W}}_i$  will have oriented structure after sufficient training. On the other hand, if there is poor self-organization, we will see little oriented-selective structure in the weight matrices. In order to quantify the degree of self-organization<sup>1</sup>, we develop a metric,  $d$ , based on the amount of oriented energy in a matrix.  $d$  depends on a shape measure  $s_i$  and an energy measure  $\eta_i$ .

In order to define the shape measure  $s_i$ , we threshold the matrix  $\tilde{\mathbf{W}}_i$  into a binary image using the method of Otsu [7] to yield a region  $R$  of ‘ones’. We

<sup>1</sup> A more complete measure of self-organization will include the topological organization of the orientation-selective weight matrices. For instance, one could measure the correlation amongst neighboring weight matrices. However, for the sake of simplicity we did not develop such a topological measure, which is a topic for future research.

calculate second order central moments,  $\mu_{20}$ ,  $\mu_{02}$  and  $\mu_{11}$  where a central moment of order  $pq$  for the region  $R$  is defined as

$$\mu_{pq} = \frac{1}{A} \sum_{x,y \in R} (x - \bar{x})^p (y - \bar{y})^q \quad (9)$$

where  $A$  is the area of the region  $R$  and  $(\bar{x}, \bar{y})$  is the centroid of  $R$ , and  $p, q = 0, 1, 2, \dots$ . If the elements of the matrix are considered to be a 2D image which has an intrinsic shape, then one can define measures of this shape. The shape measure we use,  $s_i$  is defined by

$$s_i = \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02}} \quad (10)$$

The shape measure  $s_i$  reflects the eccentricity of a region, and varies between 0 for a circle and 1 for a highly-elongated region.

Next, consider the mean squared energy  $\eta_i$  defined by

$$\eta_i = \sum_{m=1}^{m=N} \sum_{n=1}^{n=N} \tilde{\mathbf{W}}_i(m, n)^2 / N^2 \quad (11)$$

The measure of oriented energy is then defined by the following product  $d_i = \eta_i s_i$ . The reason we use such a measure is because the shape measure  $s_i$  is scale invariant, and a weight matrix consisting of a single blob of noise can have high eccentricity, giving rise to the false conclusion that the matrix shows good self-organization. The global measure of self-organization,  $d$ , is defined as the average of the measures  $d_i$ , as  $d = \sum d_i / M$ , where  $M$  is the number of matrices considered, 100 in this case.

## 4.2 Variation of $d$ with the Number of Training Iterations

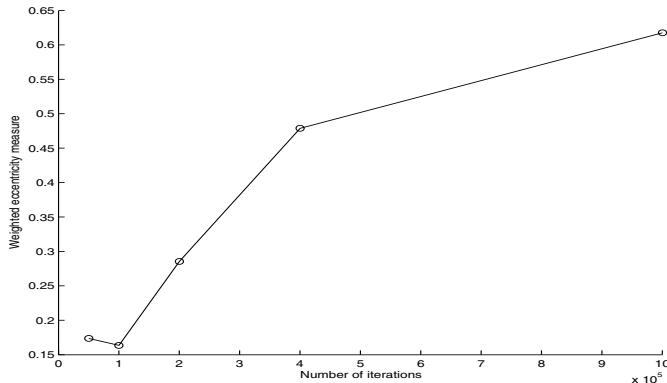
As Fig. 3 shows, the measure  $d$  of self-organization increases as the number of training iterations increases. The rate of increase in the self-organization is higher in the beginning, and slows down with an increasing number of iterations.

## 4.3 Variation of $d$ with Respect to the Curvature Gain

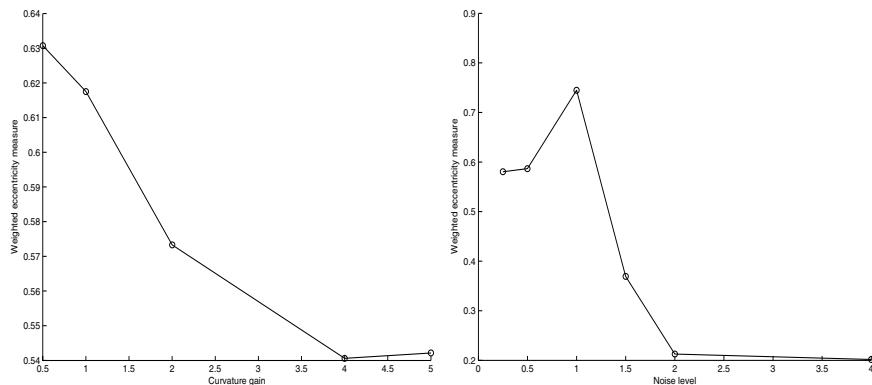
As Fig. 4(a) shows, the measure  $d$  of self-organization decreases with respect to the curvature gain  $\gamma$  in equation 1. This shows that if straight lines are used as stimuli, the map shows better self-organization. However, using straight lines as the sole input form may not result in the correct organization of association fields in V2 [4], which are essential to the line completion experiment described in Section 2. This is because V2 is conjectured to capture information related to curvature of lines [4].

## 4.4 Variation of $d$ with the Amount of Noise in the System

We corrupt each pixel in the the curved line image with additive noise which is drawn from a uniform distribution with zero mean and in the range  $[-\delta, \delta]$ .



**Fig. 3.** Plot of the weighted eccentricity measure,  $d$  versus the number of iterations



**Fig. 4.** (a) Plot of the weighted eccentricity measure,  $d$  versus the curvature gain,  $\gamma$ . A fixed training cycle consisting of 1 million iterations was used for all cases. (b) Plot of the weighted eccentricity measure,  $d$  versus the noise level,  $\delta$ . A fixed training cycle consisting of 1 million iterations was used for all cases

Fig. 4(b) shows that the measure  $d$  of self-organization decreases with increasing noise level,  $\delta$ . It is expected that natural images will contain a fair amount of noise. (For instance, the contours of tree branches may be interrupted by foliage). This shows that if synthetic curved lines are used as stimuli, the resulting map shows better self-organization.

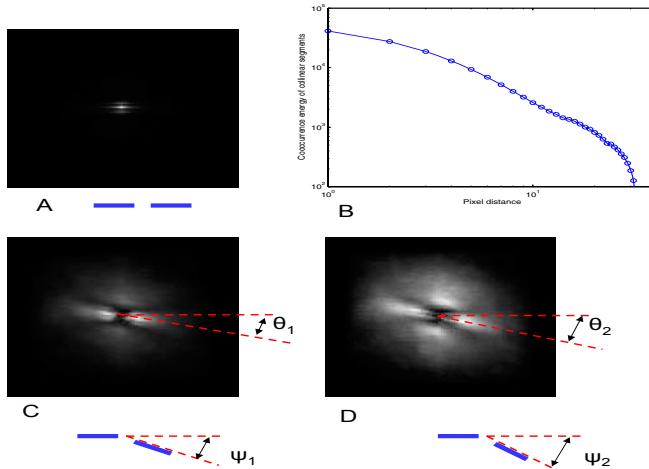
Furthermore, to achieve a comparable level of self-organization as measured by  $d$ , an algorithm using natural images would need to use a much larger number of training cycles, typically by a factor of two times. This can be seen by comparing Fig. 3 and 4(b).

#### 4.5 Comparison of the Curve Statistics with That of Natural Images

We follow the method of Sigman [10] . . . to generate statistics of images containing curved lines drawn by our algorithm. Let  $E(x, y, \phi)$  denote the energy at pixel  $(x, y)$  of an image, where  $\phi$  is the dominant orientation at this point.  $E(x, y, \phi)$  and  $\phi$  are computed using steerable filters as described in [10]. We are interested in the co-occurrence statistics of a line at orientation  $\phi$  with a line of orientation  $\psi$  at a distance  $(\Delta x, \Delta y)$  away. This is expressed as

$$C(\Delta x, \Delta y, \phi, \psi) = \frac{1}{N} \sum_{n=1}^N \int \int E_n(x, y, \phi) E_n(x + \Delta x, y + \Delta y, \psi) dx dy \quad (12)$$

where  $N$  is the total number of images and the integral is calculated for each image. The angles are quantized into 16 equal bins from 0 to  $180^\circ$ . Fig. 5 shows the result of applying this method to a set of 2000 curved lines drawn by the algorithm in Sec. 3.2. Fig. 5(A) shows the co-occurrence matrix  $C(\Delta x, \Delta y, 0, 0)$ , i.e. the co-occurrence of energy for collinear segments. This matrix is displayed as a grayscale 2D image. Fig. 5(B) shows a plot of the logarithm of the distance in pixels on the  $x$  axis versus the logarithm of the collinear energy on the  $y$  axis. This plot clearly shows that the distribution of collinear energy is not an exponential function, but follows a power law, as anticipated by Sigman . . . [10]. Furthermore, collinear interactions extend over long distances.



**Fig. 5.** This figure shows the statistics of energy co-occurrence for various orientations

Fig. 5(C) and (D) explore the phenomenon of co-circularity, which refers to the tendency of pairs of oriented edges in natural images to be co-circular, ie tangent to a common circle [10]. Fig. 5(C) shows  $C(\Delta x, \Delta y, 0, \psi_1 = 22^\circ)$ . The

lobes of maximum intensity have orientation  $\theta_1 = 11^0$ , which is half of  $\psi_1$ , as predicted by the co-circularity rule in Sigman . . . [10]. Similarly, Fig. 5(D) shows  $C(\Delta x, \Delta y, 0, \psi_2 = 34^0)$ . The lobes of maximum intensity in this case have orientation  $\theta_2 = 16.5^0$ , which is approximately half of  $\psi_2$ , as predicted by the co-circularity rule in Sigman . . . [10].

The statistics of the artificially generated curved lines as depicted in Fig. 5 are in agreement with those of natural images as shown in Fig. 2 and Fig. 4 of Sigman . . . [10] in terms of their power law distribution and co-circularity. This shows that using curved lines suffices to train self-organizing maps to achieve organization similar to that in cortical areas V1 and V2.

## 5 Conclusions

In this paper, we developed a systematic approach to analyze the variation of self-organized maps with respect to variations in their input. We presented the benefits of using an artificial curve-generation algorithm for generating inputs to self-organized maps. We developed a measure for the degree of self-organization, and analyzed the behavior of a self-organization method, infomax, with respect to variations in the input. We showed that the amount of curvature of the curved lines affects the degree of self-organization. We also showed that the degree of self-organization decreases with increasing noise in the image.

Thus, it is possible to achieve self-organization in cortical simulations of V1 and V2 faster through the use of the curved line inputs than with images of natural scenes. Since the statistics of the two types of inputs are similar, the fidelity of the maps in representing the statistics of natural scenes is not compromised. The use of the curved line generation algorithm should thus be of great value to researchers who are investigating computational models of the visual cortex.

## References

1. James A. Bednar. *Learning to See: Genetic and Environmental Influences on Visual Development*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, 2002. Technical Report AI-TR-02-294.
2. A.J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, 1995.
3. E. Erwin, K. Obermayer, and K. Schulten. Models of orientation and ocular dominance columns in the visual cortex: A critical comparison. *Neural Computation*, 7(3):425–468, 1995.
4. D. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: Evidence for a local association field. *Vision Research*, 33(2):173–193, 1993.
5. D.H. Hubel, T.N. Wiesel, and S. Levay. Plasticity of ocular dominance columns in monkey striate cortex. *Phil. Trans. R. Soc. Lond. B*, 278:377–409, 1977.
6. A. Hyvärinen, P. O. Hoyer, and J. Hurri. Extensions of ICA as models of natural images and visual processing. pages 963–974, Nara, Japan, April 2003.
7. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems Man and Cybernetics*, 9(1):62–66, 1979.

8. A. R. Rao, C. C. Peck, G. A. Cecchi, and J. R. Kozloski. Contour completion and figure ground separation using self-organizing cortical maps. Society for Neuroscience Annual Meeting, October 2004.
9. J. Sharma, A. Angelucci, and M. Sur. Induction of visual orientation modules in auditory cortex. *Nature*, 404:841 – 847, April 2000.
10. M. Sigman, G. Cecchi, C. Gilbert, and M. Magnasco. On a common circle: natural scenes and gestalt rules. *PNAS*, 98(4):1935–1940, Feb 2001.
11. E. P. Simoncelli and B. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, March 2001.
12. X. Wu. An efficient antialiasing technique. *SIGGRAPH Comput. Graph.*, 25(4):143–152, 1991.

# Modeling Inaccurate Perception: Desynchronization Issues of a *Chaotic Pattern Recognition Neural Network*

Dragos Calitoiu, B. John Oommen, and Dorin Nusbaum

Carleton University, Ottawa, K1S 5B6, Canada

**Abstract.** The usual goal of modeling natural and artificial perception involves determining how a system can extract the object that it perceives from an image which is noisy. The “inverse” of this problem is one of modeling how even a *clear* image can be perceived to be blurred in certain contexts. We propose a chaotic model of Pattern Recognition (PR) for the theory of “blurring”. The paper, which is an extension to a Companion paper [3] demonstrates how one can model blurring from the view point of a chaotic PR system. Unlike the Companion paper in which the chaotic PR system extracts the pattern from the input, this paper shows that the perception can be “blurred” if the dynamics of the chaotic system are modified. We thus propose a formal model, the Mb-AdNN, and present a rigorous analysis using the Routh-Hurwitz criterion and Lyapunov exponents. We also demonstrate, experimentally, the validity of our model by using a numeral dataset.

## 1 Introduction

Sensations can be explained as simple experiences that are caused by physical stimuli. Perceptions are processes by which the brain organizes and interprets sensations [2]. What we know about the world depends on the information gathered by our senses and channeled into our brains. Our brain organizes sensations in such a way that enables us to locate and recognize objects in the environment. However, we are capable of perceiving only those objects to which we devote attention. Perception is not entirely innate, we can learn to perceive [10]. In this paper we consider the “inverse” of the perception problem, namely that of modeling how a *clear* image can be perceived to be blurred in certain contexts. We propose a chaotic model of PR for the theory of “blurring”.

### 1.1 Perceptual Organization and Inaccurate Perception

Sensations are caused by the sensory system, that is stimulated by objects and by events in the environment. The brain organizes sensations in such a way that it can locate and recognize objects and events in the environment, and then detect an activity related to these objects. Understanding perceptual organization is a matter of understanding how one perceives form, depth, and motion. Gestalt

psychologists argue that the key to understanding perception is to identify .. . . . between various parts of a stimulus as the relationship between the lines of a triangle.

Perception, in itself, is not a reception of what is received. It involves a processing of what is received, and an activity of analysis, synthesis and generalization. The essence of perception is the continuous exploration of the objects. This exploration can become, under certain conditions, disorganized and incoherent, which can lead to a high level of perturbation of the operations involved in perception. This paper intends to model the perturbation of perception.

Perception is a conscious process, which is oriented and organized. It involves many actions, such as measuring, decomposing, recomposing, grouping, classifying, transforming and modeling of the objects . As an active mechanism, perception is created, corrected, and verified by actions. When an action, involving objects is limited, the perception is poor, incomplete or erroneous.

The brain is a system with many complex functions and creating an artificial model, which can handle all the functions associated with the brain, is very difficult. Often, the idea of optimizing one function leads to a decline in importance of other functions involved in the overall activity of the model. In this research we study the concept of perception and its “inverse”, which is denoted here as .. . . . or .. . . . . We would like to understand the function of such an artificial recognition system by modelling inaccurate perception. We accomplish this by exploring a model, used for Pattern Recognition (PR), which was presented in the Companion paper [3]. This model can also be used to analyze the effects of anomalies to the process of perception.

To understand this, consider a real life analogy. There are two types of problems which affect the process of perception in visual recognition:

- (i) Issues regarding the stimulus, which in this case involve the resolution of the image, and
- (ii) Issues regarding the recognition system itself. Three reasons why vision can be blurred are:
  - the eye itself has a disease that affects the .. . . . of a stimulus (e.g. by decreasing the visual acuity, for example by myopia, hyperopia, astigmatism);
  - the optical nerve is unhealthy, thus affecting the .. . . . . of a stimulus;
  - the visual cortex is unhealthy, thus affecting the .. . . . of a stimulus.

The second class of problems, which are related to the quality of a recognition system, have not been studied extensively in the modelling of artificial perception systems. This leads us, quite naturally, to the “inverse” problem, which consists of the perturbation of recognition. This is unlike previous studies, which primarily address the degradation of the quality of the stimulus (e.g., by adding noise, decreasing the resolution, decreasing the number of bits per pixel). In this paper we propose a new approach for such modeling, where the ability of a system can be modified by changing its dynamics without changing the input (i.e., the stimulus). We will try to provide an answer to a general question: “Is there a chaotic model for why the system inaccurately processes knowledge even if the stimulus is perfect?”.

## 1.2 Artificial Perception Models Involved in Recognition

The four best approaches for modeling artificial PR systems are: template matching, statistical classification, syntactic or structural recognition, and Artificial Neural Networks (ANNs) [6, 8, 11, 13]. Some popular models of ANNs have been shown to have associative memory and learning [5, 9, 12]. The learning process involves updating the network architecture and modifying the weights between the neurons so that the network can efficiently perform a specific classification or clustering task.

An associative memory permits its user to specify part of a pattern or a key, and to then retrieve the values associated with that pattern. This ability of the brain to “jump” from one memory state to another is one of the hallmarks of the brain, and this phenomenon we want to emulate.

The evidence that indicates the possible relevance of chaos to brain functions was first obtained by Freeman [7] through his clinical work on the large-scale collective behavior of neurons in the perception of olfactory stimuli. Thus, mimicing this identification on a neural network can lead to a new model of pattern recognition,

A Chaotic Neural Network (CNN) with non-fixed weights of the connections between neurons can reside in one of the infinite number of possible states that are allowed by the functions of the network. In cases where a memorized pattern is given as the input to the network, we want the network to resonate with that pattern, generating that pattern with, hopefully, small periodicity, where the actual period of resonance is not of critical importance. Between two consecutive appearances of the memorized pattern, we would like the network to be in one of the non-memorized infinite number of states with a high probability, and in the memorized states with an arbitrary small probability. All this is achieved using our NN, the Modified Adachi Neural Network (M-AdNN), introduced in [3]. How to “juggle” between chaos and periodicity is, indeed, an art, and this is what we attempt to control.

The with the memorized pattern given as input, and the through several states from the infinite set of possible states (even when the memorized pattern is inserted as the input) represent the difference between this kind of PR and the classical types which correspond to the strategies associated with syntactical or statistical pattern recognition.

Classical neural computation does not include Chaotic Neural Networks. By modifying the M-AdNN model, proposed in [3], we formulate a model for the foundation of inaccurate perception, caused by the properties of the system and not by the quality of the stimulus. To our knowledge, such a phenomenon has not been proposed in previous models. The reason for excluding such a phenomenon in the modeling is not due to the lack of evidence of this property in human

---

<sup>1</sup> The term “sympathetic resonance” is used for lack of a better one. Quite simply, all we require is that the trained pattern periodically surfaces as the output of the CNN.

perception; rather it is due to a lack of a mechanism that can “implement” it. Thus, in essence, the contribution of our paper is a formal chaotic model for both perception and blurring, which has been experimentally verified.

## 2 Adachi’s Model for Neural Networks: The AdNN

The AdNN, which actually is a Hopfield-like model, is composed of  $N$  neurons, topologically arranged as a completely connected graph. It is modelled as a dynamical associative memory, by means of the following equations relating internal states  $\eta_i(t)$  and  $\xi_i(y)$ ,  $i = 1..N$ , and the output  $x_i(t)$  as:

$$\begin{aligned} x_i(t+1) &= f(\eta_i(t+1) + \xi_i(t+1)), \\ \eta_i(t+1) &= k_f \eta_i(t) + \sum_{j=1}^N w_{ij} x_j(t), \\ \xi_i(t+1) &= k_r \xi_i(t) - \alpha x_i(t) + a_i. \end{aligned}$$

In the above,  $x_i(t)$  is the output of the neuron, which has an analog value in  $[0,1]$  at the discrete time “.”, and  $f$  is the logistic function with the steepness parameter  $\varepsilon$  satisfying  $f(y) = 1/(1 + \exp(-y/\varepsilon))$ . Additionally,  $k_f$  and  $k_r$  are the decay parameters for the feedback inputs and the refractoriness, respectively,  $w_{ij}$  are the synaptic weights to the  $i^{th}$  constituent neuron from the  $j^{th}$  constituent neuron, and  $a_i$  denotes the temporally constant external inputs to the  $i^{th}$  neuron. Finally, the feedback interconnections are determined according to the following symmetric auto-associative matrix of the stored patterns as in:  $w_{ij} = \frac{1}{p} \sum_{s=1}^p (2x_i^s - 1)(2x_j^s - 1)$  where  $x_i^s$  is the  $i^{th}$  component of the  $s^{th}$  stored pattern.

In his instantiation, Adachi set  $N = 100$ ,  $p = 4$ ,  $k_f = 0.2$  and  $k_r = 0.9$ .

Adachi also investigated the dynamics of the associative network with external stimulations corresponding to one of the stored patterns when the external inputs are applied, and they did this by increasing their bias terms,  $a_i$ , as in:  $a_i = 2 + 6x_i^s$ . The rationale for this can be seen in [1, 3, 4].

## 3 PR Using Chaotic Neural Networks : The M-AdNN

In [3] we proposed a model of CNNs which modified the AdNN as presented below. In each case we give a brief rationale for the modification.

1. The M-AdNN has two global states used for all neurons, which are  $\eta(t)$  and  $\xi(t)$  obeying:

$$\begin{aligned} x_i(t+1) &= f(\eta_i(t+1) + \xi_i(t+1)), \\ \eta_i(t+1) &= k_f \eta_i(t) + \sum_{j=1}^N w_{ij} x_j(t), \\ \xi_i(t+1) &= k_r \xi_i(t) - \alpha x_i(t) + a_i. \end{aligned} \tag{1}$$

After time  $t + 1$ , the global states are updated with the values of  $\eta_N(t + 1)$  and  $\xi_N(t + 1)$  as:  $\eta(t + 1) = \eta_N(t + 1)$  and  $\xi(t + 1) = \xi_N(t + 1)$ .

**Rationale:** At every time instant, when we compute a new internal state, we only use the contents of the memory from the internal state  $\dots \dots N$ . This is in contrast to the AdNN in which the updating at time  $\dots$  uses the internal state values of  $\dots$  the neurons at time  $\dots$ .

2. The weight assignment rule for the M-AdNN is the classical variant:  $w_{ij} = \frac{1}{p} \sum_{s=1}^p (x_i^s)(x_j^s)$ . This again, is in contrast to the AdNN which uses  $w_{ij} = \frac{1}{p} \sum_{s=1}^p (2x_i^s - 1)(2x_j^s - 1)$ .

**Rationale:** We believe that the duration of the transitory process will be short if the level of chaos is low. A simple way to construct hyperchaos with all Lyapunov positive exponents is to couple  $N$  chaotic neurons, and to set the couplings between the neurons to be small when compared with their self-feedbacks, i.e  $w_{ii} \gg w_{ij}(i \neq j)$ . For the M-AdNN, the value of  $w_{i,i}$  will be zero in the identical setting. Clearly, the M-AdNN has a smaller self-feedback effect than the AdNN.

3. The external inputs are applied, in the M-AdNN, by increasing the biases,  $a_i$ , from 0 to unity whenever  $x_i^s = 1$ , keeping the other biases to be 0 whenever  $x_i^s = 0$ .

**Rationale:** In our case  $a_i = x_i^s$ , as opposed to the AdNN in which  $a_i = 2 + 6x_i^s$ . In the case of the latter, the range of inputs is [2, 8] unlike the M-AdCNN for which the range is [0, 1]. Thus, it is expected that the M-AdNN will be more “receptive” to external inputs, and therefore lead to a superior PR system.

The M-AdCNN has the following properties:

**Theorem 1:** The M-AdNN described by Equations (1) is locally unstable whenever  $N^{1/2} > \max(1/k_r, 1/k_f)$ , as demonstrated by its Lyapunov spectrum.

**Theorem 2:** As seen by a Routh-Hurwitz analysis, the  $\dots \dots$  conditions for the M-AdNN described by Equations (1) to be locally unstable, are that  $k_f > 0$  and  $k_r > 0$ .

The proofs for these theorems can be found in the Companion paper [3].

### 3.1 Designing Chaotic PR Systems

As opposed to the accepted models of statistical, syntactical and structural PR, we do not foresee chaotic PR systems to report the identity of testing pattern with a “proclamation” of the class of the tested pattern. Instead, what we are attempting to achieve is to have the chaotic PR system continuously demonstrate chaos as long as there is no pattern to be recognized or whenever a pattern that is not to be recognized is presented. But, when a pattern, which is to be recognized, is presented to the system, we would like the proclamation of the

identity to be made by requiring the chaos level to decrease significantly, and the system to simultaneously possess a strong periodic component, which we refer to as “sympathetic resonance”.

In our testing [3], only one pattern is recognized. If the testing pattern is close enough to two or more “training” patterns, then both (all) may be periodically repeated. This is the rationale for the array count which records the periodicity of all the trained patterns in the output. The procedure for the PR system is in [4].

Instead of invoking Fourier analysis of the output spectrum, we maintained an array of output signals and tested the sequence for periodicity in the time domain.

## 4 A Chaotic NN for Inaccurate Perception: The Mb-AdNN

The rationale for transforming the AdNN into the M-AdNN was essentially one of forcing the former system into chaos, and at the same time attempting to demonstrate a periodic behaviour if we needed to. When it concerns inaccurate perception, it appears as if we have to strike a happy medium between the AdNN and the M-AdNN. For the first part, unlike the AdNN, we would like the new system to demonstrate chaos. But, on the other hand, unlike M-AdNN, we don’t want the system to be “all too periodic” (i.e., or stable) if the trained non-noisy patterns are presented, because we are, indeed, attempting to model the, . . . , or “blurring”.

We shall show how this can be achieved by modifying the dynamics of the control system so as to force the number of non-zero eigenvalues to be  $2m$ , where  $m$  is a “parameter” of the model. In each case, we show how we modify the AdNN (or the M-AdNN) to yield the Mb-AdNN and present the rationale for the modification.

1. The Mb-AdNN has two global states used for the first  $m$  neurons, which are  $\eta(t)$  and  $\xi(t)$  obeying:

$$\begin{aligned} x_i(t+1) &= f(\eta_i(t+1) + \xi_i(t+1)), \\ \eta_i(t+1) &= k_f \eta_i(t) + \sum_{j=1}^N w_{ij} x_j(t), \text{ for } i \leq m, \\ \eta_i(t+1) &= k_f \eta_i(t) + \sum_{j=1}^N w_{ij} x_j(t), \text{ for } i > m, \\ \xi_i(t+1) &= k_r \xi_i(t) - \alpha x(t) + a_i, \text{ for } i \leq m, \\ \xi_i(t+1) &= k_r \xi_i(t) - \alpha x(t) + a_i, \text{ for } i > m. \end{aligned} \quad (2)$$

After time  $t+1$ , the global states are updated with the values of  $\eta_m(t+1)$  and  $\xi_m(t+1)$  as:  $\eta(t+1) = \eta_m(t+1)$  and  $\xi(t+1) = \xi_m(t+1)$ .

**Rationale:** Unlike in the AdNN, at every time instant, when we compute a new internal state vector, we shall use the contents of the memory from  $m$  other internal states. Our current model lies in between the two extremes, namely (a) The AdNN model, where the updating at time . . . uses the

- internal state values of the neurons at time  $t$ , and (b) The M-AdNN model (which demonstrates PR capability), where the updating at time  $t$  uses the internal state values of any of the neurons at time  $t$ .
2. The weight assignment rule for the Mb-AdNN is the same as in M-AdNN and the rationale is the same.
  3. The external inputs are applied, in the Mb-AdNN, as in M-AdNN and the rationale is the same..

We now briefly state the stability issues concerning the Mb-AdNN by using Lyapunov Exponents, and then by using the Routh-Hurwitz Criterion.

**Theorem 3:** The Mb-AdNN described by Equations (2) is locally unstable whenever  $N^{1/2} > \max(1/k_r, 1/k_f)$ , as demonstrated by its Lyapunov spectrum.

The proof for this theorem can be found in [4].

**Remark:** There is a value of  $m$  for which the maximum between  $1/2\ln(N - m + 1) + \ln(k_f)$  and  $1/2\ln(N - m + 1) + \ln(k_f)$  is smaller than 0. For this value of  $m$ , the system is will lean more towards behaving like the AdNN, i.e., possessing no chaos because of the absence of positive Lyapunov exponents. For values of  $m$  smaller than this critical value, the system is will lean towards behaving like the M-AdNN, because it will exhibit chaos by virtue of the corresponding positive Lyapunov exponents. How  $m$  relates to the desynchronization issue is yet unsolved.

**Theorem 4:** As seen by a Routh-Hurwitz analysis, the conditions for the Mb-AdNN described by Equations (2) to be locally unstable, are that  $k_f > 0$  and  $k_r > 0$ .

The proof for this theorem can be found in [4].

We shall now experimentally demonstrate how blurring is achieved as  $m$  increases.

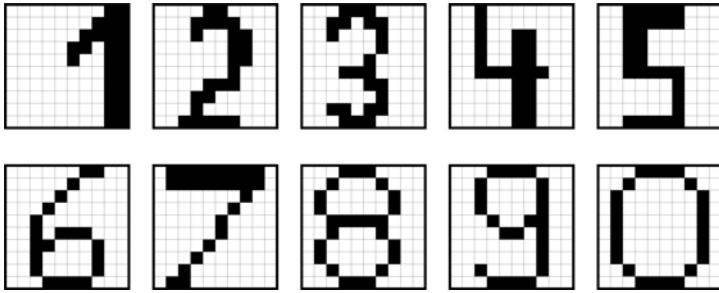
## 5 Experimental Results

As the reader can observe, a chaotic PR system is distinct with respect to its operation and characteristics, in comparison to more traditional systems. In the training phase, we present the system with a set of patterns, and thus by a sequence of simple assignments (as opposed to a sequence of iterative computations), it “learns” the weights of the CNN. The testing involves detecting a periodicity in the system, signaling the user that a learned pattern has occurred, and then inferring what the periodic pattern is. We shall now demonstrate how the latter task is achieved in a . . . .

In a simulation setting, we are not dealing with a real-life chaotic system (as the brain<sup>2</sup>). Indeed, in this case, the output of the CNN is continuously monitored, and a periodic behavior can be observed by studying the frequency

---

<sup>2</sup> How the brain is able to record and recognize such a periodic behaviour amidst chaos is yet unexplained [7].



**Fig. 1.** The set of patterns used in the PR experiments. These were the  $10 \times 10$  bitmaps of the numerals  $0 \dots 9$ . The initial state used was randomly chosen

spectrum, or by processing the outputs as they come, in the time domain. Notice that the latter is an infeasible task, as the number of distinct outputs could be countably infinite. This is a task that the brain, or, in general, a chaotic system, seems to be able to do, quite easily, and in multiple ways. However, since we have to work with sequential machines, to demonstrate the periodicity, we have opted compare the output patterns with the various trained patterns. Whenever the distance between the output pattern and ... trained pattern is less than a predefined threshold, we mark that time instant with a distinct marker characterized by the class of that particular pattern. The question of determining the periodicity of a pattern is now merely one of determining the periodicity of...<sup>3</sup>.

We conducted numerous experiments on the Adachi dataset [1] and other datasets. However, in the interest of space, we report the results of training/testing on a numeral dataset described below. The training set had 10 patterns, which consisted of  $10 \times 10$  bit-maps of the numerals  $0 \dots 9$  (see Figure 1). The parameters used for Equations (1) were  $N = 100$  neurons,  $\varepsilon = 0.00015$ ,  $k_f = 0.2$  and  $k_r = 0.9$ .

**PR Capabilities:** The trained M-AdNN, demonstrated a periodic response when non-noisy external stimuli were applied, after an initial non-periodic transient phase. The transient phase was... short - its mean length was 23.1 time units and most of the transitory phases were of length 24 units. The actual length of the transient phase in each case is given in Table 1. The system resonated sympathetically with the input pattern, with a fairly small periodicity.

**Blurring Effects:** We now examine the “blurring” phenomenon demonstrated by the Mb-AdNN. When  $m = 1$ , the Mb-AdNN degenerates to the M-AdNN

---

<sup>3</sup> The unique characteristic of such a PR system is that each trained pattern has a unique attractor. When a testing pattern is fed into the system, the system converges to the attractor that best characterizes it. The difficult question, **for which we welcome suggestions**, is one of knowing which attractor it falls on [3].

**Table 1.** The transitory phase (named “Transient”) and the periodicity for Mb-AdNN as the value of  $m$  is increased from 1. The testing samples were the *exact* non-noisy versions of the original “numerals” training set. Notice that the chaotic PR phenomenon decreases as  $m$  increases. It disappears after  $m = 8$

Pattern		$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
1	Transient	15	15	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	7,15	7,15	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2	Transient	24	24	24	11	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	25	12	$\infty$	$\infty$	$\infty$	$\infty$
3	Transient	24	24	24	29	16	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	25	25	12	$\infty$	$\infty$	$\infty$
4	Transient	24	24	24	11	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	25	12	$\infty$	$\infty$	$\infty$	$\infty$
5	Transient	24	24	15	107	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	7,15	120	$\infty$	$\infty$	$\infty$	$\infty$
6	Transient	24	24	24	11	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	27	27	25	12	$\infty$	$\infty$	$\infty$	$\infty$
7	Transient	24	24	24	29	31	31	31	42
	Periodicity	26	26	25	25	32	32	32	7,15
8	Transient	24	24	24	29	16	16	16	$\infty$
	Periodicity	26	26	25	25	12	12	12	$\infty$
9	Transient	24	24	24	11	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	26	12	$\infty$	$\infty$	$\infty$	$\infty$
10	Transient	24	24	24	11	$\infty$	$\infty$	$\infty$	$\infty$
	Periodicity	26	26	26	12	$\infty$	$\infty$	$\infty$	$\infty$

leading to a system that exhibits PR, as shown in [3]. Thereafter, as  $m$  increases, this PR capabilities decreases in a “smooth” or “gradual” manner when it concerns the number of patterns recognized. But, for the other patterns, the degeneration of the PR capabilities is . . , - the system “suddenly” changes from being able to recognize a pattern to not being able recognize it.

For  $m = 2$  (with four eigenvalues different from zero) the Mb-AdNN continued to demonstrates PR capabilities for all the ten patterns. The mean length of the transient phase was 24 time units, and the average periodicity was 23.1 time units. As soon as  $m$  was increased to 3 (with six eigenvalues different from zero) the system displayed chaos, but failed to be periodic when the numeral ‘1’ is presented. The number of classes for which the periodic behaviour was demonstrated decreases to 3 for  $m = 5$  and keeps falling till the unity value when  $m = 8$ . For example, when  $m = 5$ , the system recognized only the patterns corresponding to the numerals ‘3’, ‘7’ and ‘8’, and then when  $m = 7$ , the system recognized only the patterns corresponding to the numerals ‘7’ and ‘8’. We also emphasize that the periodicity for the same pattern class sometimes changed as the value of  $m$  changed. Thus, the periodicity (for  $m = 7$ ) for the numeral ‘7’ was 32 time units, and for the numeral ‘8’ was 12 time units; on the other hand, the periodicity (for  $m = 8$ ) for ‘7’ was a double cycle of periodicity 7 and

15. Observe that the number of patterns for which such a periodic behavior is displayed systematically decreases as  $m$  is increased, indicating the phenomenon of increased inaccurate perception or “blurring” even though the input is exact (i.e., non-noisy).

## 6 Conclusion

In this paper, we have studied the problem which is the “inverse” problem of Pattern Recognition (PR), namely that of modeling how even a ..... image can be perceived to be blurred in certain contexts. To our knowledge, there is no solution to this in the literature other than for the oversimplified model in which the true image is garbled with ..... by the perceiver himself. In this paper we have proposed a chaotic model PR for the theory of “blurring”. The paper, which is a extension to a companion paper [3] showed how we can model blurring from the view point of a chaotic PR system. Unlike the companion paper in which the chaotic PR system extracted the pattern from the input, in this case we showed that ..... the perception can be “blurred” if the dynamics of the chaotic system are modified. We thus propose a formal model for chaotic “blurring”, and present a rigorous analysis using the Routh-Hurwitz criterion and Lyapunov exponents. We also experimentally demonstrated the validity of our model by using a numeral dataset.

## References

1. Adachi, M., Aihara, K.: Associative Dynamics in a Chaotic Neural Network. *Neural Networks* **10** (1997) 83–98
2. Buskist, W., Gerbing, D. W.: *Psychology: Boundaries and Frontiers*. Harpper Collins (1990)
3. Calitoiu, D., Oommen, B.J., Nussbaum, D.: Periodicity and Stability Issues of a *Chaotic* Pattern Recognition Neural Network. *Submitted for Publication*. A preliminary version of this paper will be presented at CORES'2005, the 2005 Conference on Computer Recognition Systems, Wroclaw, Poland
4. Calitoiu, D., Oommen, B.J., Nussbaum, D.: Modeling Inaccurate Perception: Desynchronization Issues of a *Chaotic* Pattern Recognition Neural Network. *Submitted for Publication*, Unabridged version of this paper (2005)
5. Fausett, L.: *Fundamentals of Neural Networks*. Prentice Hall (1994)
6. Friedman, F., Kandel, A.: *Introduction to Pattern Recognition, statistical, structural, neural and fuzzy logic approaches*. World Scientific (1999)
7. Freeman, W.J.: Tutorial in neurobiology: From single neurons to brain chaos. *International Journal of Bifurcation and Chaos*, **2** (1992) 451–482.
8. Fukunaga, K.: *Introduction to Statistical Pattern Recog*. Academic Press (1990)
9. Kohonen, T.: *Self-Organizing Maps*. Springer, Berlin (1997)
10. Piaget, J.: *The origins of intelligence in children*. International Univ. Press (1952)
11. Ripley, B.: *Pattern Recog. and Neural Networks*. Cambridge Univ. Press (1996)
12. Schurmann, J: *Pattern classification, a unified view of statistical and neural approaches*. John Wiley and Sons, New York (1996)
13. Theodoridis, S., Koutroumbas, K.: *Pattern recognition*. Academic Press (1999).

# A High-Reliability, High-Resolution Method for Land Cover Classification into Forest and Non-forest

Roger Trias-Sanz<sup>1,2</sup> and Didier Boldo<sup>1</sup>

<sup>1</sup> Institut Géographique National,  
2/4, avenue Pasteur, 94165 Saint-Mandé, France

Roger.Trias.Sanz@ieee.org

<sup>2</sup> SIP-CRIP5, Université René Descartes,  
45, rue des Saints-Pères, 75006 Paris, France

**Abstract.** We present several methods for per-region land-cover classification based on distances on probability distributions and whole-region probabilities. We present results on using this method for locating forest areas in high-resolution aerial images with very high reliability, achieving more than 95% accuracy, using raw radiometric channels as well as derived color and texture features. Region boundaries are obtained from a multi-scale hierarchical segmentation or from a registration of cadastral maps.

## 1 Introduction

Several research projects are underway at the French national mapping agency (IGN) which attempt to partially or fully automate the production of fine-scale topographical maps. As part of this research effort, the goal of the Salade project is to produce a high-reliability land cover classification of rural areas in France, using high-resolution (50cm) orthorectified digital imagery with color and near-infrared channels, and cadastral maps.

The goal is to be able to fully automate interpretation of easy areas, and to use human interpreters only in complicated regions. Thus, considered classification algorithms must also produce a confidence map which indicates the algorithm's confidence in the produced classification, for each region. The classification must be very accurate where the algorithm gives high confidence values, but may be incorrect in some areas as long as the algorithm shows its lack of confidence for these areas. As an intermediate result in this research, we present in this paper several methods for classifying pixels into forest and non-forest in high-resolution aerial images, with very high reliability.

Typical classification systems classify each pixel separately using classifiers such as  $k$ -nearest neighbors, neural networks, or maximum a posteriori (MAP) classification. The raw radiometry vector for each pixel is used as the input for the classifiers. The class map—the output, which gives the class for each image pixel—is usually noisy. In some cases, the class map is geometrically regularized using ad hoc methods, median filtering, or probabilistic relaxation [1, 2] in order to reduce noise. Other algorithms incorporate contextual constraints in the classification process itself by means of Markov Random

Fields [3, 4], and are notoriously slow. If an a priori segmentation of the image into homogeneous regions is available, per-region classification algorithms exist that give the majority class for all pixels in a region [5, 6].

The algorithms presented here are of the per-region kind. Regions are obtained either from the registration of geometrically imprecise cadastral maps onto the images [7, 8], or from a fine image segmentation if cadastre data is not available. They are assumed to be homogeneous, that is to say, they contain only one land cover class. The probability distribution for the input vectors within each region is compared to the distributions for each terrain class created in a training phase, either using a distance on probability distributions or by a probabilistic approach. That distance or probability is itself used as a measure of the algorithm's confidence in the classification.

By using per-region classification we avoid the noise problem of per-pixel classifiers. Availability of prior region data —in the form of registered cadastral maps— means that we do not introduce any artificial regularity as may be the case with MRF-based classifiers or probabilistic relaxation regularization.

In the remainder of this article we present the training procedure (section 2.1) and the classification algorithms (section 2.2). In section 3 we show a thorough evaluation of each algorithm on two real test areas.

## 2 Algorithm

Classification involves three steps. First, a partition of the image to be classified is obtained. The cadastre, which partitions terrain into cadastre regions, can be used, but it should be registered to the image (see [7] or [8]) because edges of cadastre regions are geometrically imprecise and do not exactly correspond to actual edges in a land cover class map. Alternatively, a partition can be obtained by simply running a segmentation algorithm configured to give a segmentation with small regions; there is a risk of obtaining biased results in the classification confidence measures if the segmentation uses the same image channels as the classification, which is why using the cadastre is a better option. If a cadastre region is not homogeneous, but contains more than one land cover class, confidence will be very low for whatever single class is given for the region, and the problem may be detected.

Next, probability models are estimated for each terrain class. This involves manually defining, in a training image, a set of polygons for each terrain class, constructing a histogram for the radiometries —or other input features such as texture or derived color channels— of pixels in a class, selecting a probability model for each class, and estimating the parameters for these models, using the radiometry histograms. Generalization occurs at this point, by preferring a worse-fitting but simpler model over a more complex, better-fitting one. This is discussed in section 2.1.

Finally, each region in the image to be classified is processed in turn. The histogram of the region is constructed, and compared to the probability models for each terrain class. The most probable class, or that with the probability model closest to the region's histogram, is selected for that region. This is discussed in section 2.2.

## 2.1 Estimation

Let  $X = (x_1, \dots, x_N)$  be the sequence of values of all  $N$  training pixels of a certain land cover class, in arbitrary order. For a  $d$ -channel image,  $x_i \in D \subset \mathbb{R}^d$ . Let  $x_{ij}$  denote the  $j$ -th component of  $x_i$ . Let  $\bar{\mu}$  be the sample mean of  $X$ ,  $\bar{\mathbf{S}}$  its sample covariance matrix,  $\bar{s}_{nm}$  be the elements of  $\bar{\mathbf{S}}$ , and  $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$  the set of terrain classes in our problem. In this step the parameters for several probability models are estimated by from the sample  $X$ . In addition, a prior probability for class  $\omega_i$ ,  $p(\omega_i)$ , is calculated as the ratio of training pixels of that class to the total number of training pixels —region size is assumed not to depend on terrain class. The model which best balances complexity and quality of fit to the data is selected for each terrain class.

**Model Estimation.** In the implementation used for the evaluation of section 3, we provide, in addition to a standard  $d$ -dimensional Gaussian model, the following probability models. Specific applications may require the definition of additional models.

*Laplacian Random Variable.* A 1-dimensional Laplacian random variable  $V$  of parameters  $\lambda$  and  $m$  has probability density function  $v(x) = \frac{1}{2\lambda} e^{-|x-m|/\lambda}$ . The higher-dimensional version is not easily found in the literature; we can define one as follows: Let  $A = (A_1, \dots, A_d)$  be a vector of independent 1-dimensional Laplacian random variables of mean zero ( $m = 0$ ) and  $\lambda = 1$  (and therefore variance = 2). The random variable  $W = \mathbf{M} \cdot A + \mu$ , where  $\mathbf{M}$  is an invertible  $d \times d$  matrix and  $\mu \in \mathbb{R}^d$ , is a  $d$ -dimensional Laplacian random variable. Its probability density function  $w(x)$  is

$$w(x) = \frac{1}{2^d |\det \mathbf{M}|} \cdot e^{-\|\mathbf{M}^{-1}(x-\mu)\|_1}, \quad (1)$$

where  $\|\cdot\|_1$  is the  $L_1$  norm,  $\|(x_1, x_2, \dots, x_n)\|_1 = \sum_{i=1}^n |x_i|$ . Its mean is  $E(W) = \mu$  and its covariance matrix is  $\text{cov } W = 2 \cdot \mathbf{M} \cdot \mathbf{M}^T$ , where  $\mathbf{M}^T$  denotes matrix transposition. Estimates  $\hat{\mu}$  and  $\hat{\mathbf{M}}$  for its parameters  $\mu$  and  $\mathbf{M}$  can be calculated from the sample's mean  $\bar{\mu}$  and covariance  $\bar{\mathbf{S}}$  —using a Cholesky decomposition— with

$$\hat{\mu} = \bar{\mu}, \quad 2 \cdot \hat{\mathbf{M}} \cdot \hat{\mathbf{M}}^T = \bar{\mathbf{S}}. \quad (2)$$

*Rectangular Uniform Random Variable.* A  $d$ -dimensional random variable  $V$  with constant probability over a rectangular support  $[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_d, b_d] \subset \mathbb{R}^d$ , and zero elsewhere. Its  $a$  and  $b$  parameters are related to its mean and covariance as  $E(V_j) = (b_j - a_j)/2$  and  $E((V_j - E(V_j))^2) = ((b_j - a_j)^2)/12$ , and we can therefore give estimates  $\hat{a}$  and  $\hat{b}$  for parameters  $a$  and  $b$  from the sample  $X$  as

$$\hat{a} = \bar{\mu} - w/2, \quad \hat{b} = \bar{\mu} + w/2, \quad \text{where } w_i = \sqrt{12 \bar{s}_{ii}}, \quad w = (w_1, \dots, w_d). \quad (3)$$

*Raw Non-parametric Model.* We can also estimate a “non-parametric” random variable whose probability density function follows the distribution histogram of the sample  $X$ . Let's partition the data domain  $D$  into equally-shaped histogram bins,  $\{D_i\}_i$ , with  $\cup_i D_i = D$  and  $\forall i \neq j. D_i \cap D_j = \emptyset$ . Let's construct a relative frequency histogram  $h$  from  $X$  by counting the number of values in  $X$  that belong to the  $i$ -th histogram bin,

for each  $i$ :  $h_i = \text{card}\{j : x_j \in D_i\}/N$ . We have  $\sum_i h_i = 1$ . We can define a random variable  $V$  with values in  $D$  with probability density function  $v(x)$

$$v(x) = h_i \quad \text{with } i \text{ such that } x \in D_i. \quad (4)$$

In this implementation, several such models are estimated for different partitions of  $D$ .

*Kernel Density Estimation.* With raw non-parametric estimation, even if the underlying density is known to be smooth, a large sample size is necessary in order to obtain an estimation which has both a high number of histogram bins and a high count in each histogram bin. *Kernel density estimation* [9, 10] can be used to overcome this problem, by convolving a histogram  $h_i$  with many bins by a suitable  $d$ -dimensional kernel —in this implementation, Gaussian kernels of different variances.

**Model Selection.** Finally, one probability model must be selected from those estimated in the previous step. We cannot simply select the model that best fits the sample, because this will tend to select models with a high number of degrees of freedom, and cause *overfitting*. To avoid this we will select, for each land cover class, the estimated model with the lowest value of the *Bayes Information Criterion*, or BIC [11], a measure which balances the fit to the data and a measure of the model complexity, defined as

$$\text{BIC} = -2 \ln p(X | M) + p \ln N, \quad (5)$$

where  $p(X | M)$  is the probability of the observations given the model,  $p$  is the number of parameters in the model (the complexity), and  $N$  is the number of observations.

## 2.2 Classification

Let  $X = (x_1, \dots, x_N)$  be the sequence of values of all  $N$  pixels in a certain region to be classified, taken in arbitrary order. Using the estimation procedures in the last step, for each class  $\omega_i$  we have an estimated random variable  $V_{\omega_i}$  with probability density function  $v_{\omega_i}$ . The goal of the this step is to assign a class to the region  $X$ ,  $\kappa(X) \in \Omega$ , and to obtain a measure of confidence for that classification,  $c(X)$ , which should be higher for more confident decisions. In this we assume that the pixel values in a region depend only on the class of that region but not on the class of other regions.

**Classification by Distribution Distance.** Let's partition the data domain  $D$  into equally-shaped histogram bins,  $\{D_i\}_i$ , with  $\cup_i D_i = D$  and  $\forall i \neq j. D_i \cap D_j = \emptyset$ . Let's construct a relative histogram  $h$  from  $X$  by counting the number of values in  $X$  that belong to the  $i$ -th histogram bin, for each  $i$ :  $h_i = \text{card}\{j : x_j \in D_i\}/N$ . In our implementation,  $d$ -dimensional square bins, each of side 8, were chosen.

Distribution distance-based classification algorithms depend on the definition of  $d(X, V)$ , a dissimilarity measure between a distribution  $X$  and a random variable  $V$  (not necessarily a distance in the mathematical sense). For a given dissimilarity, the class for region  $X$  is the one closest to it, and we propose to use as confidence measure the dissimilarity measure itself:

$$\kappa(X) = \operatorname{argmin}_{\omega \in \Omega} d(X, V_\omega), \quad c(X) = - \min_{\omega \in \Omega} d(X, V_\omega). \quad (6)$$

One such measure is the Kullback-Leibler divergence, also called the relative entropy. The Kullback-Leibler divergence  $d$  between a sample  $X$  and a random variable  $V$  of probability density function  $v$  is

$$d(X, V) = \mathbb{E}_X (X, V) = 2 \sum_i (h_i \log h_i - h_i \log p(V \in D_i)), \quad (7)$$

where, of course,  $p(V \in D_i) = \int_{z \in D_i} v(z) dz$ . Another measure is the  $\chi^2$  statistic, commonly used in the  $\chi^2$  test [12],

$$d(X, V) = \chi^2(X, V) = N \sum_i \frac{(h_i - p(V \in D_i))^2}{p(V \in D_i)}, \quad (8)$$

omitting from the sum any term with  $h_i = p(V \in D_i) = 0$ . However, the  $\chi^2$  statistic depends linearly on  $N$ , the number of pixels in the region. While this does not affect the choice of class for the region, it may affect the confidence measures. If we refuse to classify regions where the confidence measure falls under a certain threshold, we might keep an incorrect classification for a small region while rejecting a correct classification for a large region. The problem is avoided by using this confidence measure instead:

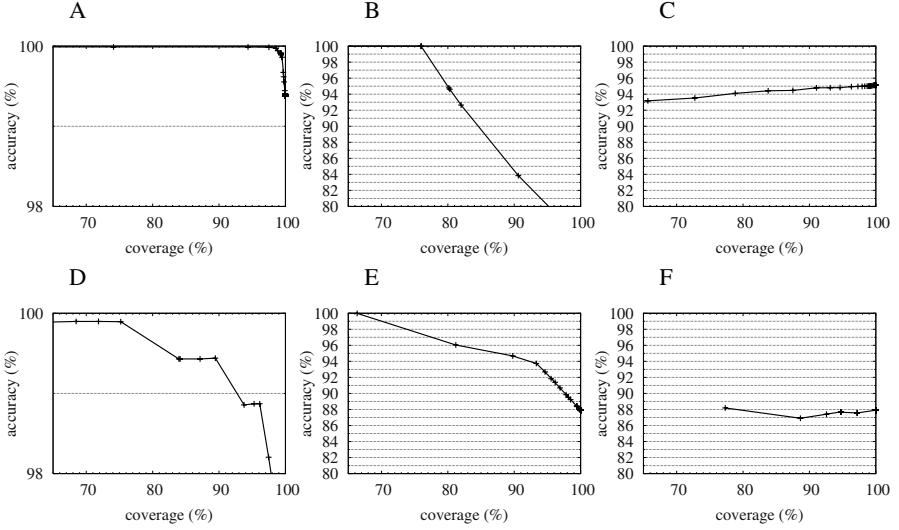
$$c(X) = - \min_{\omega \in \Omega} \frac{1}{N} \chi^2(X, V_\omega). \quad (9)$$

**Classification by Per-region Probability.** Let us further assume that, in a region, the value for a pixel is independent from the values of other pixels —as in standard per-pixel classification; existing dependencies can be captured through the use of texture features. We can calculate the probability of all pixels belonging to a certain class, and choose the class giving maximum probability. From a maximum a posteriori approach (MAP), this gives

$$\begin{aligned} \kappa_{\text{MAP}}(X) &= \operatorname{argmax}_{\omega \in \Omega} p(\omega | X) = \operatorname{argmax}_{\omega \in \Omega} \frac{p(X | \omega) p(\omega)}{p(X)} = \operatorname{argmax}_{\omega \in \Omega} p(X | \omega) p(\omega) \\ &= \operatorname{argmax}_{\omega \in \Omega} p(\omega) \prod_{n=1}^N p(x_n | \omega) = \operatorname{argmax}_{\omega \in \Omega} p(\omega) \prod_{n=1}^N v_\omega(x_n). \end{aligned} \quad (10)$$

There is open debate on whether MAP or maximum likelihood (ML) approaches should be used. Briefly, MAP gives less probability to classes found rarely in the training data. ML gives equal chances to all classes, which is not optimal for risk minimization but may be desired in some cases. From an ML approach we obtain the classification

$$\kappa_{\text{ML}}(X) = \operatorname{argmax}_{\omega \in \Omega} p(X | \omega) = \operatorname{argmax}_{\omega \in \Omega} \prod_{n=1}^N p(V_\omega = x_n) = \operatorname{argmax}_{\omega \in \Omega} \prod_{n=1}^N v_\omega(x_n). \quad (11)$$



**Fig. 1.** Several graphs of classification accuracy as a function of coverage showing excellent (A, D), correct (B, E), and undesired behaviors (C, F). A: Saint-Leger, segmentation regions, MAP<sup>1/N</sup>, “forest” class. B: Toulouse, cadastre regions, MAP<sup>1/N</sup>, “forest” class. C: Toulouse, segmentation, MAP<sup>1/N</sup>, global accuracy. D: Saint-Leger, cadastre,  $KL$ , global accuracy. E: Toulouse, segmentation, MAP<sup>1/N</sup>, “forest” class. F: Toulouse, segmentation, MAP<sup>1/N</sup>, “forest” class, using  $c(X) = \max_{\omega \in \Omega} p(\omega) \prod_{n=1}^N v_\omega(x_n)$  instead

We propose to use as confidence measures

$$c_{\text{MAP}}(X) = \max_{\omega \in \Omega} p(\omega) \left( \prod_{n=1}^N v_\omega(x_n) \right)^{\frac{1}{N}}, \quad c_{\text{ML}}(X) = \max_{\omega \in \Omega} \left( \prod_{n=1}^N v_\omega(x_n) \right)^{\frac{1}{N}}. \quad (12)$$

where, as with equations 8 and 9, the  $1/N$  exponent makes the measure independent of region size (compare Fig. 1 E, with the  $1/N$ , with Fig. 1 F, without).

### 3 Evaluation

We have implemented the classification algorithms described in section 2.2, and the estimation methods described in section 2.1. Using these, we present results for two sites. In the first, *Saint-Leger*, we use aerial images at 80cm resolution, resampled to 50cm, with red, green, and blue channels. In the second, *Toulouse*, we use digital aerial images at 20cm resolution, also resampled to 50cm, with red, green, blue, and near-infrared channels (the images before resampling were not available to us). Training and evaluation polygons are defined on both sites. Classification is not only performed on the raw radiometric channels but also on derived color and textural channels —the latter because few channels are available but spatial resolution is very high: Following suggestions in [13], for Saint-Leger we use the raw green channel, the second Karhunen-Loëve transformed channel (from [14],  $k_2 = (\text{red} - \text{blue})/2$ ), and the local proportion

of pixels whose gradient module is over a threshold [15]. For Toulouse we use the green channel, the same gradient-based measure, and the normalized difference vegetation index (NDVI).

Using the training polygons, probability models are obtained, for each site, for terrain classes “field”, “forest”, “road”, “bare soil”, “orchard”, “quarry”, “uncultivated field”, “other vegetation”, “water”, “sand”, “vineyard”, and “shrubs”. For each class, several models are computed as described in section 2.1 and the best fit—according to the Bayes Information Criterion—is selected. Regions are obtained either from a fine segmentation of the images [16] or by registering geometrically imprecise cadastral maps onto the images [7, 8]. Per-region classification is then performed using the methods described in section 2.2. Finally, all classes except the “forest” class are merged into a single “non-forest” class. This is done after classification, and not at the modeling stage, because non-forest classes have quite distinct signatures which would be more difficult to model jointly.

For baseline comparison, we have also implemented a per-pixel MAP classifier, a per-pixel ML classifier, and a per-region classifier based on majority vote [5] from per-pixel MAP:

$$\kappa_{\text{MAP}}(\{x\}) = \underset{\omega \in \Omega}{\operatorname{argmax}} p(\omega) v_\omega(x), \quad (\text{per-pixel MAP}), \quad (13)$$

$$\kappa_{\text{ML}}(\{x\}) = \underset{\omega \in \Omega}{\operatorname{argmax}} v_\omega(x), \quad (\text{per-pixel ML}), \quad (14)$$

$$\kappa_{\text{maj-MAP}}(X) = \underset{\omega \in \Omega}{\operatorname{argmax}} \operatorname{card}\{x_i \in X : \kappa_{\text{MAP}}(x_i) = \omega\}, \quad (\text{majority MAP}). \quad (15)$$

Evaluation results are summarized in table 1. This shows the ratio of correctly classified pixels at 100% coverage,—accepting all classifications regardless of their confidence value—for the Saint-Leger and Toulouse sites using the presented classification algorithms, and using either an image oversegmentation or registered cadastre data as input regions to the per-region classifiers. The Saint-Leger test site has an area of 5.7 km<sup>2</sup>. The Toulouse test site has an area of 37.6 km<sup>2</sup>. However, since only 7.0 km<sup>2</sup> of cadastre data is available for the Toulouse site, results for that site using registered cadastre edges as regions are less significant; results under “Toulouse cadastre” in table 1 correspond to the

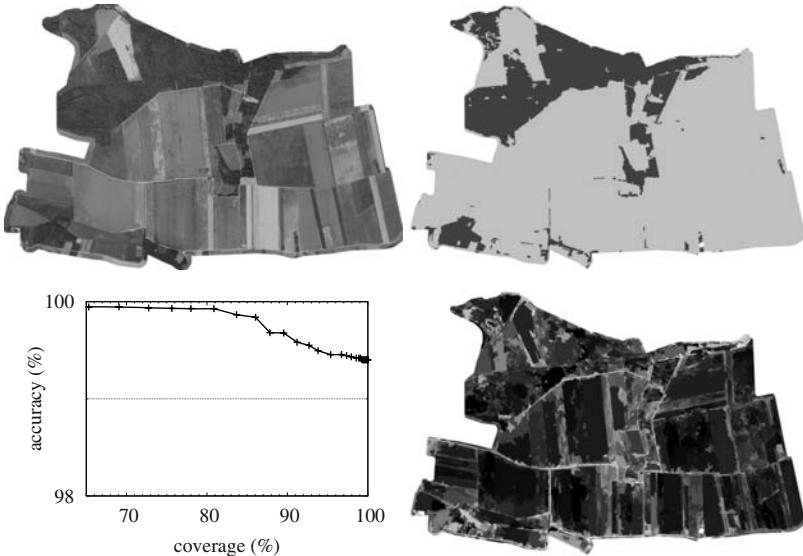
**Table 1.** Classification accuracies —correctly classified pixels in the evaluation set—at full coverage

Method:	Test site: Regions	Saint-Leger		Toulouse	
		segmentation	cadastre	segmentation	cadastre
KL (eq. 7)		99.4%	97.0%	93.5%	85.7%
$\chi^2$ (eq. 8)		99.4%	97.6%	87.6%	63.3%
$\chi^2/N$ (eq. 9)		99.4%	97.6%	87.6%	63.1%
MAP <sup>1/N</sup> (eqs. 10 and 12)		99.4%	97.0%	95.1%	90.1%
ML <sup>1/N</sup> (eqs. 11 and 12)		99.4%	97.0%	95.0%	90.1%
Per-pixel MAP (eq. 13)		98.2%	98.2%	92.8%	92.0%
Per-pixel ML (eq. 14)		97.6%	97.3%	90.0%	86.8%
Majority vote MAP (eq. 15)		99.4%	97.8%	93.1%	84.8%

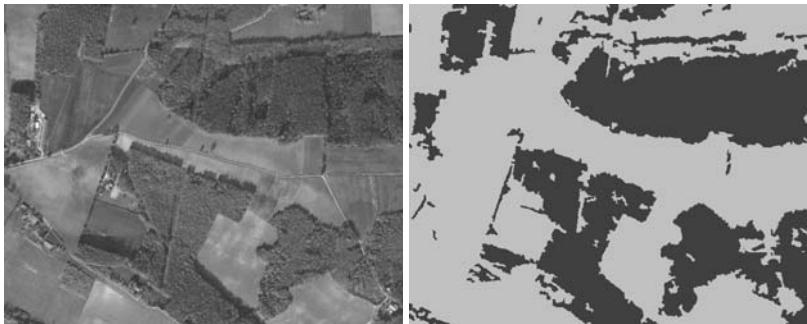
subset of the Toulouse site where cadastre data is available. Using regions extracted from cadastral maps instead of an over-segmentation of the source image gives slightly worse accuracy, but output regions tend to have higher geometrical quality. Execution time is approximately 40 s/km<sup>2</sup> on a 2.4GHz Intel Pentium 4 processor for a single algorithm.

For the Saint-Leger site, all algorithms except per-pixel MAP and ML achieve better than 99% pixel-wise accuracy —accuracy is defined as the ratio of correctly classified pixels in the evaluation set over the total number of pixels in the evaluation set. For the much more diverse and complex Toulouse site, the best algorithm, MAP<sup>1/N</sup>, achieves an accuracy of over 95%; this is slightly better than the per-pixel MAP and ML methods, and has the advantage of not having the dotted noise typical of per-pixel classification. It is also slightly better than a majority vote of a per-pixel MAP classification. For comparison, recent papers —which often use per-pixel classification on images of lower spatial resolution but higher spectral resolution— report accuracies around 85%–90% [17, 18, 19].

In some cases, accuracy increases with decreasing coverage —as a larger fraction of the least-confidently-classified regions is rejected, accuracy in the accepted regions increases. This indicates meaningful confidence values. In that case we can obtain higher accuracies at the cost of decreased coverage. In other cases, however, the calculated confidence measure appears not to be indicative of the quality of classification. In Fig. 1 we show several confidence measure graphs, some showing the desired behavior (increasing accuracy as coverage decreases) and some not. For example, in the Toulouse



**Fig. 2.** Saint-Leger site, classified using the  $\chi^2/N$  method (eq. 9) and segmentation regions. Top left: input image. Top right: class map (white: unclassified; dark gray: forest; light gray: non-forest). Bottom right: confidence values (darker is higher confidence). Bottom left: classification accuracy (% correctly classified pixels) as a function of coverage (% classified pixels, depending on a threshold on the classification confidence value)



**Fig. 3.** A portion of the Toulouse site, classified using the  $\text{MAP}^{1/N}$  method (eqs. 10 and 12) and segmentation regions. Left: input image. Right: class map (white: unclassified; dark gray: forest; light gray: non-forest). Graphs of accuracy versus coverage are given in Fig. 1 C and E

site, using regions obtained from an over-segmentation, and the  $\text{MAP}^{1/N}$ , the accuracy of the “forest” class can be increased from 87.9% at 100% coverage to 96.0% at 81% coverage (Fig. 1 E).

Figures 2 and 3 show, for the Saint-Leger site and a portion of the Toulouse site respectively, the source image and the class map using segmentation-derived regions and the  $\chi^2/N$  and  $\text{MAP}^{1/N}$  algorithms. For Saint-Leger, we also show the classification accuracy as a function of coverage, and the confidence map; accuracy indeed increases with decreasing coverage, indicating that the confidence measure is meaningful there. For the Toulouse site, global confidence measures—shown in Fig. 1 C—are not meaningful and cannot be used to obtain higher accuracy at lower coverage; confidence classes for certain classes, however, behave correctly—such as that for the “forest” class shown in Fig. 1 E.

## 4 Conclusion

We have presented several methods for land-cover classification. To avoid the noise usually associated with per-pixel classification, we chose per-region algorithms. However, in contrast to the more common per-region classification method, which involves a per-pixel classification followed by a majority vote, our methods use distances on probability distributions and whole-region probabilities.

These methods are evaluated for locating forest areas in high-resolution color and color-infrared aerial images with very high reliability, achieving more than 95% accuracy. Because few channels are available, but spatial resolution is high, we use texture features in addition to raw radiometry and derived color channels as classification input. Region boundaries are obtained from a multi-scale hierarchical segmentation or from a registration of cadastral maps—the latter gives lower classification accuracies but better spatial precision. When using segmentation regions, the presented  $\text{MAP}^{1/N}$  method outperforms the per-pixel baseline methods also evaluated. We obtain slightly more accurate classifications than other studies, while, thanks to the use of textural features, being able to use higher spatial resolution images—and therefore having much improved spatial precision.

In addition, we have proposed several confidence measures that should indicate those classifications the classifier is unsure about; this would allow a trade-off between classification accuracy and coverage —leaving some areas unclassified, to be processed by a human operator, in exchange for higher accuracy in the classified regions. Evaluation shows that these confidence measures behave as expected for some cases but not all: further research is needed on the subject. Additional research into better-fitting probability models and into optimal texture features and color transformations is also desirable. Good results in these areas may make it possible to obtain similar accuracies on classifications with a richer set of land cover classes.

## References

1. Rosenfeld, A., Hummel, R., Zucker, S.: Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics* **6** (1976) 320–433
2. Wang, J.P.: Stochastic relaxation on partitions with connected components and its application to image segmentation. *IEEE Trans. on PAMI* **20** (1998) 619–636
3. Deng, H., Clausi, D.A.: Unsupervised image segmentation using a simple MRF model with a new implementation scheme. In: Proc. 17th ICPR, Cambridge, UK, IAPR (2004)
4. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on PAMI* **6** (1984) 721–741
5. Aplin, P., Atkinson, P.M.: Predicting missing field boundaries to increase per-field classification accuracy. *Photogr. Eng. and Remote Sensing* **70** (2004) 141–149
6. De Wit, A.J.W., Clevers, J.G.P.W.: Efficiency and accuracy of per-field classification for operational crop mapping. *International Journal of Remote Sensing* **25** (2004) 4091–4112
7. Trias-Sanz, R.: An edge-based method for registering a graph onto an image with application to cadastre registration. In: Proc. of ACIVS 2004, Brussels, Belgium (2004) 333–340
8. Trias-Sanz, R., Pierrot-Deseilligny, M.: A region-based method for graph to image registration with application to cadastre data. In: Proc. ICIP, Singapore, IEEE (2004)
9. Rosenblatt, M.: Remarks on some nonparametric estimates of a density functions. *Annals of Mathematical Statistics* **27** (1956) 642–669
10. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York, USA (1986)
11. Schwarz, G.: Estimating the dimension of a model. *Annals of Statistics* **6** (1978) 461–464
12. Cramér, H.: Mathematical Methods of Statistics. Princeton U. Press, Princeton, USA (1946)
13. Giffon, S.: Classification grossière par radiométrie et texture en zones agricoles. Master's thesis, Université Jean Monnet, Saint Étienne, France (2004) (In French).
14. Van de Wouwer, G., Scheunders, P., Livens, S., Van Dyck, D.: Wavelet correlation signatures for color texture characterization. *Pattern Recognition* **32** (1999) 443–451
15. Baillard, C.: Analyse d'images aériennes stéréo pour la restitution 3-D en milieu urbain. PhD thesis, École Nat. Sup. des Télécommunications, Paris, France (1997) (In French).
16. Guigues, L., Le Men, H., Cocquerez, J.P.: Scale-sets image analysis. In: Proc. ICIP, Barcelona, Spain, IEEE (2003)
17. Thomas, N., Hendrix, C., Congalton, R.G.: A comparison of urban mapping methods using high-resolution digital imagery. *Photogr. Eng. and Remote Sensing* **69** (2003) 963–972
18. Haapanen, R., Ek, A.R., Bauer, M.E., Finley, A.O.: Delineation of forest/nonforest land use classes using nearest neighbor methods. *Remote Sensing of Environment* **89** (2004) 265–271
19. Cablk, M.E., Minor, T.B.: Detecting and discriminating impervious cover with high-resolution IKONOS data using principal component analysis and morphological operators. *International Journal of Remote Sensing* **24** (2003) 4627–4645

# Invariance in Kernel Methods by Haar-Integration Kernels

B. Haasdonk<sup>1</sup>, A. Vossen<sup>2</sup>, and H. Burkhardt<sup>1</sup>

<sup>1</sup> Computer Science Department,

Albert-Ludwigs-University Freiburg, 79110 Freiburg, Germany

{haasdonk, burkhardt}@informatik.uni-freiburg.de

<sup>2</sup> Institute of Physics, Albert-Ludwigs-University Freiburg,  
79104 Freiburg, Germany

vossen@physik.uni-freiburg.de

**Abstract.** We address the problem of incorporating transformation invariance in kernels for pattern analysis with kernel methods. We introduce a new class of kernels by so called Haar-integration over transformations. This results in kernel functions, which are positive definite, have adjustable invariance, can capture simultaneously various continuous or discrete transformations and are applicable in various kernel methods. We demonstrate these properties on toy examples and experimentally investigate the real-world applicability on an image recognition task with support vector machines. For certain transformations remarkable complexity reduction is demonstrated. The kernels hereby achieve state-of-the-art results, while omitting drawbacks of existing methods.

## 1 Introduction

Many pattern analysis tasks are based on learning from examples, i.e. sets of observations are given, which are to be processed in some optimal way. Such tasks can consist of classification, regression, clustering, outlier-detection, feature-extraction etc. A powerful battery of algorithms for such tasks is given by so called ..., which attract increasing attention due to their generality, adaptability, theoretic foundation, geometric interpretability and excellent experimental performance, cf. [1]. The most famous representative is the support vector machine (SVM). It is meanwhile widely accepted, that additional problem specific prior knowledge is crucial for improving the generalization ability of such learning systems [2]. In particular, prior-knowledge about pattern transformations is often available. A simple example is that geometric transformations like rotations or translations of an image frequently do not change the inherent meaning of the displayed object. The insight, that the crucial ingredient for powerful analysis methods is the choice of a kernel function, led to various efforts of problem-specific design of kernel functions.

In this paper we introduce a new class of kernel-functions, so called Haar-integration kernels, which incorporate such transformation knowledge. They

are based on a successful technique for extracting invariant features, the so called Haar-integration procedure. Extension of this technique to kernel functions seems to be the first proposal, which omits various drawbacks of existing approaches. In particular the advantages are positive definiteness, steerable transformation extent, applicability in case of both continuous and discrete transformations, applicability to different kinds of base-kernel-functions and arbitrary kernel-methods.

The structure of the paper is as follows: In the next section we recall the required notions concerning kernel methods. Section 3 introduces the new proposal and derives some theoretical properties. We continue with comments on the relation to existing approaches. The subsequent Section 5 presents simple visualizations of the kernels in 2D. As sample kernel method we choose the SVM, for which we present some illustrative toy-classification results. In Section 6 real world applicability on an image recognition task is demonstrated consisting of a well known benchmark dataset for optical character recognition, the USPS digit dataset. Additionally, the kernels allow a remarkable speedup as is demonstrated in Section 7 before we finish with some concluding remarks.

## 2 Kernel Methods

In this section we introduce the required notions and notations, which are used in the sequel concerning kernel methods, cf. [1] for details on the notions and concepts. In general, a kernel method is a nonlinear data analysis method for patterns from some set  $x \in \mathcal{X}$ , which is obtained by application of the . . . . . on a given linear method: Assume some linear analysis method operating on vectors  $\mathbf{x}$  from some Hilbert space  $\mathcal{H}$ , which only accesses patterns  $\mathbf{x}$  in terms of inner products  $\langle \mathbf{x}, \mathbf{x}' \rangle$ . Examples of such methods are PCA, linear classifiers like the Perceptron, etc. If we assume some nonlinear mapping  $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ , the linear method can be applied on the images  $\Phi(x)$  as long as the inner products  $\langle \Phi(x), \Phi(x') \rangle$  are available. This results in a nonlinear analysis method on the original space  $\mathcal{X}$ . The . . . . . now consists in replacing these inner products by a kernel function  $k(x, x') := \langle \Phi(x), \Phi(x') \rangle$ : As soon as the kernel function  $k$  is known, the Hilbert space  $\mathcal{H}$  and the particular embedding  $\Phi$  are no longer required. For suitable choices of kernel function  $k$ , one obtains methods, which are very expressive due to the nonlinearity, but cheap to compute, as explicit embeddings are omitted. If for some function  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  a Hilbert space  $\mathcal{H}$  and a mapping  $\Phi : \mathcal{X} \rightarrow \mathcal{H}$  can be found such that  $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$  holds, then  $k$  is called , . . . . . A larger class of kernels which is useful for various kernel methods is the slightly weaker notion of . . . . . positive definite (cpd) kernels. Some standard kernels, which are used in practice for vectorial data  $\mathbf{x}$  are the linear, polynomial, Gaussian and negative distance kernel, where the latter is cpd, the remaining ones are pd:

$$\begin{aligned} k^{\text{lin}}(\mathbf{x}, \mathbf{x}') &:= \langle \mathbf{x}, \mathbf{x}' \rangle & k^{\text{nd}}(\mathbf{x}, \mathbf{x}') &:= -\|\mathbf{x} - \mathbf{x}'\|^\beta, \beta \in [0, 2] \\ k^{\text{pol}}(\mathbf{x}, \mathbf{x}') &:= (1 + \gamma \langle \mathbf{x}, \mathbf{x}' \rangle)^p & k^{\text{rbf}}(\mathbf{x}, \mathbf{x}') &:= e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|^2}, p \in \mathbb{N}, \gamma \in \mathbb{R}^+ \end{aligned} \quad (1)$$

As a sample kernel method, we will refer to the SVM for classification. In the case of two-class classification, this method requires a kernel function  $k$ , training patterns and class labels  $(x_i, y_i) \in \mathcal{X} \times \{\pm 1\}, i = 1, \dots, n$  and produces a classification function which assigns the class  $f(x) = \text{sgn}(\sum_i \alpha_i y_i k(x, x_i) + b)$  to a new pattern  $x$ . Here the  $\alpha_i, b$  are the parameters, which are determined during training. Multiclass problems can be solved by reducing a problem to a collection of two-class problems in various ways. We refrain from further details.

### 3 Haar-Integration Kernels

In the field of pattern recognition, particular interest is posed on invariant feature extraction, i.e. finding some function  $I(x)$ , which satisfies  $I(x) \sim I(gx)$  or even with equality for certain transformations  $g$  of the original pattern  $x$ . One method for constructing such invariant features is the so called Haar-integration technique [3]. In this approach, invariant representations of patterns are generated by integration over the known transformation group. These features have been successfully applied on various real world applications ranging from images to volume data [4, 5, 6] and have been extended to be applicable on subsets of groups [7]. A similar technique can be applied to generate invariant kernels, which leads to the definition of the Haar-integration kernels.

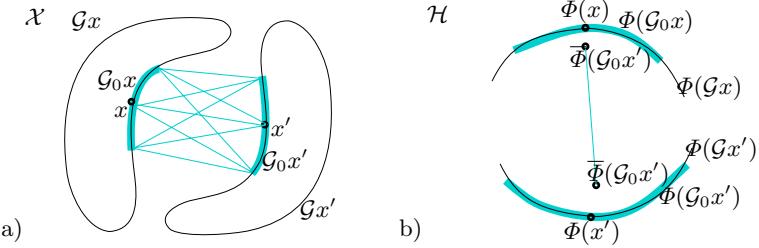
**Definition 1 (Haar-Integration Kernel).**

$$k(x, x') = \int_{\mathcal{G}_0} \int_{\mathcal{G}_0} k_0(gx, g'x') dg dg' \quad (2)$$

Haar-integration kernel (HI-kernel) of  $k_0$  with respect to  $\mathcal{G}_0$

The requirement of the integrability of  $k_0$  is practically mostly satisfied, e.g. after finite discretization of  $\mathcal{G}_0$ . The motivation of the integral (2) is demonstrated in Fig. 1 in two ways: a) in the original pattern space and b) in the  $k_0$ -induced feature space. For simplicity we assume the Haar-measure to be normalized to  $dg(\mathcal{G}_0) = 1$ . In the left figure, two patterns  $x, x'$  are illustrated in the pattern space  $\mathcal{X}$  including their orbits  $\mathcal{G}x, \mathcal{G}x'$ . The goal is to find a kernel function, which satisfies  $k(x, x') \sim k(gx, g'x')$  for small transformations  $g$  of  $x$ . If we define  $\mathcal{G}_0$  as illustrated, the Haar-integration kernel generated by  $k_0$  is the average over all pairwise combinations of  $\mathcal{G}_0x$  and  $\mathcal{G}_0x'$ . If  $\mathcal{G}_0$  is large enough, the integration ranges  $\mathcal{G}_0x$  and  $\mathcal{G}_0gx$  have a high overlap, which makes the resulting integrals arbitrarily similar. In the right sketch b), the interpretation of the kernels in feature-space is given: Instead of averaging over  $k_0(x, x')$ , the integration kernel is the inner product of the average of the sets  $\Phi(\mathcal{G}_0x'), \Phi(\mathcal{G}_0x)$ , respectively, due to

$$\left\langle \int_{\mathcal{G}_0} \Phi(gx) dg, \int_{\mathcal{G}_0} \Phi(g'x') dg' \right\rangle = \int_{\mathcal{G}_0} \int_{\mathcal{G}_0} \langle \Phi(gx), \Phi(g'x') \rangle dg dg' = k(x, x'). \quad (3)$$



**Fig. 1.** Geometric interpretation of Haar-integration kernels. a) original pattern space  $\mathcal{X}$ , b) kernel-induced feature space  $\Phi(\mathcal{X}) \subset \mathcal{H}$

Again, small transformations of  $x$  to  $gx$  results in similar sets of transformed patterns in feature space, similar averages and similar kernel values.

Some theoretical properties of these kernels are quite convenient:

**Proposition 1 (Basic Properties of Haar-Integration Kernels).**

$$\begin{aligned} & \mathcal{G}_0 = \mathcal{G} \quad k(x, x') = k(gx, g'x') \quad x, x' \in \mathcal{X}, g, g' \in \mathcal{G} \\ & k_0 \quad , \quad k \quad , \quad \end{aligned}$$

(Sketch) (i) For characteristic functions  $k_0(gx, g'x') = \chi_A(g) \cdot \chi_{A'}(g')$  with measurable  $A, A' \subset \mathcal{G}$ , we obtain with linearity of the integral and the invariance of the Haar-measure  $dg$  that  $k(hx, h'x') = dg(A) \cdot dg'(A')$ . This is independent of  $h, h'$ , thus invariant. The invariance in case of these characteristic functions transfers similarly to other measurable sets  $A \subset \mathcal{G} \times \mathcal{G}$ , linear combinations of such characteristic functions and the limit operations involved in the Lebesgue-integral definition.

(ii) The symmetry of  $k$  is obvious. If  $k_0$  is pd then  $\bar{\Phi}(x) := \int_{\mathcal{G}_0} \Phi(gx) dg$  is a mapping from  $\mathcal{X}$  to  $\mathcal{H}$  with  $\langle \bar{\Phi}(x), \bar{\Phi}(x') \rangle = k(x, x')$  according to (3). So in particular  $k$  is pd. If  $k_0$  is cpd., the kernel  $\tilde{k}_0$  following [1–Prop. 2.22] is pd (and cpd), so is the corresponding HI-kernel  $\tilde{k}$ . This function contains the HI-kernel  $k$  of  $k_0$  plus some functions depending on solely one of the arguments  $x, x'$ . Such functions maintain cpd-ness, so  $k$  is cpd.  $\square$

The HI-kernels are conceptionally an elegant seamless connection between non-invariant and invariant data-analysis: The size of  $\mathcal{G}_0$  can be adjusted from the non-invariant case  $\mathcal{G}_0 = \{\text{id}\}$ , which recovers the base-kernel, to the fully invariant case  $\mathcal{G}_0 = \mathcal{G}$ . This will be further demonstrated in Sec. 5.

## 4 Relation to Existing Approaches

We want to discuss some relations to existing feature-extraction and invariant SVM methods. Eqn. (3) indicates the relation of the HI-kernels to the [7]. The HI-kernels are inner products of corresponding

Haar-integral features in the Hilbert space  $\mathcal{H}$ . Some kernels are known to induce very high or even infinite dimensional spaces, e.g.  $k^{\text{rbf}}$ . So in these cases, the HI-kernels represent Haar-integral invariants of infinite dimension. Clearly this is a computational advantage, as these could not be computed explicitly in feature space. On the other hand, all inner products between Haar-invariant feature representations are captured by the HI-kernel approach, by a suitable base-kernel  $k_0$ . So these kernels are conceptionally richer.

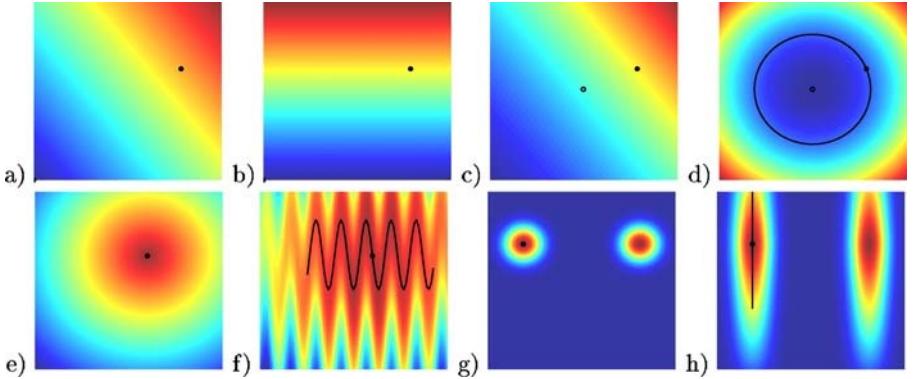
Invariance in kernel methods has been mainly proposed resulting in non-positive definite kernels as the ... [2], ... [8] or ... [9]. In contrast to these methods, the proposed kernels have the important advantage of being positive definite. They can be applied to non-differentiable, discrete transformations and to general kernels, not only to distance-based ones or the Gaussian. In contrast to [10], which theoretically constructs invariant kernels by solving partial differential equations, we obtain practically applicable kernels.

There are further methods of specially incorporating invariance in SVM. The method of ... or the nonlinear extension ... [11, 12] are theoretical nice constructions of enforcing the invariant directions into the SVM optimization problem. However they suffer from the high computational complexity. The most widely accepted method for invariances in SVM is the ... method [13]. It consists of a two step training stage. In a first ordinary SVM training step, the support vectors are determined. This set is multiply enlarged by various small transformations of each support vector. The second training stage on this set of virtual support vectors yields an invariant SVM. The problem of this approach is the enlarged memory and time complexity during training.

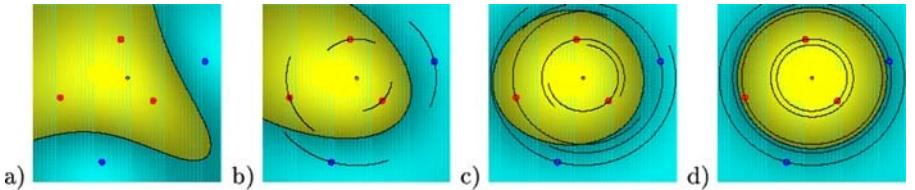
## 5 Toy-Experiments

In this section we illustrate the kernels and their application in a kernel-method on simple toy-examples. For this, we choose the patterns  $\mathbf{x}$  from the Euclidean plane  $\mathcal{X} = \mathbb{R}^2$ , and define simple transformations on these points. The transformations are translations along fixed directions, rotation around a fixed point, shift along a sinus-shaped curve and reflection along a vertical axis. By these transformations we cover linear, nonlinear and extremely nonlinear operations. Additionally, they represent both continuous and discrete transformations.

We start with illustration of the kernel functions in Fig. 2. For this, we fix one point  $\mathbf{x}'$  (black dot) and plot the kernel value  $k(\mathbf{x}, \mathbf{x}')$ , while  $\mathbf{x}$  varies over the unit-square. The kernel values are color-coded. We start with the demonstration of the invariant linear and polynomial kernel the upper row. Subplot a) demonstrates the non-invariant kernel  $k^{\text{lin}}$ , which is made invariant with respect to reflections along a perpendicular axis in b). Subplot c) illustrates the kernel  $k^{\text{pol}}$  (of degree 2), which is nicely made invariant with respect to complete rotations d). Here and in the subsequent plots, we highlight the integration range  $\mathcal{G}_0\mathbf{x}'$  by solid lines passing through the point  $\mathbf{x}'$ . So these inner-product



**Fig. 2.** Demonstration of invariant kernels. a), b) non-invariant/reflection invariant  $k^{\text{lin}}$ , c), d) non-invariant/rotational invariant  $k^{\text{pol}}$ , e), f)  $k^{\text{nd}}$  with highly nonlinear sinus invariance, g), h)  $k^{\text{rbf}}$  with simultaneous reflection and translation invariance



**Fig. 3.** Demonstration of invariant kernels in SVM classification. a) non-invariant  $k^{\text{rbf}}$ , b), c) partial rotational invariance, d) complete rotational invariance

kernels work nicely for these global transformation groups. However it turned out, that they have problems with partial invariances, e.g. reducing the range of rotations in d). The reason is, that the required nonlinearity increases: Reducing the circle of rotated patterns to a semi-circle, the isolines of the desired invariant kernel would need to be half-moon-shaped around the semi-circle. This cannot be expressed by a HI-kernel of a 2nd degree polynomial, as averaging does not increase the polynomial degree. So ideally, base-kernels are required, which can express arbitrary complex boundaries. Such kernels are given by  $k^{\text{nd}}$  or  $k^{\text{rbf}}$ . These kernels proved to work in all preceding cases and cases which we present in the lower row. Plot e) and f) illustrate the negative distance kernel ( $\beta = 1$ ) for highly nonlinear transformations consisting of shifts along sinus-curves, where the size of  $\mathcal{G}_0$  can be smoothly increased between the plots. Similarly, the Gaussian kernel is made invariant with respect to the combination of reflection and increasing  $y$ -translations in g) and h). In both cases the transformations are nicely captured covering linear, highly nonlinear, discrete transformations and combinations thereof.

We continue with demonstrating the increased separability when applied in classification problems by SVM. Given 5 (red and blue) points in Fig. 3, the effect of increasing rotational invariance (with respect to the black circle) is illustrated.

In the leftmost non-invariant case, the classification result of a standard  $k^{\text{rbf}}$  is illustrated, which correctly classifies the points, but indeed captures none of the rotational invariance. By increasing the rotational integration range, the solution is a completely invariant SVM solution in d). Similar results can be obtained for the negative distance kernel.

## 6 Real-World-Experiments

As a real world application of the kernels, we perform image classification experiments on an optical character recognition problem. In this setting partial invariances are particularly useful as e.g. only small rotations are allowed, whereas large rotations will confuse W and M, 6 and 9 or N and Z. Only small x- and y-translations are reasonable, if the patterns are already roughly centered. We restrict the real world experiments to these rigid transformations and the  $k^{\text{rbf}}$  kernel, as this turned out to capture nicely partial invariances in the previous toy-experiments.

For enabling comparisons with existing methods, we chose the well known benchmark dataset of USPS-digits. The corpus consists of 7291 training and 2007 test images of  $16 \times 16$  greyvalue images of handwritten digits. Figure 4 depicts a) some simple and b) difficult to classify example images. A list of reference results can be found in [1]. The relevant results for our purpose are the 2.5% test error rate, which is reported for humans [14] and indicates the difficulty of the dataset. A standard polynomial SVM is reported to reach 4.0% error rate [15], which is improved by the VSV-method involving translations obtaining 3.2% test error [13]. This is the most comparable result in literature. Some better results have been reported, but those apply more sophisticated deformation models, more training data etc. So they involve different kinds of prior knowledge than only rigid transformations of the images.



**Fig. 4.** Examples of USPS digits. a) easy, b) difficult to classify examples

As classifier for the dataset, we use a one-versus-rest multiclass SVM applying the HI-kernel (HI-SVM). The SVM package LIBSVM [16] was taken as a basis for the implementation. In the first set of experiments we focus on recognition accuracy. After setting the integration ranges, the remaining SVM-parameter pair is  $(C, \gamma)$ . For this we chose  $10^2$  combinations of 10 values for each parameter. One SVM model was trained for each parameter set and the results of the best models are reported in the following. Note that this is a kind of . . . . . which produces optimistically biased test-error rates compared to the true generalization performance. But this is a general phenomenon for many public datasets including USPS.

**Table 1.** USPS recognition results with HI-kernels. rotation integration (left), x-y-translation (right)

$\phi$ -range [rad]	$k^{\text{rbf}}$	test error [%]	x-y-range [pixels]	$k^{\text{rbf}}$	test error [%]
0		4.5	0		4.5
$\pm 0.04\pi$		4.1	$\pm 1$		3.7
$\pm 0.08\pi$		4.2	$\pm 2$		3.2
$\pm 0.12\pi$		3.9	$\pm 3$		3.3
$\pm 0.16\pi$		4.2	$\pm 4$		3.2

The left part of Tab. 1 lists the results which are obtained by increasing the rotation integration. Numerical integration is performed involving  $3 \times 3$  sample points for each  $\mathcal{G}_0$  integral. The table indicates that the HI-kernel clearly capture the wanted invariances, as the results improve compared to the non-invariant SVM, which has 4.5% test error. Still the results are far from the existing VSV-result, which was based on translations. Therefore, a second experiment sequence was performed by regarding translations only. Initial experiments yielded that with increasing the number of integration evaluation points to  $9 \times 9$  very good results are obtained. We increase the translation range from 0 to  $\pm 4$  pixels. The results of this are depicted in the left part of Tab. 1. The results clearly improve and equal the state-of-the art result of the VSV approach.

## 7 Acceleration

The integration kernels allow various ways of time complexity reduction. Suitable caching strategies can accelerate the computation procedure, e.g. caching the transformed patterns throughout the computation of the kernel-matrix, etc. A more fundamental complexity reduction can be performed similar as in the case of jittering kernels, if the transformations are commutative and compatible with the base-kernel in the sense that  $k_0(gx, g'x') = k_0(x, g^{-1}g'x)$ . An example of such kernels are all kernels based on distance or inner products of vectors, if the transformations are rotations, translations or even permutations of the vector entries. In this case, the complexity of a single kernel evaluation can be reduced remarkably from  $\mathcal{O}(s^{2l})$  to  $\mathcal{O}(s^l)$ , where  $s$  is the number of integration steps along each of the  $l$  transformation directions. This integral reduction can be obtained by halving the number of integrals:

$$\int_{\mathcal{G}_0} \int_{\mathcal{G}_0} k_0(gx, g'x') dg dg' = \int_{\mathcal{G}_0} \int_{\mathcal{G}_0} k_0(x, g^{-1}g'x') dg dg' = \int_{\mathcal{G}_0^{-1}\mathcal{G}_0} k_0(x, \bar{g}x') d\bar{g}. \quad (4)$$

where  $\mathcal{G}_0^{-1}$  denotes the set of all inverted  $\mathcal{G}_0$  elements and  $d\bar{g}$  denotes a suitable resulting measure. If  $\mathcal{G}_0$  is chosen reasonable, the resulting set  $\mathcal{G}_0^{-1}\mathcal{G}_0$  will be much smaller than the original  $\mathcal{G}_0 \times \mathcal{G}_0$ . We continue with a second method of

**Table 2.** Complexity comparison between HI-SVM and VSV-method with  $3 \times 3$  2D-translations

Method	test-error [%]	train-time [s]	test-time [s]	average #SV
HI-SVM	3.6	1771	479	412
HI-SVM, IR	3.6	810	176	412
HI-SVM, SV	3.6	113 + 130 + 297	466	410
HI-SVM, SV + IR	3.6	113 + 130 + 91	172	410
VSV-SVM	3.5	113 + 864 + 1925	177	4240

acceleration in the special case of SVM-classification. The support-vectors of the non-invariant SVM and the HI-SVM turn out to have a high overlap. This suggests to apply the idea of the VSV-SVM on HI-SVM: Perform a two-step training stage by initial ordinary  $k^{\text{rbf}}$  SVM training, then selecting the support vectors and performing an HI-SVM training on this SV-set.

We performed tests of all combinations of these two acceleration methods denoted as IR (integral reduction) and SV (support vector extraction) in Tab. 2, and investigated the resulting time and model complexities. For comparison, we added the VSV-model complexities. In order not to bias the results towards one of the methods, we fixed  $C = 100, \gamma = 0.01$  and the x-y-translation to  $\pm 2$  pixels, with (implicit)  $3 \times 3$  transformed patterns per sample. The recognition results are almost identical, but not very expressive as they are suboptimal due to the missing  $C, \gamma$  optimization. The experiments indicate, that the integral reduction (IR) indeed reduces the training and test-time remarkably, while the recognition accuracy remains unchanged as expected. By applying the SV-extraction step in an initial training stage, the error rate does not increase, but the training time (first training + (V)SV-extraction + second training) is again largely reduced. The testing time does only improve marginally by SV-extraction as the (rounded) number of SV in the final models is not significantly reduced. The comparison between the accelerated Haar-integral methods and the VSV yields, that the model size in terms of average number of SVs is clearly smaller applying our kernels, as the kernels themselves represent relevant information of the model by the invariance, so storage complexity is largely reduced. The HI-kernels are more expensive to evaluate than standard kernels as the VSV method uses, so testing is more expensive than the VSV-method. Still, by the acceleration methods the testing time can compete with the VSV-method. In the training time, the high value of 864 sec for the SV-extraction is to be taken with care and might be decreased by optimizing the explicit construction of the VSV set. Despite this possible optimization, the accelerated integration kernels are during training still clearly faster than the VSV-SVM. This is due to the fact, that the VSV method suffers from the 9-times enlarged SV-training set. Although both the VSV and the fast HI-kernels are expected to slowdown quadratically, the VSV seems to be more affected by this.

## 8 Conclusions

We introduced a class of invariant kernels called Haar-integration kernels. These kernels seem to be the first invariant kernels, which alleviate the problem of missing positive definiteness, as observed in other approaches. Furthermore, they are not restricted to continuous or differentiable transformations but allow explicit discrete or continuous transformations. The degree of invariance can be smoothly adjusted by the size of the integration interval. Experimental application in a particular kernel method, namely SVM, allowed a comparison to the state-of-the art method of VSV. Test on the real world USPS dataset demonstrates that state-of-the art recognition results can be obtained with these kernels. Complexity comparisons demonstrated large improvements in training and testing time by the techniques of integral reduction and training on the SVs of an ordinary  $k^{rbf}$ -SVM. The expensive HI-kernel evaluation is ameliorated by the reduced model size during testing, such that both training and testing times can compete with or outperform the VSV approach.

So in SVM learning the kernels seem a good alternative to VSV, if the testing time is not too crucial or small model size is required. In other kernel methods where the complexity grows quadratically with the number of training examples, the generation and storing of virtual examples might be prohibitive. In these situations, the proposed kernels can be a welcome approach. Perspectives are to apply the integration kernels in further kernel-methods and on further datasets. Interesting options are, to apply the technique to non-group transformations, in particular non-reversible transformations as long as "forward" integrations are possible.

## References

1. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
2. DeCoste, D., Schölkopf, B.: Training invariant support vector machines. Machine Learning **46** (2002) 161–190
3. Schulz-Mirbach, H.: Constructing invariant features by averaging techniques. In: Proc. of the 12th ICPR. Volume 2., IEEE Computer Society (1994) 387–390
4. Schael, M.: Texture defect detection using invariant textural features. In: Proc. of the 23rd DAGM - Symposium Mustererkennung, Springer Verlag (2001) 17–24
5. Sigdelkow, S.: Feature-Histograms for Content-Based Image Retrieval. PhD thesis, Albert-Ludwigs-Universität Freiburg (2002)
6. Ronneberger, O., Schultz, E., Burkhardt, H.: Automated pollen recognition using 3d volume images from fluorescence microscopy. Aerobiologia **18** (2002) 107–115
7. Haasdonk, B., Halawani, A., Burkhardt, H.: Adjustable invariant features by partial Haar-integration. In: Proc. of the 17th ICPR. Volume 2. (2004) 769–774
8. Haasdonk, B., Keysers, D.: Tangent distance kernels for support vector machines. In: Proc. of the 16th ICPR. Volume 2. (2002) 864–868
9. Pozdnoukhov, A., Bengio, S.: Tangent vector kernels for invariant image classification with SVMs. In: Proc. of the 17th ICPR. (2004)
10. Burges, C.J.C.: Geometry and invariance in kernel based methods. In: Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 89–116

11. Chapelle, O., Schölkopf, B.: Incorporating invariances in nonlinear support vector machines. In: NIPS 14, MIT-Press (2002) 609–616
12. Schölkopf, B., Simard, P., Smola, A., Vapnik, V.: Prior knowledge in support vector kernels. In: NIPS 10, MIT Press (1998) 640–646
13. Schölkopf, B., Burges, C., Vapnik, V.: Incorporating invariances in support vector learning machines. In: ICANN'96, LNCS, 1112, Springer (1996) 47–52
14. Simard, P., LeCun, Y., Denker, J.: Efficient pattern recognition using a new transformation distance. In: NIPS 5, Morgan Kaufmann (1993) 50–58
15. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proc. of the 1st KDD, AAAI Press (1995) 252–257
16. Ronneberger, O., Pigorsch, F.: LIBSVMTL: a support vector machine template library (2004) <http://lmb.informatik.uni-freiburg.de/lmbsoft/libsvmmtl/>.

# Exemplar Based Recognition of Visual Shapes

Søren I. Olsen

Department of Computer Science,  
University of Copenhagen, Denmark

**Abstract.** This paper presents an approach of visual shape recognition based on exemplars of attributed keypoints. Training is performed by storing exemplars of keypoints detected in labeled training images. Recognition is made by keypoint matching and voting according to the labels for the matched keypoints. The matching is insensitive to rotations, limited scalings and small deformations. The recognition is robust to noise, background clutter and partial occlusion. Recognition is possible from few training images and improve with the number of training images.

## 1 Introduction

Several recent successful approaches to shape recognition [3, 5, 6, 8, 9] are based on attributed image keypoints. By choosing the descriptors carefully, such approaches has shown robust to rotations, scalings, illumination changes, 3D camera viewpoint including minor deformations, and - probably most important - background clutter and partial occlusion. The present approach follows this line of research. In the recognition phase the focus is not on instance detection, but on semantic contents report. The semantic contents of a training image is supplied as a list of names (labels), e.g. 'house', 'chair' etc. The set of names form the vocabulary by which test images can be recognized. Given a new unlabeled images the system reports the label(s) with strongest support from the matched keypoints. Also an image of the recognized structures is produced.

## 2 Previous Work

Probably the most influential early work using keypoints for recognition is the paper of Schmid and Mohr [9]. Here keypoints are detected at multiple scale levels using a Harris corner detector, and a vector of differential invariants is used as descriptor. Recognition is done using multidimensional indexing and voting and by applying a model of the shape configuration, i.e. the spatial relationship between the keypoints defining a model. In later work [10, 6] the use of different interest point operators has been evaluated, and the Harris detector has been combined with other approaches to result in a scale and affine invariant detector.

Another influential work is the papers by Lowe [3, 4, 5]. Here the keypoints are detected in scale-space as the local extremes in the convolution of the image

with the difference of Gaussians. The descriptor is chosen by sampling the image gradient orientation in a rectangular grid centered at the keypoint and aligned with the dominating local gradient orientation. By using a variant of the k-d-tree for indexing, input keypoint descriptors are matched to their most similar neighbor in the database of trained descriptors. Then sets of 3 matched keypoints defining the pose are grouped using the Hough transform [5]. Next, an accurate pose of the recognized object is fitted by an iterated least squares affine fit with outlier removal. Decision to reject or accept the model hypothesis is finally made based on a probabilistic model [4].

The present work may be seen as a refinement of [8]. In this work keypoints are detected in scale-space using a model of end-stopped cells [2] and using centers of circular contours. The former type of keypoint identify corners (2 legs), junctions (3 legs) and more complicated structures with 4 legs. The directions of the legs is well suited for indexing. In the present approach the keypoint types are unchanged, but the detection method has been improved. Compared with the keypoint types used in [5, 6] fewer keypoints are in general detected. These are larger scale features more likely to mark positions with high semantic contents.

In [8] a 2-dimensional histogram of local edge point gradient orientation located within an angular sector relative to the keypoint is used as descriptor. Comparison between an input and a database keypoint is made by a modified  $\chi^2$ -test. Due to quantization problems this descriptor often does not perform well. To achieve a recognition invariant to rotations, scalings act. one method is to choose descriptors that are invariant to such transforms [9, 6]. This approach is reasonable only if the transformations can model the expected deformations well. In the present work this is not appropriate, because the chosen descriptor is not very local. To achieve rotational invariance the descriptor measurements may be made in a coordinate system aligned with the dominating local image gradient orientation [5]. For the keypoints chosen in the present system there will be either several or no such orientations. In [5] the problem of multiple dominating orientations is solved by storing as many descriptors as there are such orientations. This reduces the time for single descriptor comparisons but increases the size of the database significantly. We take a third approach, using a descriptor that is not invariant to the transforms mentioned. Instead rotational invariance and insensitivity to minor deformations is left to the matching process. This choices will lower the size of the database at the expense of a larger computational complexity during the matching.

The present work is focused on reporting the semantic contents of an image in terms of one or several labels introduced during training. This is different from detecting the existence and pose of a specific object in an image [5, 9]. Loosely speaking, the method is aimed at object shape categorization rather than image-to-image matching. Thus a requirement of positional consistency between a group of neighboring query and database keypoints is not necessary. In the present system there is no spatial relations between keypoints defining a shape. The lost discriminative power is regained by choosing a descriptor with a larger spatial support. Each query keypoint may be matched to multiple database

keypoints. The classification is then made using a simple voting mechanism where each match votes on the database keypoint label with a strength determined by the similarity between the two descriptors.

### 3 Keypoint Detection

First the essential aspects of the visual shape such as object outline contours and texture contours are extracted by edge detection using Gaussian derivative convolution and spatial non-maximum suppression of the gradient magnitude. The edge points are then used to detect the two types of keypoints: Circular structures and junctions (including corners). These keypoints mark image positions where the semantic content of the local image shape is locally rich. The core of the detection method is described in [8]. Experiments however have shown that for junctions neither the estimated localization nor the number and directions of the legs defining the junction are sufficiently stable or accurate.

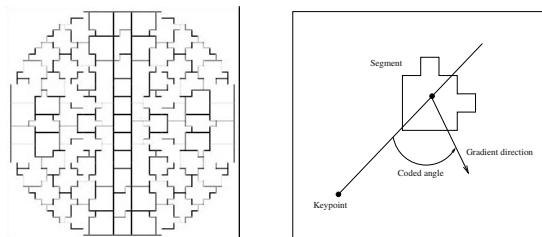
In the present work each detected junction is validated in a refinement step. For each detected junction the edge points in the local neighborhood are grouped according to their local angular positioning and gradient orientation. Then a straight line is fitted to the points in each group. If few points contribute to the fit, if the residual of the fit is bad, or if the distance from the junction point to the fitted line is large, then the group is discarded. If less than 2 groups remain the junction is discarded. Otherwise the fitted lines are intersected using least squares. If the residual of the fit is large or if the intersection point is far from the initial position of the junction this is discarded. Otherwise the intersection point becomes the new position of the junction. Thus, the remaining keypoints are well localized and defined by locally straight edges with a well defined orientation.

To achieve an optimal detection of keypoints previous methods have used a multi-resolution (scale-space) approach where keypoints initially are detected at a set of scale levels, then linked through scale-space, and accepted only if some strength measure is extreme. In [6] a single optimal scale for each keypoint is found. In [5] several (extremal) keypoints may be selected. Experiments have shown that both corners and junctions may be inconsistently detected over scale, i.e. the number of legs and their orientation may differ. In the present approach a scale-space approach is used - not to ensure optimal detection - but to enable recognition of shapes scaled in size and to prune unstable detections. First keypoints are detected in scale-space, using a sampling of  $\sqrt[4]{2}$  corresponding to 3 samples per octave, and grouped in scale-space according to their type and positioning. For each group the dominating directions of the legs are found by histogram analysis. Then a number of the registered representations that are consistent with respect to the number and directions of the keypoint legs are selected by sampling. The sampling is made such that the scale distance between two selected items is at least 2 scale levels. Finally, all isolated non-selected keypoints (with no other selected keypoints in their neighborhood spatially as well as w.r.t. scale) is selected as well. This guarantees that only stable or unique representatives are chosen. The sampling is chosen as a compromise between a

small amount of redundancy among the representatives and a good coverage of different neighborhood sizes coded by the descriptors. The net result is a reduction in the number of selected keypoints (often by 50 %) and that most of the selected ones are supported by similar detections at other scale levels. Especially for junctions, most spurious leg detections are removed in this process.

## 4 Keypoint Descriptors

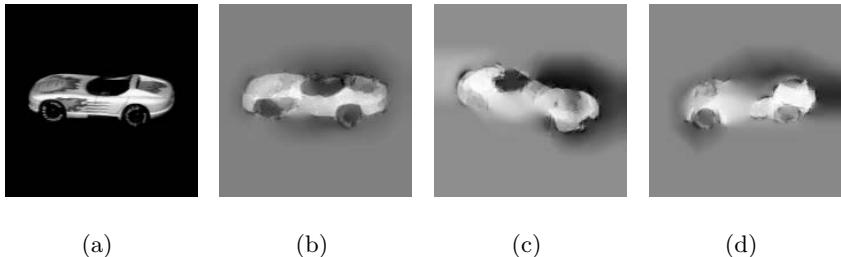
In the present work the choice of descriptors to a large extend follows the approach of local gradient sampling proposed by Lowe [3, 5]. However, both the sampling points, the quantization, and the descriptor matching is different. The exemplar approach, relying solely on the statistics of isolated image patch matches, requires that the keypoint descriptor is chosen to code the shape in a sufficiently large image patch in order to possess sufficient discriminative power. However it also must not be too selective. A reasonable trade off between discriminative power and generalization is experimentally found to correspond to a support radius between 10 and 20 pixels. In [5] a rectangular sampling grid is used. This grid is aligned with the dominating direction of gradient orientation. If several peaks in the histogram of gradient orientation are present, several keypoints are generated. This will double/triple the size of the database for corners/junctions. The advantage of the redundancy is a simpler matching of descriptors to the database. The disadvantage is that more descriptors should be matched to a larger base. We choose to represent each keypoint only once and to pay the price of having to “rotate” the descriptor vector for a each descriptor comparison. For this choice a sampling in a rectangular grid is inappropriate. We choose to sample the gradient information in 5 rings with radii about  $3i$ ,  $i = 1..5$ , and with a sampling distance of about 3 pixels in each ring, corresponding to 6, 12, 18, 24 and 30 samples (90 in total). The segmenting of the disc is made using the k-means algorithm. Each pixel within a distance of about 18 pixels from the keypoint is classified to the nearest segment sample point and the gradient with largest magnitude within each segment is chosen (see Figure 1). We then code the gradient magnitude and the angle between the gradient orientation and segment direction. Each of the two segment component are quantized to 4 bits



**Fig. 1.** Sampling regions (left), and coded angle for a segment (right)

each. Thus the description vector has the size of 90 bytes. The coding of the individual sample elements is invariant to rotations, but their position in the descriptor vector is not.

The motivation for inclusion of the (quantized) gradient magnitude in the descriptor is that this information allows a (more or less pleasing) reconstruction of a coded image. Also, a “mental image” of the recognized structures in a query image can be made based on the stored data, not the query data. As an example Figure 2 shows a query image, an image reconstructed from the coded data, and two reconstructions based on matched database descriptors. The database was constructed using 89 objects from the COIL-100 database [7], each object seen from 8 different viewing angles (45 degree separation in depth). The two query images differed from the nearest training image by a rotation of 10 degree (in depth).



**Fig. 2.** Image (a) and reconstruction (b) based on the 63 detected keypoints in the image. Two reconstructions (c) and (d) based on 22 and 14 keypoints

The reconstruction is made coarse to fine. The solution at a coarser level is used as initial value in the reconstruction at the next finer level. At each level the reconstruction is made in three steps. First a map of gradient magnitudes is constructed from the database descriptors at the positions defined by the matching input keypoints. Only matches to the winning label is used. Next, this map is anisotropically smoothed and the ridges detected. Ideally, these correspond to the recognized shape contours. For each point on the detected ridge the coded gradient information defines two equations in the partial derivatives of the reconstructed surface. Using standard regularization, a simple iterative updating procedure is finally applied. For simplicity a fixed number of iterations is used. Since no absolute intensity values are known the reconstruction can be made up to a additive constant only. In areas with few keypoints the reconstruction will be poor. Areas with no recognized keypoints will be reconstructed by a constant initial value of middle-gray. Thus the reconstructed image probably will be of low quality, but nevertheless show the recognized structures.

## 5 Shape Recognition

The database of stored keypoints is organized in four groups containing circular structures, corners, junctions, and structures with more than 3 legs. Corners are indexed by the quantized angle between the two legs. Junctions are accessed using a 2-dimensional array indexed by a similar quantization of the angles between the first and the other two legs. The stored keypoint descriptions for circular structures and structures with more than 3 legs are relatively few and compared to the query keypoints through an exhaustive search. To handle large rotations and angular quantization errors several bins are checked for corners and junctions. For an  $n$ -legged keypoint  $n \cdot 2^{n-1}$  bins are checked corresponding to the  $n$  possible ways one input leg can be matched to the first leg of the stored keypoint and the  $2^{n-1}$  combinations the  $n - 1$  angles can be quantized to the two nearest integer values. Experiments show that the indexing step reduce the number of further comparisons by a factor of 5-20. Next each input keypoint is compared to the stored representations in the union of checked bins. In [8] a fast comparison using a measure of asymmetry was used to further limit the computational burden. Such fast tests are still relevant but are - for simplicity - omitted here.

A query keypoint is compared to a database keypoint by comparing the descriptors of gradient sample values. First the query descriptor is rotated to make the orientation of the first legs of the keypoints match. To eliminate quantization problems three rotation angles, corresponding the nearest three integral sampling numbers, is determined for each ring in the descriptor. Based on a score value the best of the three rotations is chosen independently for each sample in each ring, and a total match score is computed as a weighted sum over the 5 rings. This procedure ensures rotational invariance and that a significant amount of non-trivial deformation can be handled. The weights are chosen inversely proportional to the number of samples in each ring, i.e. inversely proportional to the sample distance. Within each ring the score is computed as a sum of gradient sample differences. Let  $\mathbf{g}^q = (v^q, m^q)$  and  $\mathbf{g}^{db} = (v^{db}, m^{db})$  be the quantized sample values of orientation and magnitude for a query and a database segment, and let  $dm = |m^q - m^{db}|/16$  and  $dv = |v^q - v^{db}|$ , where the value 16 corresponds to the number of magnitude sampling intervals. Then the gradient sample difference is defined by:

$$dist(\mathbf{g}^q, \mathbf{g}^{db}) = \begin{cases} dm \cdot (dv + 1) & \text{if } dv < 2 \text{ and } dm < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

Thus small gradient orientation differences are punished mildly, and larger differences equally hard. Finally, the match score is converted to a value  $\in [0:1]$  with 1 corresponding to a perfect match. The match is accepted if this value is above an empirically determined threshold.

Each query keypoint may be linked to zero, one or a few database keypoints, each link attributed with a match-score. Since in the present study we want to report the semantic contents as specified by the list of training names, a simple voting procedure is applied. Each match votes with its score as strength and has

as many votes as there are names in the list associated with the matched database elements. Then the list of names are sorted according to the accumulated score value and the top-ranking name selected.

For the matches contributing to the highest ranking semantic association, a confidence  $C$  of the naming is estimated. This is based on the total support score  $S_0$  for the most likely naming and computed by:  $C = (S_0 / \sum S_i) \times (1 - \exp(-\frac{S_0^2}{2\sigma^2}))$ . Thus  $C$  will be low if the naming is not unique or if the support score for the naming is not strong.

When training the system with a new image, a list of names is supplied. A keypoint in a new training image is installed if it cannot be matched to any previously seen keypoint. If the keypoint can be matched to a database keypoint, the list of labels for the database keypoint is extended with the new names. The description vector of the database exemplar keypoint is not updated in any way.

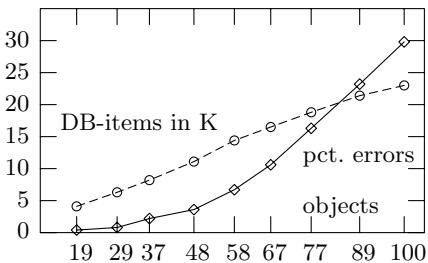
## 6 Experiments

In the experiments reported below the system was first fed with a number of images, each annotated with one label. Then the system was tested by presenting it to new unnamed images and the results of the classification was monitored. In most of the reported experiments the COIL-100-database [7] was used. This contains 100 objects each imaged from a full circle of 72 view positions. Prior to the experiments the images were preprocessed to correct for a 2-pixel column warp-around error, an incorrect constant background value and noise level, and added a supplementary border of 20 pixel width to enable detection of keypoints near the old image border. Also, the images were converted to gray-scale.

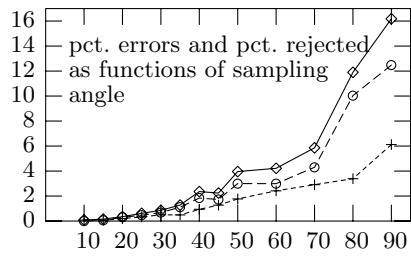
First the performance on the COIL-100-database was tested with respect to the number of objects. For subsets of the 100 objects, 8 views with an angular separation of 45 degrees were used for training and the remaining 64 views for test. First the system was first tested on all 100 objects. Then, approximately 10 objects that had the worst classification results were iteratively removed, and the system was retrained and retested on the smaller database. This procedure results in an optimal assessment curve. In all runs the threshold on the confidence  $C$  was zero, implying that all images were classified. Below in Figure 3 the results are summarized. As expected the performance decreases as the number of objects increases. For less than approximately 50 objects the misclassification rate was below 4 %.

Next, the ability of the system to perform well, when trained on few training images, were tested on subsets of the 37 object images of the COIL-100 database. The test also shows the ability to recognize objects rotated in depth. The ratio of training images to test images was varied from 36/36 to 4/68 corresponding to an angular interval between the training image from 10 to 90 degrees. As before, a confidence level of zero was used. Figure 4 below shows that a misclassification rate below 4 % is possible with as few as 6 training images per object corresponding to an angular separation of 60 degrees. Analysis showed that the average confidence value for correct and false classifications was 0.51 and

0.35 with standard deviations 0.08 and 0.06. The two distributions are highly overlapping making a threshold-based separation difficult. This was typical for other experiments as well. Assuming normal distributions an optimal confidence threshold of 0.32 was found. Using this value will equal the number of accepted false classifications and the number of rejected true classifications. Figure 4 also shows the misclassification rate and the rate of unclassified images for  $C = 0.32$ .

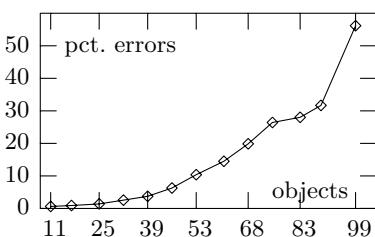


**Fig. 3.** Misclassification rate on subsets of the COIL-100 database, and the size of the build databases in kilo keypoints



**Fig. 4.** Misclassification rate for different number of training images using  $C = 0$  (upper) and  $C = 0.32$  (middle), and the percentage of unclassified images for  $C = 0.32$  (lower)

Rotation in the image plane was tested similarly. For subsets of zero-angle images of the COIL-100 database used for training, the system was tested on 71 synthetically rotated versions of each training image using a rotation step of 5 degree. Figure 5 shows that the misclassification rate was low when less than about 45 objects were to be recognized, and that the misclassification rate increased smoothly until the break-down at about 85 objects. For a small number of objects the recognition rate was independent of the rotation angle. For large databases the recognition rate was slightly better for images rotated approximately a multiplum of 90 degrees. Misclassification may happen when several



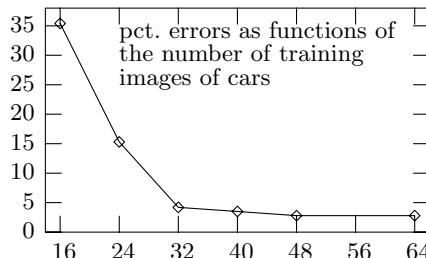
**Fig. 5.** Misclassification rate for images rotated in the image plane as a function of the number of training images

ranking	37 objects	89 objects
1	97.8	76.9
1-2	99.1	83.4
1-3	99.4	86.8
1-4	99.7	88.5
1-5	99.7	89.8
1-6	99.8	90.9
1-7	99.9	91.8
1-8	99.9	92.5
1-9	99.9	93.1
1-10	99.9	93.7

**Fig. 6.** Accumulated histograms of the recognition rate for the top-10 ranking of the correct classification

object share substructures making them alike from certain view angles. In such cases, and when the queries are expected to show several objects, a list of the top-ranking classifications may be useful. For two subsets of objects of the COIL-100 database, the system was trained on 8 images with 45 degree separation, and tested on the remaining images. Then an accumulated histogram of the ranking of the correct classification was constructed. As shown in Figure 6 the correct naming was in most cases among the first few elements in the classification priority list. However, for the larger object subset still many queries seems difficult to classify correctly. One reason is that for these subsets several of the objects were very alike. Another reason is that object names trained from images with many keypoints tend to receive more votes than object names trained from images with few keypoints. Thus images of simple objects may be misclassified. This is due to the simple unnormalized voting scheme. Please note that in general neither normalization with respect to the number of keypoints in each training images nor to the number of identically labeled keypoints seems reasonable. The first choice will penalize shapes trained from images also showing background clutter. The second choice will penalize shapes with large variability. Experiments showed that in general neither type of normalization improved the performance.

Finally, the database build on 8 views of each of 37 objects from the COIL-100 collection was extended with training images of cars in natural (mostly urban) scenes. The latter images as well as the test images were selected from the car database [1]. The training images had a size of  $40 \times 100$  pixels. The 144 query images were much larger and showed a significant amount of background clutter and partial occlusion. Figure 7 shows the amount of misclassification as a function of the number of car images in the training set. For less than 32 training images of cars the recognition rate was poor. This is not surprising because of the difficulty of the images and because the cars may point both left and right and may be lighter or darker than the background (corresponding to 4 different types of cars). For larger training sets the misclassification rate stays constant at a level about 3 %. This is caused by 4-5 images with heavy background clutter coincidentally giving rise to keypoints matching better to some of the distractors (keypoints from the COIL-object-views).



**Fig. 7.** Misclassification rate on images of cars in natural scenes as a function of the number of training images of cars

## 7 Conclusion

An exemplar based recognition scheme using attributed keypoints has been described and a few preliminary experiments has been reported. The results indicate that the system is robust to rotations, limited scalings, noise, small deformations, background clutter, and partial visibility, when the number of objects are limited (e.g. < 50). The stability w.r.t. rotations in depth has been shown to be good, and it has been shown that recognition is possible based on few training images. The results show that good performance is achievable using only local information for keypoint matching. Schmid [9] reports an experiment with 20 objects from the COIL collection, using a 20 degree separation between training as well as test images. We achieve a similar recognition rate of 99.6, but using less than half the number of training images. Much computational effort has been put on the initial keypoint detection leaving fewer - but hopefully more stable and semantically rich - keypoints. It is left for future research to investigate whether this approach is advantageous with respect to the matching success. Automatic learning of significant keypoints as opposed to keypoints caused by background clutter and irrelevant details is of high importance for achieving a good recognition rate and to avoid the database being filled with useless data. In the present approach - having no concept of an object - this might be done by removing rarely used keypoints. The viability of this approach is left for future research.

## References

1. S. Agarwal, A. Awan, D. Roth: *UIUC Image Database for Car Detection*; <http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/>
2. F. Heitger, L. Rosenthaler, R. Von der Heydt, E. Peterhans, O. Kubler: *Simulation of Neural Contour Mechanisms: from Simple to End-stopped Cells*, Vision Research vol. 32, no. 5, 1992, pp. 963-981
3. D. Lowe: *Object Recognition from Local Scale-Invariant Features*, Proc. 7'th ICCV, 1999, pp. 1150-1157
4. D. Lowe: *Local feature view clustering for 3D object recognition*, IEEE Conf. on Computer Vision and Pattern Recognition, Hawaii, 2001, pp. 682-688
5. D. Lowe: *Distinctive Image Features from Scale-Invariant Keypoints* Int. Jour. of Computer Vision 60(2), 2004, pp. 91-110
6. K. Mikolajczyk, C. Schmid: *Scale & Affine Invariant Interest Point Detectors* Int. Jour. of Computer Vision 60(1), 2004, pp.63-86
7. S.A. Nene, S.K. Nayar, H. Murase: *Columbia Object Image Library*, 1996; <http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html>
8. S.I. Olsen: *End-Stop Exemplar based Recognition*, Proceedings of the 13th Scandinavian Conference on Image Analysis, 2003, pp. 43-50.
9. C. Schmid, R. Mohr: *Local Grayvalue Invariants for Image Retrieval*, IEEE trans. PAMI, 19(5), 1997, pp.530-535
10. C. Schmid, R. Mohr, C. Bauckhage: *Evaluation of Interest Point Detectors*, Int. Jour. of Computer Vision 37(2), 2000, pp. 151-172

# Object Localization with Boosting and Weak Supervision for Generic Object Recognition

Andreas Opelt and Axel Pinz

Institute of Electrical Measurement and Measurement Signal Processing,  
Graz University of Technology, Austria  
`{opelt, pinz}@emt.tugraz.at`

**Abstract.** This paper deals, for the first time, with an analysis of localization capabilities of weakly supervised categorization systems. Most existing categorization approaches have been tested on databases, which (a) either show the object(s) of interest in a very prominent way so that their localization can hardly be judged from these experiments, or (b) at least the learning procedure was done with supervision, which forces the system to learn only object relevant data. These approaches cannot be directly compared to a nearly unsupervised method. The main contribution of our paper thus is twofold: First, we have set up a new database which is sufficiently complex, balanced with respect to background, and includes localization ground truth. Second, we show, how our successful approach for generic object recognition [14] can be extended to perform localization, too. To analyze its localization potential, we develop localization measures which focus on approaches based on Boosting [5]. Our experiments show that localization depends on the object category, as well as on the type of the local descriptor.

## 1 Introduction

There is recent success in weakly supervised object categorization from input images (e.g. [4], [14], [8]). Systems can learn based on given piles of images containing objects of certain categories, and piles of counterexamples, not containing these objects. These approaches cope well with the generalization over an object category and perform well in categorization. There are two main aspects in analyzing these approaches with respect to object localization. First, the data needs to be complex enough to challenge a system regarding its localization performance. Second, it is important to discuss the amount of used supervision. Clearly the task of localization becomes easier when one uses a high degree of supervision (e.g. the segmented object) to train the classifier. One might argue that a high degree of supervision during training is similar to human categorization behavior, as humans can easily separate the object of interest from the background. We are interested in designing a vision system that can learn to localize categories with the lowest possible amount of supervision, which should be useful for a broad variety of applications.

In [14] we presented an approach which uses almost no supervision (just the image labels) and also performs well on complex data. This combination brings up the question of localization. As we use Boosting [5] in our categorization approach we want to focus on measuring localization performance related to that learning technique. There are two main contributions of this paper: First, we set up a new image database which is sufficiently complex, balanced, and provides localization ground truth<sup>1</sup>. Second, we define and incorporate localization measures that correspond with the feature selection process during the learning step (based on AdaBoost [5]). object.

## 2 Related Work

The extensive body of literature on generic object recognition reduces if one is also interested in localization. The first group of approaches deals with a tradeoff between generic classification with low supervision and localization performance with higher supervision (e.g. [2], [4], [16]) generally on easier data. Other approaches are really good in localization but just for specific objects (e.g. [15], [8]). Subsequently, we discuss some of the most relevant and most recent results with special emphasis of the problem of localization. The method introduced by Lazebnik et al. [8] is based on semi-local affine parts which are extracted as local affine regions that stay approximately affinely rigid over several different images of the object. The localization performance of that approach is good, but in contrast to our approach they focus on specific object recognition.

In [4] Fergus et al. presented their recent success on object categorization using a model of constellations of parts learned by an EM-type learning algorithm. This leads to a very good recognition performance, but their training images do not have the complexity to show the difficulties in localization with weak supervision. Compared to that our data is highly complex. Learning object relevant data with low supervision from highly cluttered images was discussed by Ruthishauser et al. [15]. On our data their attention algorithm did not work so well. Also the authors do specific object recognition whereas we try to solve the generic problem.

The work by Agarwal et al. [1] solves the problem of localization in a very elegant manner. They localize cars viewed from the side by detecting instances of a sparse, part-based representation. However, they learn their model from sample image portions, which are cut out and show just the objects themselves. In this sense, their approach should be regarded as highly supervised with respect to localization.

Leibe and Schiele [10] also use a sparse, part-based representation forming a codebook for each category. But they add an implicit shape model which enables them to automatically segment the object as a result of their categorization. Having these segments means also that the object is localized. This approach is also scale invariant. In a similar manner as for [1], we notice that localization is less difficult due to the higher degree of supervision in using easier training images.

---

<sup>1</sup> The database used in [14] was not balanced concerning the background in the positive and negative images.

### 3 Method and Data

#### 3.1 Database and Localization Ground Truth

We have set up a new image database (“GRAZ-02”<sup>2</sup>). It contains four categories: Person (P), Bike (B) and Cars (C) and counterexamples (N, meaning that it contains no bikes, no persons and no cars). Figure 1 shows two example images for each of the four categories. This database is sufficiently complex in terms of intra class variation, varying illumination, object scale, pose, occlusion, and clutter, to present a challenge to any categorization system. It is also balanced with respect to background, so that we can expect that a significant amount of learned local descriptors should be located on the objects of interest. So the backdoor of categorizing images of e.g. cars by searching for traffic signs and streets is not easily possible. All relevant objects in all images of categories P, B, and C have been manually segmented. This is not used for training, but provides a localization ground truth which is required for experimental evaluation. Some examples are shown in figure 2.



**Fig. 1.** Two example images for each category of our database (all of these images were correctly categorized using our approach from [14]). Column 1: Bikes (B), 2: Persons (P), 3: Cars (C), 4: counter-class (N)

#### 3.2 Image Categorization

We build on our categorization framework first introduced in [14]. Its localization abilities should be studied, because it shows good results on complex images with no other supervision than the image labels. It is briefly summarized here as a prerequisite to understand the subsequent sections on localization and learning. To train a classifier, the learning algorithm is provided with a set of labeled training images. Note that this is the only amount of supervision required. The object(s) are not pre-segmented, and their location and pose in the images are unknown. The output of the learning algorithm is a

<sup>2</sup> The database and the ground truth are available for download at: <http://www.emt.tugraz.at/~pinz/data>

final classifier  $H(I) = \text{sign}(\sum_{j=1}^T h_j(I)w_{h_j})$  (further on also called “final hypothesis”) which predicts if a relevant object is present in a new image  $I$ . It is formed by a linear combination of  $T$  weak classifiers  $h_j$  each weighted by  $w_{h_j}$ . The output of a weak classifier on an image  $I$  is defined as:  $h_j(I) = 1$  if  $d_M(h_j(I), p_I) \leq th_{h_j}$  and  $h_j(I) = 0$ , otherwise. Here  $th_{h_j}$  denotes the classifiers threshold and  $d_M(h_j(I), p_I)$  defines the minimum distance (here we use the Euclidean distance for SIFTs and Mahalanobis distance for the other descriptors) of the weak classifier  $h_j(I)$  (also called “weak hypothesis”) to all patches  $p_I$  in an image  $I$ . For details on the algorithm see [14] and [5]. The learning procedure works as follows: The labeled images are put into a preprocessing step that transforms them to greyscale. Then two kinds of regions are detected. Regions of discontinuity are the elliptic regions around salient points, extracted with various detectors (Harris-Laplace [12], affine Harris-Laplace [13], DoG [11]). Regions of homogeneity are obtained by using Similarity-Measure-Segmentation [6], and Mean-Shift segmentation [3]. Next, the system calculates a number of local descriptors of these regions of discontinuity and homogeneity (basic moments, moment invariants [7], SIFT descriptors [11], and certain textural moments [6]). These detection and description methods can be combined in various ways. AdaBoost [5] is used as learning technique. The result of the training procedure is saved as the final hypothesis. A new test image  $I$  is categorized by calculating the weighted sum  $H(I)$  of the weak hypotheses that fired. Firing means that  $d_M(h_j(I), p_I) < th_{h_j}$ , as mentioned before. An overview of the image categorization system is depicted inside the framed part of figure 3.

### 3.3 Object Localization

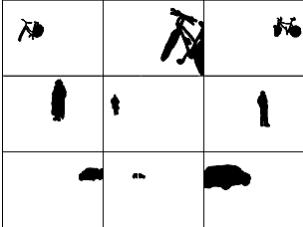
So far the system is able to learn and to categorize. Now, we extend it aiming at object localization and at the possibility to measure the localization performance. Figure 3 shows the extended framework with all additional components highlighted in grey.

Image categorization is based on a vector of local descriptors (of various types, see section 3.2). They can be located anywhere (around salient points or homogeneous regions) in the image. These categorization results lack a systematic investigation in terms of object localization. Which patches are located on the objects, which ones on the background? How is the relation of object vs. background patches? To answer these questions, we define two localization measures  $\lambda_h$  and  $\lambda_d$ , which correspond with the way, features are selected and weighted by AdaBoost.

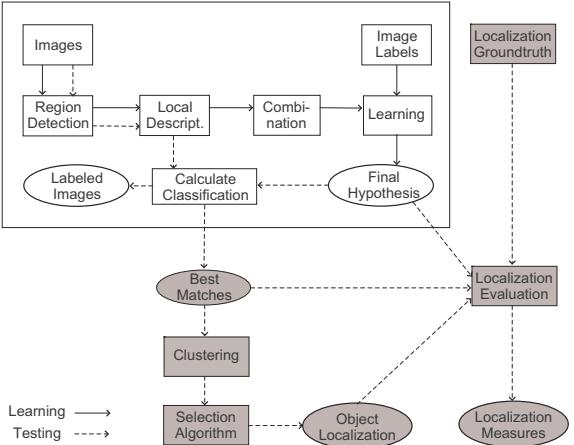
$\lambda_h$  evaluates the localization abilities of a  $\dots, \dots, \dots, \dots, \dots$ :

$$\lambda_h = \frac{\sum_{j=1}^T (w_{h_j} | d_M(h_j, p_I) < th_{h_j}, p_M \in obj)}{\sum_{j=1}^T (w_{h_j} | d_M(h_j, p_I) < th_{h_j}, p_M \notin obj)} \quad (1)$$

Where  $p_M$  is defined as the patch in an image  $I$  with the minimum distance to a certain hypothesis  $h_j$ , and “obj” is the set of points forming the ground truth of image  $I$  (i.e. the pixel coordinates of the segmented object in the ground truth data). Thus, a large value of  $\lambda_h$  depicts a situation, where many patches of a final hypothesis are located on the object, and few ones in the background.



**Fig. 2.** Ground truth examples. Row 1: Bikes, row 2: Persons, row 3: Cars



**Fig. 3.** Our original categorization framework ([14], shown inside the frame), and the extensions for object localization (highlighted in grey)

$\lambda_d$  evaluates the localization in a . . . :

$$\lambda_d = \frac{\sum_{i=1}^m c(I_i|obj)}{\sum_{i=1}^m c(I_i|bg)} \quad (2)$$

with

$$c(I_i|X) = \begin{cases} 1 & \text{if } \sum_{j=1}^T (w_{h_j}|d_M(h_j, p_I) < th_{h_j}, p_M \in X) \\ > \sum_{j=1}^T (w_{h_j}|d_M(h_j, p_I) < th_{h_j}, p_M \notin X) \\ 0 & \text{else} \end{cases}$$

Where  $m$  is the number of test images  $I$  and  $X$  is a set of pixels obtained from ground truth data (we again use ‘‘obj’’ for the set of pixels belonging to the object (ground truth), and ‘‘bg’’ for the others). Thus,  $\lambda_d$  calculates the ratio of the images categorized mainly by object relevant data versus the number of images categorized mainly by contextual information.

$\lambda_h$  enables us to estimate the learned localization abilities, and  $\lambda_d$  gives us an accumulated object localization quality for a number of test cases. But we are also interested in individual localization results. To obtain the localization of the object in a specific test image  $I$  we compute the positions of the best matching description vectors for each weak hypothesis, and calculate spatial clusters using kmeans<sup>3</sup> (see figure 3). Having  $k$  clusters  $C_{cl}, cl = 1, \dots, k$ , the difficult task is to find out which one represents the object location. Our straightforward ‘‘Selection Algorithm’’ consists of the following steps:

<sup>3</sup> One could also use agglomerate clustering here. This would avoid setting a fixed parameter  $k$ , but would introduce the need of a threshold for the agglomerations. However, we set  $k$  to relative small numbers and got good results.

1. Calculate cluster weights  $W_{cl} = \sum_{j=1}^T (w_{h_j} | d_M(h_j, p_I) < th_{h_j}, p_M \in C_{cl})$  for every cluster  $cl = 1, \dots, k$ .
2. Count the number of best matches  $P_{cl}$  in each cluster.
3. Set a cluster rectangle  $R_{cl}$  covering all cluster points for each cluster.
4. Increase the rectangle size by  $e$  pixels on each side.
5. Select the cluster  $C_{max}$  where both,  $W_{cl}$  and  $P_{cl}$  have the highest value. If no such cluster is available take the one where  $P_{cl}$  is maximal (we found that using  $P_{cl}$  instead of  $W_{cl}$  gives better results).
6. If  $R_{C_{max}}$  intersects with other  $R_{cl}$  extend  $R_{C_{max}}$  to cover the intersecting  $R_{cl}$ .
7. If  $R_{C_{max}}$  is closer than  $d$  pixels to another cluster  $R_{cl}$  extend  $R_{C_{max}}$  to cover the intersecting  $R_{cl}$ .
8. Go back to 6. and iterate  $l$  times. If either  $l$  is reached or no further changes occurred in steps 6. and 7. exit with  $R_{C_{max}}$  as object location.

This algorithm delivers an object location in a test image  $I$  which is described by the coordinates of a rectangle  $R_{C_{max}}^I$ . Note that multiple object detection in one image is not possible without a spatial object model. If our data contains multiple objects (just some cases) we aim for the detection of one of the object instances. To measure this effective localization performance we use the evaluation criterion proposed by Agarwal et al. [1]. It describes that the object has to be located within an ellipse which is centered at the true location. If  $(i', j')$  denotes the center of the rectangle corresponding to the true location (ground truth) and  $(i, j)$  denotes the center of our rectangle  $R_{C_{max}}$  then for  $(i, j)$  to be evaluated as correct detection it requires to satisfy

$$\frac{(i - i')^2}{\alpha_{height}^2} + \frac{(j - j')^2}{\alpha_{width}^2} \leq 1, \quad (3)$$

where  $\alpha_{height}, \alpha_{width}$  denote the size of the ellipse. Note that we do not use the measure for a multiscale case as Agarwal et al., because we need to cope with training objects at varying scales.

## 4 Experiments and Results

### 4.1 Parameter Settings

The results were obtained using the same set of parameters for each experiment. All the parameter settings regarding the learning procedure are similar to the ones we used in [14] and [6]. The thresholds for reducing the number of salient points are set to  $t_1 = 30000$  and  $t_2 = 15000$ .

For the localization method we used  $k = 3$  cluster centers. For the selection algorithm the following parameters were used:  $e = 20$ ,  $d = 10$  and  $l = 2$ . For the evaluation criterion of Agarwal et al. [1] we used  $\alpha_{height} = 0.5 \cdot h_{R_{GT}}$  and  $\alpha_{width} = 0.5 \cdot w_{R_{GT}}$  with  $h_{R_{GT}}$  and  $w_{R_{GT}}$  being the height and width of the box delimiting the ground truth of an image.

## 4.2 Image Categorization

For comparison with other approaches regarding categorization, we used the Caltech database. We got better or almost equal results on this rather easy dataset (classification rates ranging between 90% and 99.9%, for details see [14], [6]). From our database we took a training set consisting of 150 images of the object category as positive images and 150 of the counter-class as negative images. The tests were carried out on 300 images half belonging to the category and half not<sup>4</sup>. Table 2 shows the categorization results measured in ROC-equal-error rates of various specific combinations of region extractions and description methods on the three categories of this database. The average ratio of the size of the object versus the image size (counted in number of pixels) is: 0.22 for Bikes, 0.17 for Persons and 0.09 for Cars.

## 4.3 Localization and Localization Measures

Localization performance on easy datasets is good. For example on motorbikes (Caltech) localization gets results above 90%. This data shows the object highly prominent with just little background clutter, what reduces the localization complexity. We thus proceed by presenting localization results for our more complex GRAZ-02 dataset. The left half of table 1 shows the values of the measure  $\lambda_h$  for the various techniques (the same as in table 2) on all three categories. Comparing these results with those in table 2 shows, that even if the categorization performance on the category Persons is good, the framework might use mainly contextual information for classification (e.g. it uses parts of streets or buildings). Focusing on the other two categories one can recognize that SIFTs and Similarity-Measure (SM) also tend to use contextual information, whereas the moment invariants (MI) use more object relevant data. The right half of table 1 shows the results for  $\lambda_d$ . The following clear coherence can be seen. If a high percentage of the weighted weak hypotheses contain object data instead of contextual information (which means  $\lambda_h$  is high), then also the value of  $\lambda_d$  (meaning a new training image was classified mainly by object related information) is high.

**Table 1.** The measures  $\lambda_h$  and  $\lambda_d$  using various description techniques

-	$\lambda_h$				$\lambda_d$			
Data	MI ( $t_1$ )	MI ( $t_2$ )	SIFTs	SM	MI ( $t_1$ )	MI ( $t_2$ )	SIFTs	SM
Bikes	3.0	1.17	0.45	0.85	2.19	2.0	0.5	0.17
Persons	0.28	0.39	0.25	0.39	0.42	0.56	0.12	0.16
Cars	1.13	1.18	0.1	0.25	0.52	0.59	0.06	0.08

To perform useful localization with this weakly supervised system we may require  $\lambda_h > 1.0$ , which just means that a significant number of local descriptors

<sup>4</sup> The images are chosen sequentially from the database. This means we took the first 300 images of an object class and took out every second image for the test set.

**Table 2.** The ROC-equal-error rates of various specific combinations of region extractions and description methods on the three categories of our new dataset (MI ... moment invariants, SM ... Similarity Measure)

Data	MI ( $t_1$ )	MI ( $t_2$ )	SIFTs	SM
Bikes	72.5	76.5	76.4	74.0
Persons	81.0	77.2	70.0	74.1
Cars	67.0	70.2	68.9	56.5

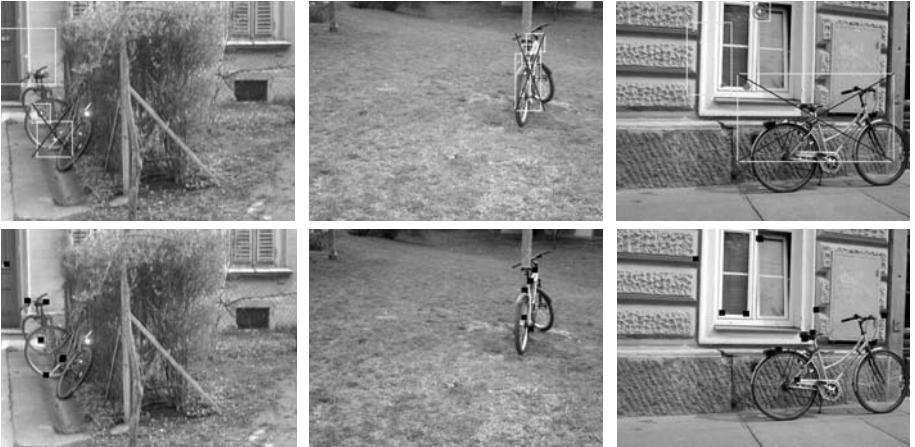
**Table 3.** A comparison of the localization criterion by Agarwal et al. [1] with our ground truth in the first two rows. And additional for Motorbikes (100 images) of the Caltech database in the last row

Data	L(T)	L(F)	L.P.	L+Cat
Bikes	115	35	76.7	56.0
Persons	83	67	55.3	48.2
Cars	72	78	48.0	35.8
Motorbikes	96	4	96.0	88.5

is relevant for object localization. This is also supported by the observation that high values of  $\lambda_d$  correspond with high values of  $\lambda_h$ .

Table 3 shows the results (with Moment Invariants and affine invariant interest points ( $t_1$ )) achieved by comparing the localization measure of Agarwal et al. [1] with our ground truth. The first row (L(T)) shows the number of all positive test images where just the correct localization was measured, not the categorization performance. The second column (L(F)) shows the same rate for the false localizations. The third column (L.P.) shows the localization performance on the test images in percent. Note that values around 50 percent are not close to guessing, regarding that the objects cover just a small region in the images. The last column shows the result in ROC-equal error rate for categorization combined with correct localization. It can be seen that the localization performance on the category Bikes is highest, but even on Persons the performance is surprisingly high. The last row shows that localization is much easier for the simpler Caltech (motorbikes) dataset. To compare with an existing approach we mention the classification performance of 94% achieved by Leibe [9] on this dataset. Their model based approach also localizes the object, but uses high supervision in the training procedure (whereas we use almost no supervision). This is not in contradiction with the results presented in table 1. It just shows that even if a significant number of local descriptors is located in the background (low values for  $\lambda_h$  and  $\lambda_d$ ), the selection of the relevant  $R_{C_{max}}$  is still quite good.

Figure 4 shows examples of the localization of Bikes in test images. The bottom row shows the direct localization with the black squares representing regions with a high probability of the object location (each black square may contain several best matches for firing hypotheses). In the top row we show the effective localization where the light gray squares mark the clusters and the dark gray cross marks the final output  $R_{C_{max}}$  of our Selection Algorithm. Note that we did not use ground truth for this localization. The performance of the Selection Algorithm can be shown as it finds the correct location in images with a high percentage of hypotheses firing on the object (the first two columns) as well as finding the correct location when more hypotheses fire in the background (the third column of figure 4 shows an example). In general the localization often fails when the object appears at a very low scale.



**Fig. 4.** Examples of the localization performance for Bikes

## 5 Summary and Conclusions

In summary, this work shows the first systematic evaluation of weak supervision for a weakly supervised categorization system. Supervision is regarded weak, when labeled training images are used which contain the objects of interest at arbitrary scales, poses, and positions in the images. A further important requirement is a balance of background with respect to different object categories, so that learning of context is inhibited. We have set up a very complex new image database which meets all the above requirements. We also acquired localization ground truth for all relevant objects in all images.

We have extended our categorization system [14] that calculates a large number of weak hypotheses which are based on a variety of interest operators, segmentations, and local descriptors. Learning in this system is based on Boosting. Localization measures have been defined and evaluated which are in correspondence with such a learning approach. Our ‘direct’ localization measures  $\lambda_h$  and  $\lambda_d$  show that even if a balanced database is used, many descriptors are still located in background regions of the images. However, the more general localization measure of Agarwal et al. [1] still yields rather good results (regarding the image complexity). Furthermore, there is a significant intra-class variability. Localization performance is class-dependent. For our database the best localization can be achieved for Bikes, and is much better than the localization for Persons and Cars. On easier datasets like e.g. motorbikes (Caltech) the localization is rather straightforward. This is because the prominence of the object reduces the complexity of a weakly supervised approach to distinguish between object and background.

An important general question might be raised: Have we already reached the frontier of categorization and localization based on local features without using any further model or supervision? We believe, that a general cognitive approach

should avoid more supervision but will require more geometry. Thus, our future research will focus on the learning of sparse geometric models.

## Acknowledgements

This work was supported by the Austrian Science Foundation FWF, project S9103-N04.

## References

1. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11), Nov. 2004.
2. P. Carbonetto, G. Dorko, and C. Schmid. Bayesian learning for weakly supervised object classification. Technical report, INRIA Rhone-Alpes, Grenoble, France, August 2004.
3. D. Comaniciu and P. Meer. Mean shift: A robust approach towards feature space analysis. In *IEEE PAMI*, volume 24(5), pages 603–619, 2002.
4. R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. European Conference of Computer Vision*, pages 242–256, 2004.
5. Y. Freund and R. Schapire. A decision theoretic generalisation of online learning. *Computer and System Sciences*, 55(1):119–139, 1997.
6. M. Fussenegger, A. Opelt, A. Pinz, and P. Auer. Object recognition using segmentation for feature detection. In *Proc. ICPR*, 2004.
7. L. Van Gool, T. Moons, and D. Ungureanu. Affine / photometric invariants for planar intensity patterns. In *Proc. ECCV*, pages 642 – 651, 1996.
8. S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *In Proc. of British Machine Vision Conference*, 2004.
9. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague*, May 2004.
10. B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive means-shift search. In *DAGM'04 Pattern Recognition Symposium, Tuebingen, Germany*, Aug. 2004.
11. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
12. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, pages 525–531, 2001.
13. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, pages 128–142, 2002.
14. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. ECCV*, pages 71–84, 2004.
15. U. Ruthishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *Proc. CVPR*, 2004.
16. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. CVPR*, 2004.

# Clustering Based on Principal Curve

Ioan Cleju<sup>1</sup>, Pasi Fränti<sup>2</sup>, and Xiaolin Wu<sup>3</sup>

<sup>1</sup> University of Konstanz,

Department of Computer and Information Science,

Fach M697, 78457 Konstanz, Germany

cleju@inf.uni-konstanz.de

<sup>2</sup> University of Joensuu,

Department of Computer Science, P. O. Box 111,

80110 Joensuu, Finland

franti@cs.joensuu.fi

<sup>3</sup> McMaster University,

Department of Electrical and Computer Engineering,

L8G 4K1, Hamilton, Ontario, Canada

xwu@mail.ece.mcmaster.ca

**Abstract.** Clustering algorithms are intensively used in the image analysis field in compression, segmentation, recognition and other tasks. In this work we present a new approach in clustering vector datasets by finding a good order in the set, and then applying an optimal segmentation algorithm. The algorithm heuristically prolongs the optimal scalar quantization technique to vector space. The data set is sequenced using one-dimensional projection spaces. We show that the principal axis is too rigid to preserve the adjacency of the points. We present a way to refine the order using the minimum weight Hamiltonian path in the data graph. Next we propose to use the principal curve to better model the non-linearity of the data and find a good sequence in the data. The experimental results show that the principal curve based clustering method can be successfully used in cluster analysis.

## 1 Introduction

Clustering is a general classification procedure that divides a set of objects in different classes. Objects from the same class should be similar to each other. There is no indication about their number but only the properties of the objects in the set. Clustering algorithms are used in a multitude of applications mostly related to pattern matching, image processing, data mining, information retrieval or spatial data analysis. In image analysis, the interest for clustering algorithms comes from tasks as compression, segmentation or recognition. The NP-completeness of the problem [1] motivated the research for good and fast heuristics.

The scalar quantization, a special case of clustering problem, can be optimally solved in linear time [2, 3]. The main difference to vector quantization is that the scalar space is naturally ordered and optimal clusters are always subsequences of the dataset. The main contribution of this work is a new heuristic clustering method, based on the principal curve, which orders the dataset and applies a segmentation algorithm similar to the one used for scalar quantization.

## 1.1 Problem Definition

The result of the clustering is a *partitioning* of the original set that maps each data point to its class, or cluster. The quality of the clustering is defined by the objective function  $f$  that assigns a real number to each possible clustering. Thus, the clustering problem refers to finding a clustering that optimizes the objective function. The size of the clustering is defined as the number of different clusters. The *K-clustering* problem is a simplification of the original clustering problem and refers to finding a clustering of a given size  $K$ .

In this work we will consider only the K-clustering problem, although for simplicity we might refer to it as just the clustering problem. As the objective function, we will use the *mean squared error* (MSE). In this sense, it is of interest to designate to each cluster a unique representative, and assign the value of the representative to the mean vector of the cluster (centroid). The set of representatives for every cluster defines the *codebook*. The elements of the codebook are called *code-vectors*. The error representation for a point is defined as the distance from the point to its corresponding code-vector. The goal is to find a clustering that minimizes the MSE. Notice that on these considerations, a specification for one of partitioning, clustering or codebook will uniquely determine the other two.

## 1.2 Related Work

The clustering algorithms are very diverse, a good overview can be found in [4]. At the top level, the different approaches can be classified as hierarchical and partitional. Hierarchical algorithms produce a series of partitions, known as dendrogram, while partitional ones produce one partition. The dendrogram can be constructed top-down, by hierarchical divisive algorithms, or bottom-up by agglomerative clustering algorithms. The most used algorithms in cluster analysis are squared error algorithms, such as K-means [5]. The resulted clustering is highly dependent on the initialization and therefore K-means is usually used to fine-tune a solution given by another algorithm. Graph theoretic algorithms model the data as a graph (e.g. minimum spanning tree) and delete the edges that are too expensive [6]. Mixture resolving approaches assume that the data was generated by an unknown distribution and try to determine its parameters [7]. Fuzzy algorithms [8] and artificial neural networks [9] have been successfully used in clustering. New approaches using genetic algorithms give also good results [10].

## 1.3 Overview of Our Work

In this work we study different possibilities to reformulate the clustering problem as order constrained clustering [11]. The latter problem can be optimally solved in polynomial time. Section 2 defines order constrained clustering, shows that scalar quantization is a special case of order constrained clustering and explains a method for solving it optimally. A possibility to apply a similar method on vector spaces using the clustering based on principal axis algorithm is described in subsection 2.1. The approach has been applied in literature [12], but we extend it introducing a sequence tuning method based on minimum weight Hamiltonian path, in subsection 2.2. The main contribution of this work, presented in section 3, describes our new

clustering algorithm based on the principal curve. Section 4 presents the results and section 5 provides the conclusions and future development possibilities.

## 2 Order Constrained Clustering

*Order constrained clustering* is a special case of clustering [11]. The set is ordered and the clusters are subsequences of the dataset sequence. The problem can be optimally solved in polynomial time [11], however the clustering is usually optimal only in the context of the order constraint.

Scalar quantization can be formulated as order constrained clustering considering the order in the scalar space. Due to the convex hull of the optimal clusters, the solution for order constrained scalar quantization is optimal for the unconstrained problem as well. Scalar quantization can be solved using the *minimum weight K-link path* problem [13]. An oriented graph is constructed for the ordered set, containing edges from any node to all the nodes that appear later in the sequence. The weight of an edge is equal to the distortion of one cluster that contains all the data points between the corresponding nodes. The minimum weight path from the first to the last node, consisting of  $K$  edges, corresponds to the optimal clustering of the set. It can be optimally found by a dynamic programming procedure in linear time [2, 3].

The order constrained clustering problem in vector space can be formulated as the minimum weight K-link problem as well. The quality of the result strongly depends on the order relation. Different possibilities to order the dataset will be studied next (see Fig. 1, Fig. 3).

### 2.1 Clustering Based on Principal Axis

The dataset can be sequenced using the projections over a one-dimensional space (see Fig. 1). The principal axis of the set, computed using the first principal component [14], is a linear space that probably contains the most information, as it maximizes the dispersion of data.

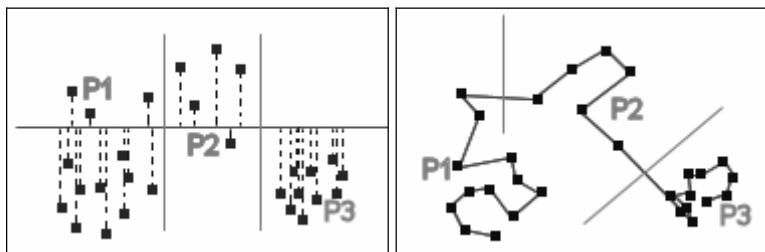
The method has been applied to color quantization in [12]. Principal axis based clustering (PAC) gives good results only if the dispersion of the data along the principal axis is significantly higher than the dispersion in other directions, so that the dataset fits the linear model. It can be observed (see Fig. 2) that the code-vectors are close to the principal axis. The improvement obtained by K-means tuning is considerable but the solution is dependent on the initialization clusters.

### 2.2 Revising the Order by Hamiltonian Shortest Path

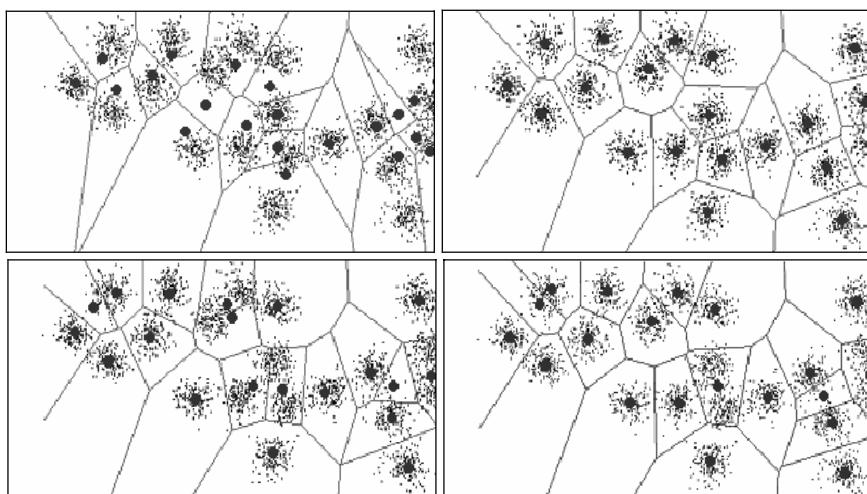
The quality of the order depends on how the spatial relations of the data points are maintained during the projection to the one-dimensional space. The projection on the principal axis introduces great distortions in preserving the spatial relations. We therefore aim at tuning the order by minimizing the total weight of the sequence path.

We will consider the order given by the principal axis as an approximation for the *minimum weight Hamiltonian path* (MWHP), and try to improve it by simple heuristics (see Fig. 1), as finding the minimum weight Hamiltonian path in a graph is an NP-complete problem [15]. First, we consider short subsequences for which the

minimum weight path can be found easily, and optimize the subsequence path length correspondingly. As the size of the subsequences iteratively increases, we do not look for the optimal path but for approximations.



**Fig. 1.** Examples showing clustering based on principal axis (*left*) and minimum weight Hamiltonian path (*right*)



**Fig. 2.** Results of the algorithms based on principal axis projection: PAC (*upper left*), PAC tuned by K-means (*upper right*), PAC tuned by MWHP (*bottom left*) and PAC tuned by MWHP and K-means (*bottom right*)

The new clustering result shows significant improvement, but after K-means tuning it is not necessarily better (see Fig. 2). The order given by the principal axis assures a good dispersion of code-vectors along its direction; PAC is advantageous to use if the data dispersion is significant only in principal axis direction. It seems that the best order should be flexible enough to capture the global layout of the clusters but not too detailed, in order to prevent the path jumping between clusters, or going through the same cluster several times.

### 3 Clustering Based on Principal Curve

As the principal axis is a linear model too rigid to fit the data, we propose to use *principal curves* [16, 17, 18, 19, 20, 21] to order the dataset. The principal curves are one-dimensional spaces that can capture the non-linearity from the data and can better preserve the adjacency of data points. Fig. 3 shows an example of the new method. The main steps are:

- construction of the curve,
- projection over the curve and forming the data sequence,
- order constrained clustering, and
- constructing the Voronoi cells.

There are different definitions for the principal curve. The initial approach intuitively describes it as a smooth one-dimensional curve that passes through the “middle” of data [16]. The curve is self-consistent, smooth and does not intersect itself. An application that uses closed principal curves to model the outlines of ice floes in satellite images is developed in [17]. An improved variant that combines the former approaches is applied to classification and feature extraction [18]. Principal curves are defined in [19] as continuous curves of a given maximal length, which minimize the squared distance to the points of the space. An incremental method similar to K-means for finding principal curves is developed in [20]. Instead of considering the total length of the curve as in [19], the method in [21] constrains the sum of the angles along the curve (the total turn).

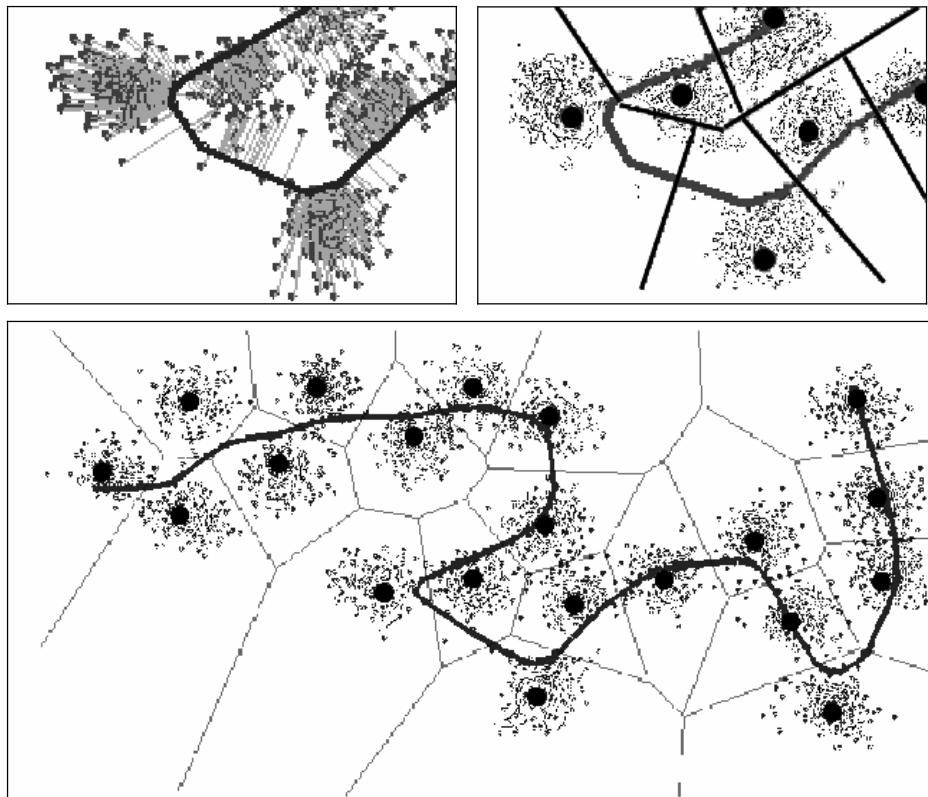
We propose to use the *principal curve with length constraint*, as it is defined in [19], to project and order the dataset; from now on we will simply refer to this curve as principal curve. The principal curve minimizes the distortion of the points to the curve. This assures that the adjacency of the points can be preserved on the curve.

The correspondence between the principal curves and Kohonen’s self-organizing maps [9] has been pointed out in the literature [22]. However the algorithm for principal curve with length constraint substantially differs from the ones that are based on self-organizing maps; the distances to the data points are measured from the vertices and the segments as well, and not only from the vertices [19].

The practical learning algorithm of the principal curve constructs a sub-optimal polygonal line approximation of the curve. The main difference to the theoretical algorithm is that the length is not provided as a parameter, but the algorithm optimizes it. The procedure is iterative, each step one segment is split and then all the segments of the polygonal line are optimized.

In the segment optimization step, each vertex position is separately optimized, keeping all the other vertexes fixed. A Lagrangian formulation that combines the squared error of the data points that project on adjacent edges and the local curvature is minimized. The smoothing factor that weights the local curvature measure is heuristically found. It can be controlled by the penalty coefficient. The modification of the penalty coefficient determines the shape and indirectly controls the length of the curve. A small value will determine a very long curve, at the limit just a path in the dataset that has null error. A very large value will determine a shape very similar to the principal axis. The stopping criterion also uses a heuristic test that is controlled by a parameter.

After the set is ordered along the principal curve, dynamic programming is applied and the optimal clustering for the constrained clustering is found. Voronoi cells are formed and K-means can be iterated. As shown in Fig. 4, for a good parameterization of the curve the results are very close to the local optimum.



**Fig. 3.** Projection of data points over the principal curve (*upper left*), optimal segmentation of the sequence (*upper right*) and final clusters (*bottom*)

### 3.1 Choice of Parameters

Except for the number of clusters, the parameters for the clustering algorithm are used for the curve construction. One parameter influences the stopping criterion and another one influences the curvature (the penalty coefficient). The parameter that controls the stopping criterion does not significantly influence the clustering result and the value provided in [19] is good for our purpose.

Changes of the penalty coefficient influence the shape of the curve and the clustering (see Fig. 4). Longer curves allow data points from one cluster to project on several regions of the curve, while shorter curves allow different clusters to project on overlapping regions. This coefficient must therefore be tuned depending on the data.

Our experiments showed that the value proposed in [19] to 0.13 does not provide the best results in clustering (see Section 4).

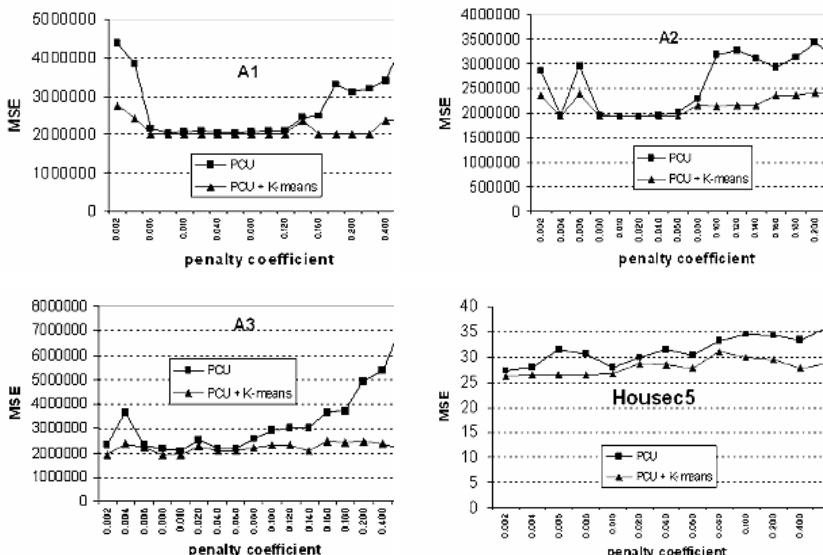
### 3.2 Complexity of the Method

The principal curve algorithm has the complexity of  $O(N^{5/3})$ . It can be reduced to  $O(SN^{4/3})$  if  $S$ -segments polygonal approximation of the curve is considered. The overall complexity for clustering is  $O(KN^2)$  and it is given by the dynamic programming technique applied to order constrained clustering.

## 4 Experimental Results

For the experiments we have used three types of datasets. The A datasets (A1, A2, A3) are artificial and contain different numbers of two-dimensional Gaussian clusters having about the same characteristics. The S sets (S1, S2, S3, S4) are two-dimensional artificial datasets containing clusters with varying complexity in terms of spatial data distributions. The real datasets (House, Bridge, Camera, Missa) come from image data, representing color information (House), 4×4 non-overlapping blocks of gray image (Bridge and Camera) and 4×4 difference blocks of two subsequent frames in the video sequence (Missa). Correspondingly, the datasets have 3 and 16 dimensions.

The experiments have been carried out for 15 different values of the penalty coefficient, ranging from 0.001 to 0.22. For the artificial datasets that present clusters in the data, the MSE as a function of penalty coefficient clearly has a minimum. Good values are obtained for the penalty coefficient in the range 0.01 to 0.08 (see Fig. 4).



**Fig. 4.** Results (MSE) for principal curve clustering (PCU) as a function of the penalty coefficient

For data that does not have the evidence of clusters, the minimum is not clear and good clustering can be obtained for lower values of the penalty coefficient as well.

The comparative results include K-means and *randomized local search* (RLS) [23]. The K-means MSE values are the best results obtained by 10 repeated trials. The values for the RLS method have been considered when the value of the MSE stabilizes; a slightly better solution is found after a larger number of iterations. The results for clustering based on principal axis (PAC) and principal curve (PCU), with and without the K-means tuned versions, are compared. The MSE value for the principal curve clustering was chosen as the best for the different penalty coefficients. Numerical results are shown in Tables 1, 2 and 3.

**Table 1.** Comparison of the results for the A datasets

Method	A sets		
	A1 (*10 <sup>5</sup> )	A2 (*10 <sup>5</sup> )	A3 (*10 <sup>5</sup> )
K-means	20.24	19.32	19.29
RLS	20.24	19.32	19.29
PAC	83.00	156.57	176.59
PAC + K-means	20.24	27.41	36.95
PCU	20.30	19.33	20.59
PCU + K-means	20.24	19.32	19.29

**Table 2.** Comparison of the results for the S datasets

Method	S sets			
	S1 (*10 <sup>7</sup> )	S2 (*10 <sup>8</sup> )	S3 (*10 <sup>8</sup> )	S4 (*10 <sup>8</sup> )
K-means	134.44	13.27	16.88	15.70
RLS	89.17	13.27	16.88	15.70
PAC	840.48	77.34	57.11	63.40
PAC + K-means	143.54	18.65	16.88	15.70
PCU	89.18	13.29	16.94	15.91
PCU + K-means	89.17	13.27	16.88	15.70

**Table 3.** Comparison of the results for the image datasets

Method	Real datasets			
	House	Bridge	Camera	Missa
K-means	36.4	365	278	9.64
RLS	35.6	364	270	9.50
PAC	51.6	430	355	13.07
PAC + K-means	39.3	366	276	10.05
PCU	37.3	377	295	9.99
PCU + K-means	36.1	365	273	9.69

The results of the principal curve clustering are significantly better than those based on principal axis. This is especially observed for the more complicated datasets A and S, where PCU performs better than PAC + K-means. The difference between the two methods reduces for the real datasets. The real datasets present high linear correlation that enables the PAC algorithm to obtain a good result.

The comparison with repeated K-means and RLS show that the results are very close to each other and to the global optimum (see the repeating values in the tables that represent the global optimum as well).

## 5 Conclusions

In this work we have considered solving the clustering problem using one-dimensional projections of the dataset. We have continued studying the principal axis clustering and provide a possibility to revise the sequence in the sense of minimum weight Hamiltonian path.

The main part of the work concentrates on clustering along the principal curve. The principal curve has the advantage of being a non-linear projection that can model diverse types of data. The tests have shown that the principal curve can be successfully applied in clustering for complex datasets. The method proves to perform also on multidimensional datasets.

For future, the parameterization of the principal curve should be improved by automatically optimizing the value of the penalty coefficient; the number of clusters should be considered as well.

## Acknowledgments

The work is based on the Master's thesis of Ioan Cleju for his M.Sc. degree under 'International Master's Program in Information Technology (IMPIT)' supported by the University of Joensuu, Finland. The work of Xiaolin Wu was supported in part by NSERC, NSF and Nokia Research Fellowship. Ioan Cleju's work is currently supported by DFG Graduiertenkolleg/1042 'Explorative Analysis and Visualization of Large Information Spaces' at University of Konstanz, Germany.

## References

1. Slagle, J.L., Chang, C.L., Heller, S.L.: A Clustering and Data-Reorganization Algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 5 (1975) 121-128
2. Wu, X.: Optimal Quantization by Matrix Searching. *Journal of Algorithms*, Vol. 12 (1991) 663-673
3. Soong, F.K., Juang, B.H.: Optimal Quantization of LSP Parameters. *IEEE Transactions on Speech and Audio Processing*, Vol. 1 (1993) 15-24
4. Jain, A.K., Murty, M. N., Flynn, P.J.: Data Clustering: A review. *ACM Computing Surveys*, Vol. 31 (1999)

5. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1 (1967) 281-296
6. Zahn, C.T.: Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. IEEE Transactions on Computers (1971) 68-86
7. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, New Jersey (1988)
8. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the Fuzzy c-Means Clustering Algorithm. Computers and Geosciences, Vol. 10 (1984) 191-203
9. Kohonen, T.: Self-Organizing Maps. Springer, Berlin Heidelberg (1995)
10. Fränti, P.: Genetic Algorithm with Deterministic Crossover for Vector Quantization. Pattern Recognition Letters, Vol. 21 (2000) 61-68
11. Gordon, A.D.: Classification. Chapman and Hall, London (1980)
12. Wu, X.: Color Quantization by Dynamic Programming and Principal Analysis. ACM Transactions on Graphics, Vol. 11 (1992) 348-372
13. Aggarwal, A., Schieber, B., Tokuyama, T.: Finding a Minimum Weight K-link Path in Graphs with Monge Property and Applications. In: Proceedings of the 9th Annual Symposium on Computational Geometry (1993) 189-197
14. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall, New Jersey (1988)
15. Garey, M., Johnson, D.: Computers and Intractability: A Guide to NP-Completeness. W.H. Freeman, New York (1979)
16. Hastie, T., Stuetzle, W.: Principal Curves. Journal of the American Statistical Association, Vol. 84 (1989) 502-516
17. Banfield, J.D., Raftery, A.E.: Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering about Principal Curves. Journal of the American Statistical Association, Vol. 87 (1992) 7-16
18. Chang, K., Ghosh, J.: Principal Curves for Non-Linear Feature Extraction and Classification. In: Proceedings SPIE, (1998) 120-129
19. Kegl, B., Krzyzak, A., Linder, T., Zeger, K.: Learning and Design of Principal Curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22 (2000) 281-297
20. Verbeek, J.J., Vlassis, N., Kroese, B.: A k-Segments Algorithm for Finding Principal Curves. Pattern Recognition Letters, Vol. 23 (2002) 1009-1017
21. Sandilya, S., Kulkarni, S.R.: Principal Curves with Bounded Turn. IEEE Transactions on Information Theory, Vol. 48 (2002) 2789-2793
22. Mulier, F., Cherkassky, V.: Self-organization as an Iterative Kernel Smoothing Process, Neural Computation, Vol. 7 (1995) 1165-1177
23. Fränti, P., Kivijäri, J.: Randomized Local Search Algorithm for the Clustering Problem. Pattern Analysis and Applications, Vol. 3 (2000) 358-369

# Block-Based Methods for Image Retrieval Using Local Binary Patterns

Valtteri Takala, Timo Ahonen, and Matti Pietikäinen

Machine Vision Group, Infotech Oulu, PO Box 4500,  
FI-90014 University of Oulu, Finland  
[{vallu, tahonen, mfp}@ee.oulu.fi,](mailto:{vallu, tahonen, mfp}@ee.oulu.fi)  
<http://www.ee.oulu.fi/mvg/>

**Abstract.** In this paper, two block-based texture methods are proposed for content-based image retrieval (CBIR). The approaches use the Local Binary Pattern (LBP) texture feature as the source of image description. The first method divides the query and database images into equally sized blocks from which LBP histograms are extracted. Then the block histograms are compared using a relative  $L_1$  dissimilarity measure based on the Minkowski distances. The second approach uses the image division on database images and calculates a single feature histogram for the query. It sums up the database histograms according to the size of the query image and finds the best match by exploiting a sliding search window. The first method is evaluated against color correlogram and edge histogram based algorithms. The second, user interaction dependent approach is used to provide example queries. The experiments show the clear superiority of the new algorithms against their competitors.

## 1 Introduction

Content-based image retrieval (CBIR) has gained a reasonable amount of interest in recent years. The growing number of image and video databases in the Internet and other information sources has forced us to strive after better retrieval methods. There is certainly a continuous need for novel ideas in all areas of CBIR.

While choosing feature descriptors for image retrieval we have several choices to begin with. The most common categories of descriptors are based on color, texture, and shape, and there are many alternatives in each of these. The popular color features in today's content-based image retrieval applications include color histograms [1], color correlograms [2], color moments [3] and MPEG-7 color descriptors [4]. As for the texture feature extractors, that have been under research since the late 1960s, there exist numerous methods. Many approaches like two of the MPEG-7 texture descriptors [4] are based on Gabor filtering [5]. Others put their trust on algorithms that rely on DFT transformation [6]. There are also usable features like MR-SAR [7], Wold features [8] and, of course, the old but still popular Tamura approach [9]. In addition to the previous ones, simple algorithms based on statistical feature distribution have proved to be efficient

in texture classification. For example, the edge histogram [10], which is a block-based descriptor included in the MPEG-7 standard, LBP [11], and its derivative LEP [12] have been in successful use.

LBP is one of the best texture methods available today. It is invariant to monotonic changes in gray-scale and fast to calculate. Its efficiency originates from the detection of different micro patterns (edges, points, constant areas etc.). LBP has already proved its worth in many applications [13], [14], [15] in which texture plays an important role. There already exist some CBIR platforms with LBP features included, but the use of the operator has been limited to the original version [11] and it has been applied on full images only [16].

Most of the current CBIR texture descriptors used in commercial systems are calculated for full images. The full image approach is well justified as it usually keeps the size of the feature database reasonably low – depending on the used features and the amount of images, of course. Still there is a problem while considering only full images. The local image areas of interest are easily left unnoticed as the global features do not contain enough information for local discrimination. A way to pay attention to local properties is to use image segmentation. However, the segmentation is usually prone to errors so it is not very suitable for images with general – in other words unknown – content. Another way to enhance the retrieval results is to apply the image extractor to the subimage areas without using any type of segmentation and compare the obtained feature descriptors separately. For instance, in [17] five constant subimage zones were used with several different features. In this paper a similar kind of approach is used, but instead of constant areas it is extended to arbitrary-sized image blocks which can be overlapping.

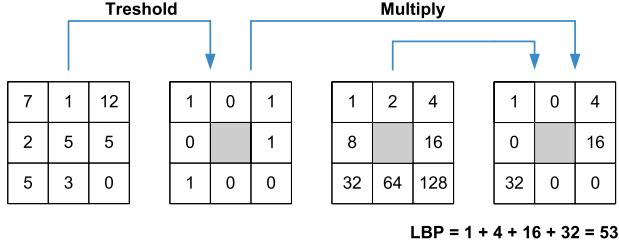
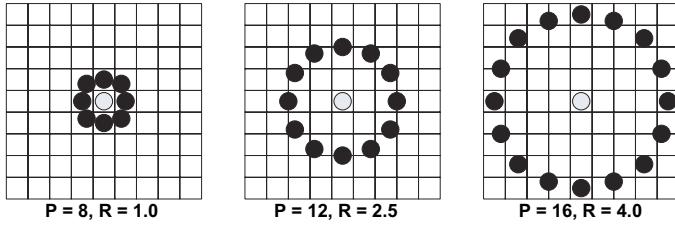
## 2 Texture Descriptor

### 2.1 LBP

The original LBP method [11], shown in Fig. 1, was first introduced as a complementary measure for local image contrast. It operated with eight neighboring pixels using the center as a threshold. The final LBP code was then produced by multiplying the thresholded values by weights given by powers of two and adding the results in a way described by Fig. 1. By definition, LBP is invariant to any monotonic transformation of the gray scale and it is quick to compute.

The original LBP has been extended to a more generalized approach [18] in which the number of neighboring sample points is not limited. In Fig. 2 three different LBP operators are given. The predicate ( $\text{radius}$ ,  $R$ ) has no constraints like in the original version and the samples ( $P$ ) that do not fall on exact pixel positions are interpolated by using bi-linear interpolation.

With larger neighborhoods, the number of possible LBP codes increases exponentially. This can be avoided, to some extent, by considering only a subset of the codes. One approach is to use so called uniform patterns [18] representing the statistically most common LBP codes. With them the size of the feature

**Fig. 1.** The original LBP**Fig. 2.** The general  $\text{LBP}_{P,R}$  with three different circularly symmetric neighbor sets, where subscript  $P$  stands for the number of samples and  $R$  for the sampling radius

histogram generated by an LBP operator can be reduced without significant loss in its discrimination capability. For example, if we consider only those LBP codes that have U value of 2 (U refers to the measure of uniformity, that is the number of 0/1 and 1/0 transitions in the circular binary code pattern), in case of a  $3 \times 3$  operator ( $\text{LBP}_{8,1}^{u2}$ ) we get a feature vector of 58 bins instead of original 256 bins. When the remaining patterns are accumulated to a single bin the histogram becomes 59. That is only a fraction (59/256) of the original.

The spatial support area of LBP feature extractor can be extended by using operators with different radii and sample counts and combining the results [18]. By utilizing N operators we get N different LBP codes which can be connected to form a single feature descriptor vector of N codes. While inserting the marginal distributions of feature extractors one after another, the distance between the sample and model is given by Eq. 1:

$$L_N = \sum_{n=1}^N L(S^n, M^n), \quad (1)$$

where  $S^n$  and  $M^n$  are the sample and model distributions extracted by the  $n$ th operator.

## 2.2 Nonparametric Dissimilarity Measure

A distance function is needed for comparing images through their LBP features. There are many different dissimilarity measures [19] to choose from. Most of the

LBP studies have favored a nonparametric log-likelihood statistic as suggested by Ojala et al. [18]. In this study, however, a relative  $L_1$  measure similar to the one proposed by Huang et al. [2] was chosen due to its performance in terms of both speed and good retrieval rates when compared to the log-likelihood and other available statistics. In the initial tests the log-likelihood and relative  $L_1$ , which were clearly better than the rest, produced even results but the calculation of relative  $L_1$  measure took only a third of the time required by log-likelihood. The dissimilarity measure is given in Eq. 2, where  $\mathbf{x}_1$  and  $\mathbf{x}_2$  represent the feature histograms to be compared and subscript  $i$  is the corresponding bin.

$$L_1^{relative}(\mathbf{x}_1, \mathbf{x}_2) = \sum_i \frac{|\cdot_{1,i} - \cdot_{2,i}|}{\cdot_{1,i} + \cdot_{2,i}}, \quad (2)$$

### 3 Block-Based CBIR

#### 3.1 The Block Division Method

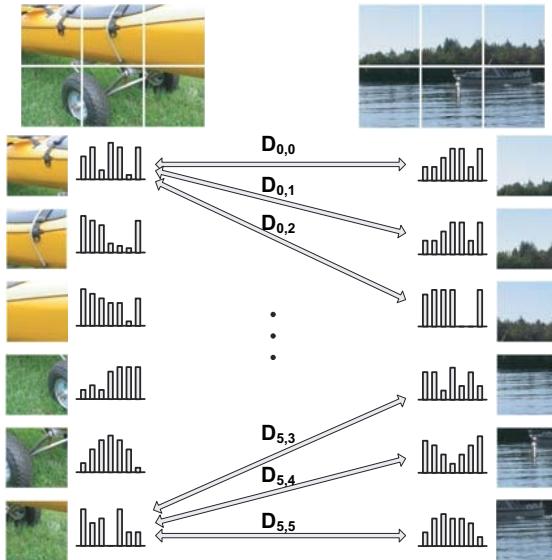
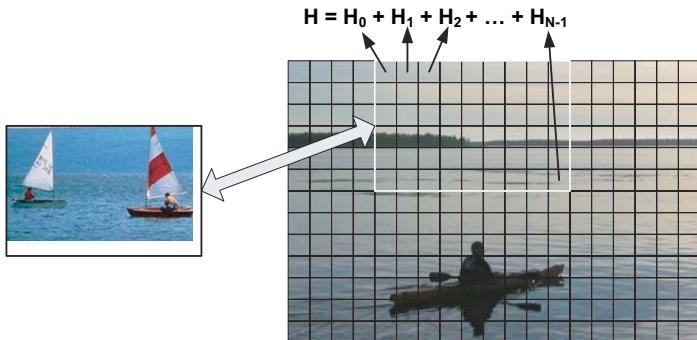
The block division method is a simple approach that relies on subimages to address the spatial properties of images. It can be used together with any histogram descriptors similar to LBP. The method works in the following way: First it divides the model images into square blocks that are arbitrary in size and overlap. Then the method calculates the LBP distributions for each of the blocks and combines the histograms into a single vector of sub-histograms representing the image. In the query phase the same is done for the query image(s) after which the query and model are compared by calculating the distance between each sub-histogram of the query and model. The final image dissimilarity  $D$  for classification is the sum of minimum distances as presented by Eq. 3:

$$D = \sum_{i=0}^{N-1} min_j(D_{i,j}), \quad (3)$$

where  $N$  is the total amount of query image blocks and  $D_{i,j}$  the distance (relative  $L_1$ ) between the  $i$ th query and  $j$ th model block histograms. An example of the approach in operation is shown in Fig. 3. Note, that in this figure the shown histograms are only examples and not the actual LBP distributions of corresponding image blocks.

#### 3.2 The Primitive Blocks

Another way to utilize image blocks is to use small constant-sized elements referred here as primitive blocks. Instead of larger and less adaptive equivalents, the primitive blocks can be combined to match the size of the query image with reasonable accuracy and speed as there is no heavy processing involved like in the pixel-by-pixel sliding window methods. In this approach the model images are handled as in the previous method but the query images are left untouched

**Fig. 3.** The block division method**Fig. 4.** The primitive blocks approach

and only a global LBP histogram is produced for each of them. The model's sub-histograms  $\mathbf{H}_i$  are summed up to a single feature histogram according to

$$\mathbf{H} = \sum_{i=0}^{N-1} \mathbf{H}_i, \quad (4)$$

by first adapting to the size of the query image, and then they are normalized. The primitive blocks (actually the corresponding block histograms) are connected in the way depicted in Fig. 4, where the search window goes through the whole model image and does the matching by using the distance measure of

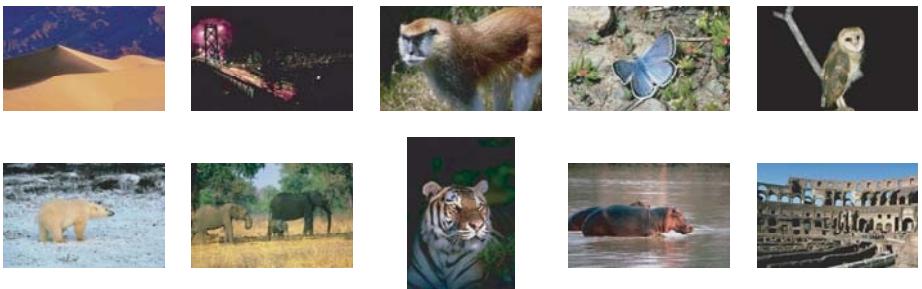
Eq. 2. The size of the search window is the same or a bit larger, depending on the chosen block size, than the area dictated by the query image dimensions.

While using primitive blocks, there exist two types of overlapping. The blocks themselves can overlap, as in the case of previous block division method, and then the measure of overlapping is determined in single pixels. The same applies to the search window areas consisting of primitive blocks but in their case the overlap is quantified to the dimensions of the used primitive blocks.

## 4 Experiments

### 4.1 Test Database and Configurations

Both image retrieval methods were tested on a database consisting of commercial Corel Image Gallery [20] images of sizes  $384 \times 256$  and  $256 \times 384$ . The image categorization was set according to the original image database structure of the Corel set, so there were 27 categories of 50 images each making up 1350 images in total. No further categorization was utilized. This kind of categorization may sound rude but it was used to ensure the reproducibility of the tests. The following categories (physical database folder names) where chosen from the image gallery: Apes, Bears, Butterflies, Cards, Death Valley, Dogs, Elephants, Evening Skies, Fancy Flowers, Fireworks, Histology, Lighthouses, Marble Textures, Night Scenes, Owls, Rhinos and Hippos, Roads and Highways, Rome, Skies, Snakes Lizards and Salamanders, Space Voyage, Sunsets Around the World, Tigers, Tools, Waterscapes, Wildcats, and Winter. Some example images are shown in Fig. 5.



**Fig. 5.** Image examples from the Corel Gallery database

The category experiments were carried on five different image categories (Apes, Death Valley, Fireworks, Lighthouses, and Tigers), so there were 250 queries per experiment. Two different image feature descriptors, one based on color and the other one on texture, were chosen to be compared to the queries attained with LBP operators. The first one of them was the color correlogram [2], which is still one of the most powerful color descriptors, and the other one

was the edge histogram [10], that operates with image blocks and is included in the MPEG-7 Visual Standard [4]. The correlogram was applied with four distances (1, 3, 5, and 7) and four quantization levels per color channel, that is 64 quantization levels in total. The edge histogram used the standard parameters as used by Park et al. [21], thus the method produced histograms of 80 bins.

LBP was applied both on full images and image blocks of sizes  $128 \times 128$  and  $96 \times 96$ . Two clearly different LBP operators were tried out: one using eight uninterpolated samples and a predicate of 1 ( $\text{LBP}_{8,1}^{u2}$ ) and a multiresolution version with three different radii and eight interpolated samples in each ( $\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$ ). Both operators relied on uniform patterns with U value of 2 ( $u2$ ), so the corresponding histograms had 59 and 177 bins, respectively.

## 4.2 Results

The results of the experiments are shown in Table 1. The used measures are precision (the ratio between the correct and all retrieved images) and recall (the ratio between the correct retrieved images and all correct images in the

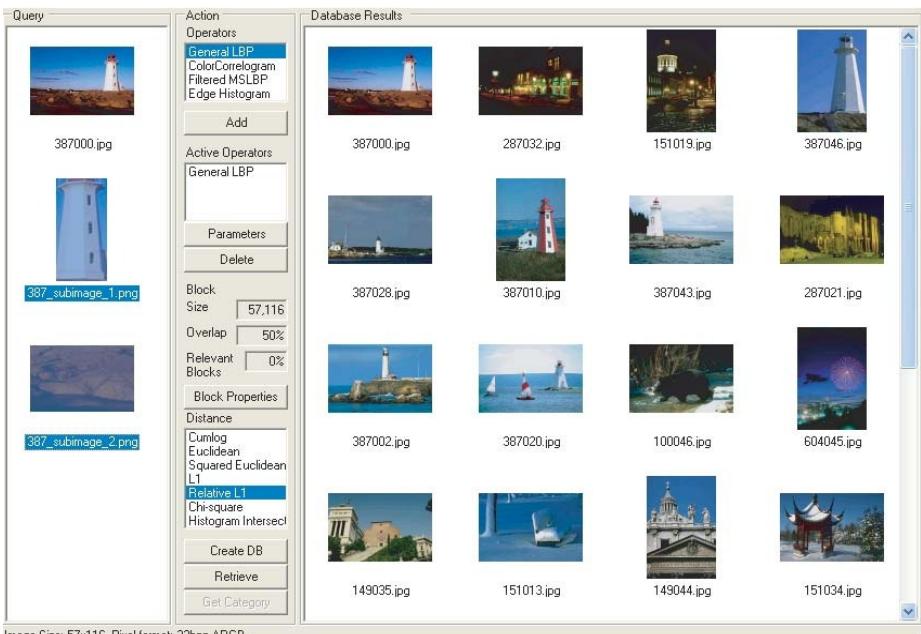
**Table 1.** The results (precision/recall) for different methods

Method	Block size(overlap)	10 images(%)	25 images	50 images
Color Correlogram	full image	37.8/7.5	25.3/12.7	18.7/18.7
Edge Histogram	image dependent	26.3/5.3	18.4/9.2	14.2/14.2
$\text{LBP}_{8,1}^{u2}$	full image	34.7/6.9	24.6/12.3	19.0/19.0
$\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$	full image	36.9/7.5	25.3/12.7	18.7/18.7
$\text{LBP}_{8,1}^{u2}$	$128 \times 128(0 \times 0)$	36.9/7.4	26.5/13.2	21.3/21.3
$\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$	$128 \times 128(0 \times 0)$	37.4/7.5	26.6/13.3	20.4/20.4
$\text{LBP}_{8,1}^{u2}$	$128 \times 128(64 \times 64)$	43.0/8.6	31.3/15.7	23.7/23.7
$\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$	$128 \times 128(64 \times 64)$	43.3/8.7	31.0/15.5	24.0/24.0
$\text{LBP}_{8,1}^{u2}$	$96 \times 96(0 \times 0)$	40.5/8.1	29.0/14.5	22.5/22.5
$\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$	$96 \times 96(0 \times 0)$	41.0/8.2	29.2/14.6	21.5/21.5
$\text{LBP}_{8,1}^{u2}$	$96 \times 96(48 \times 48)$	45.5/9.1	31.6/15.8	24.0/24.0
$\text{LBP}_{8,1}^{u2} +_{8,2,4}^{u2} +_{8,5,4}^{u2}$	$96 \times 96(48 \times 48)$	45.3/9.0	32.5/16.2	23.4/23.4

database), and the numbers are marked as percentages. The LBPs using overlapping blocks achieve precision rates of about 45 % (over 9 % for recall) for 10 images and are clearly better than any of the other extractors. Their results for 50 images are almost as good as those obtained with the best full image operators for 25 images. Using blocks, especially overlapping ones, instead of full images seems to make a clear difference. It is also to be noticed that using overlap has a large impact regardless of the block size. Several percentages are gained through 50 % overlapping.

The test with the primitive block approach was performed with an  $\text{LBP}_{8,1}^{u2}$  operator without interpolation. Fig. 6 shows an example query obtained by using

$32 \times 32$  sized primitive blocks with overlap of two pixels (the overlap between search windows was set to 50 %). The query images have been taken from an original database image and they have been outlined in such a way that they are composed of mostly homogeneous texture. The actual retrieval task was not an easy one: two subimages were needed to get satisfying results, that is seven out of 16 images from the right category (Lighthouses). The chosen subimages have both very distinctive appearance but the image of a rock appeared to have more positive influence on the outcome than the carefully outlined picture of the white lighthouse itself. The probable reason for this is the clear distinction in the properties of the two subimages – the image of the rock is rich in its texture content.



**Fig. 6.** A test query. The left box of images in the user interface is the query group from which only the outlined (darkened) subimages were selected for the eventual search

Some additional tests were also conducted with a stamp database consisting of about 900 German stamp images [22]. A couple of LBP extractors were used and their performance was evaluated against a commercial stamp image retrieval software. The block division method fared at least as well or even better than the matured retrieval application making use of multiple different image features (color, texture, motive, and image size and aspect ratios).

## 5 Conclusions

In this paper, we considered the use of LBP texture features combined with two different block-based image division methods. The results obtained show that the LBP can be successfully used to retrieve images with general content as it is fast to extract and it has useful qualities like invariance to monotonic transitions in gray scale and small descriptor size. The color correlogram, that represents the current state of the art in CBIR, was clearly outperformed by one of the developed subimage approaches.

The increased retrieval rates of the tested methods come at the expense of higher computational demands. The time needed for query grows linearly with the amount of used image blocks. With large images and small block sizes the required processing capacity slips easily out of the grasp of applications that have real-time requirements. Still, it should be noted that it does not seem to be necessary to use large numbers of small blocks as, according to the obtained results, a few blocks per image is usually enough to make a considerable difference when compared to descriptors calculated for full images.

The method based on primitive blocks was hard to assess as there is a level of user interaction involved in the query procedure. Nevertheless, it has some important properties that increase its value in the field of CBIR: It is faster than conventional search window approaches as it does not extract features for every possible search window size separately. Another noteworthy feature is that it can be used to find objects consisting of a single texture or larger entities with several different areas of interest as the query can be adjusted by using more than one sample image.

For the future studies, there are many ways that could enhance the performance and speed of the studied methods. For instance, different block matching algorithms, like the three-step search method, could be used to speed up the matching process. Another possibility could be to use image blocks that are weighted according to their spatial positions. In the case of multiresolution LBP, the use of weights could be extended to emphasize the LBPs containing the most relevant texture information. These and other enhancements could improve the usability of LBP features in the CBIR of the future.

**Acknowledgments.** This study was funded in part by the Academy of Finland.

## References

1. Swain M., Ballard D.: Color Indexing. In: Third International Conference on Computer Vision. (1990) 11–32
2. Huang J., Kumar S. R., Mitra M., Zhu W.-J., Zabih R.: Image Indexing Using Color Correlograms. In: IEEE Conference on Computer Vision and Pattern Recognition (1997) 762–768
3. Stricker M., Orengo M.: Similarity of Color Images In: SPIE Conference on Storage and Retrieval for Image and Video Databases (1995) 381–392

4. Manjunath B. S., Ohm J.-R., Vasudevan V., Yamada A.: Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology* **11** (2001) 703–715
5. Manjunath B. S., Ma W. Y.: Texture Features for Browsing and Retrieval of Image Data. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 837–842
6. Sim D.-G., Kim H.-K., Oh D.-H.: Translation, Rotation, and Scale Invariant Texture Descriptor for Texture-Based Image Retrieval. In: *International Conference on Image Processing* (2000) 742–745
7. Mao J., Jain A.: Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition* **25** (1992) 173–188
8. Francos J. M., Meiri A. Z., Porat B.: On a Wold-Like Decomposition of 2-D Discrete Random Fields. In: *International Conference on Acoustics, Speech, and Signal Processing* (1990) 2695–2698
9. Tamura H., Mori S., Yamawaki T.: Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics* **8** (1978) 460–473
10. Park S. J., Park D. K., Won C. S.: Core Experiments on MPEG-7 Edge Histogram Descriptor ISO/IEC JTC1/SC29/WG11 - MPEG2000/M5984 (2000)
11. Ojala T., Pietikäinen M., Harwood D.: A Comparative Study of Texture Measures with Classification Based on Feature Distribution. *Pattern Recognition* **29** (1996) 51–59
12. Yao C.-H., Chen S.-Y.: Retrieval of Translated, Rotated and Scaled Color Textures. *Pattern Recognition* **36** (2003) 913–929
13. Ahonen T., Hadid A., Pietikäinen M.: Face Recognition with Local Binary Patterns Lecture Notes in Computer Science, Vol. 3021. Springer-Verlag, Berlin Heidelberg New York (2004) 469–481
14. Mäenpää T., Turtinen M., Pietikäinen M.: Real-Time Surface Inspection by Texture. *Real-Time Imaging* **9** (2003) 289–296
15. Pietikäinen M., Nurmela T., Mäenpää T., Turtinen M.: View-Based Recognition of Real-World Textures. *Pattern Recognition* **37** (2004) 313–323
16. Veltkamp R. C., Tanase M.: Content-Based Image Retrieval Systems: A Survey. Revised and extended version of Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University (2002)
17. Laaksonen J., Koskela M., Oja E.: PicSOM: Self-Organizing Maps for Content-Based Image Retrieval. In: *IEEE International Joint Conference on Neural Networks* (1999) 2470–2473
18. Ojala T., Pietikäinen M., Mäenpää T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 971–987
19. Puzicha J., Rubner Y., Tomasi C., Buhmann J. M.: Empirical Evaluation of Dissimilarity Measures for Color and Texture. In: *Seventh IEEE International Conference on Computer Vision* (1999) 1165–1172
20. Corel Corporation. URL: <http://www.corel.com/> (2005)
21. Park D. K., Jeon Y. S., Won C. S.: Efficient Use of Local Edge Histogram Descriptor. In: *2000 ACM workshop on Standards, Interoperability and Practices* (2000) 52–54
22. Takala V.: Local Binary Pattern Method in Context-Based Image Retrieval. Master's thesis, Department of Electrical and Information Engineering, University of Oulu (2004) (in Finnish)

# Enhanced Fourier Shape Descriptor Using Zero-Padding

Iivari Kunttu, Leena Lepistö, and Ari Visa

Tampere University of Technology,

Institute of Signal Processing,

P.O. Box 553, FI-33101 Tampere Finland

{Iivari.Kunttu, Leena.Lepisto, Ari.Visa}@tut.fi

<http://www.tut.fi>

**Abstract.** The shapes occurring in the images are essential features in image classification and retrieval. Due to their compactness and classification accuracy, Fourier-based shape descriptors are popular boundary-based methods for shape description. However, in the case of short boundary functions, the frequency resolution of the Fourier spectrum is low, which yields to inadequate shape description. Therefore, we have applied zero-padding method for the short boundary functions to improve their Fourier-based shape description. In this paper, we show that using this method the Fourier-based shape classification can be significantly improved.

## 1 Introduction

The description of the object shape is an important task in image analysis and pattern recognition. The shapes occurring in the images have also a remarkable significance in image classification and retrieval. The basic problem in shape classification is to define similarity between two shapes. Therefore, different visual features (descriptors) have been developed to characterize the shape content of the images. Common shape description techniques have been reviewed in a recent study of Zhang and Lu [17]. Another review of the state of the art in shape description techniques is provided by Loncaric [9].

The shape description techniques can be divided into two types, boundary based and region based techniques [1]. The region based methods consider the whole area of the object whereas the boundary based shape descriptors use only the object boundary in the shape description. The most popularly used region-based methods are different moments [5],[15]. The best-known boundary based shape descriptors include chain codes [3] and Fourier descriptors [13]. Also autoregressive (AR) [2] models have been used in shape description. Simple shape features such as circularity [1], eccentricity, convexity, principle axis ratio, circular variance and elliptic variance [6] include boundary-based descriptors. Recently, growing research interest has been focused on Curvature Scale Space (CSS) shape representation [11] that has been selected to be used as the boundary-based shape descriptor of MPEG-7 standard.

However, despite the fact that Fourier descriptor is over 30 years old method [4],[13], it is still found to be valid shape description tool. In fact, Fourier descriptor has proved to outperform most other boundary-based methods in terms of classification accuracy and efficiency. This has been verified in several comparisons. Kauppinen et al. [7] made a comparison between autoregressive models and Fourier descriptors in shape classification. In most cases, Fourier descriptors proved to be better in the classification of different shapes. In the comparison made by Mehtre et al. [10], the accuracy of chain codes, Fourier descriptors, and different moments was compared in the shape retrieval. In this case, best results were obtained using moments and Fourier descriptors. In a recent study of Zhang and Lu [16], Fourier descriptors and Zernike moments outperformed CSS representation in terms of retrieval accuracy and efficiency. Similar results were obtained also in [8], in which Fourier descriptors outperformed CSS in defect shape retrieval.

In addition to good classification and retrieval performance, there also other reasons which make Fourier descriptors probably the most popular of the boundary-based shape representations. The main advantages of the Fourier-based shape descriptors are that they are compact and computationally light methods with low dimensionality. Furthermore, they are easy to normalize and their matching is a very simple process. Also their sensitivity to noise is low when only low frequency Fourier coefficients are used as descriptors.

In this paper, the area of Fourier shape descriptors is revisited. We present a method for enhancing the performance of Fourier-based shape description by increasing frequency resolution of the Fourier spectrum calculated for the boundary function of an object shape. Using this method, a more accurate shape representation in frequency domain can be achieved. This is particularly beneficial in the case of objects with relatively short boundary function, in which cases the spectrum estimate has low resolution. The experiments presented in this paper prove that using this technique, the shape classification accuracy can be easily improved.

## 2 Shape Representation Using Fourier Descriptors

In this paper, the shape description methods are based on the boundary line of the object. The boundary can be presented using some shape signature i.e. function derived from the boundary coordinates of the object [1]. Complex coordinate function is a well-known shape signature [7]. It presents the boundary coordinates in an object centered complex coordinate system. Let  $(x_k, y_k)$ ,  $k=0,1,2,\dots,N-1$  represent the boundary coordinates, in which  $N$  is the length of the boundary. The complex coordinate function  $z(k)$  expresses the boundary points in an object centered coordinate system in which  $(x_c, y_c)$  represents the centroid of the object:

$$z(k) = (x_k - x_c) + j(y_k - y_c) \quad (1)$$

Hence, using this function, the boundary is represented independent of the location of the object in the image. In this way the translation invariance can be achieved.

## 2.1 Fourier Description of the Boundary Function

Fourier descriptors characterize the object shape in a frequency domain. The descriptors can be formed for the complex-valued boundary function using the discrete Fourier transform (DFT). The Fourier transform of a boundary function generates a set of complex numbers, which characterize the shape in frequency domain. Fourier transform of  $z(k)$  is:

$$F(n) = \frac{1}{N} \sum_{k=0}^{N-1} z(k) e^{-j2\pi k / N} \quad (2)$$

for  $n=0,1,2,\dots,N-1$ . The transform coefficients  $F(n)$  form the Fourier spectrum of the boundary function. The translational invariance of this shape representation is based on the object centered shape signature. Furthermore, the coefficients have also to be normalized to achieve invariance to rotation and scaling. The descriptors can be made rotation invariant by ignoring the phase information and using only the magnitudes of the transform coefficients  $|F(n)|$ . In the case of complex-valued boundary function, the scale can be normalized by dividing the magnitudes of the transform coefficients by  $|F(1)|$  [7].

## 2.2 Zero-Padding Method

Even if Fourier descriptor is a powerful tool of boundary-based shape description, its performance is somewhat dependent on the frequency resolution of the Fourier spectrum. When the boundary function  $z(k)$  is Fourier transformed, the resulting Fourier spectrum is of the same length as boundary function. Therefore, in the case of short boundary functions the frequency resolution is also low. To obtain better resolution, the number of the datapoints in the boundary function should be increased. In practice, this is not always feasible, because the boundary lines of the objects are usually defined pixel-by-pixel, and the number of the boundary points depends on the image resolution. However, there is an alternative approach for this purpose. Zero-padding [12] is a commonly used method in signal processing. It can be used to increase the frequency resolution by adding zeros to the function to be Fourier transformed. Hence, a new function is defined as:

$$z_{zp}(k) = \begin{cases} z(k) & \text{for } 0 \leq k \leq N-1 \\ 0 & \text{for } N \leq k \leq N_{zp} - 1 \end{cases} \quad (3)$$

in which  $N_{zp}$  is the length of a desired frequency spectrum. By using additional zeros in the input signal of DFT, new spectrum values are being interpolated among the original values in the spectrum. This way, the density of the frequency samples is increased in the spectrum. In practice, the desired spectrum length is selected such that  $N_{zp}=2^p$  in which  $p$  is a positive integer. This is beneficial because DFT is usually

implemented using FFT algorithm, in which input functions of length  $2^p$  are preferred to decrease computing time.

### 2.3 Descriptors

The Fourier spectrum represents the frequency content of the boundary function. General shape of the object is represented by the low frequency coefficients of  $F(n)$ , whereas high frequency coefficients represent the fine details in the object shape. A common approach to shape representation is the use of a subset of low-frequency coefficients as a shape descriptor. Consequently the shape can be effectively represented using a relatively short feature vector. In our experiments, the feature vector is formed using *Contour Fourier* method [7], which applies the complex coordinate function. In the case of complex valued boundary functions, the coefficients are taken both positive and negative frequency axis. The feature vector is formed as:

$$x = \left[ \frac{|F_{-(L/2-1)}|}{|F_1|} \dots \frac{|F_{-1}|}{|F_1|} \frac{|F_2|}{|F_1|} \dots \frac{|F_{L/2}|}{|F_1|} \right]^T \quad (4)$$

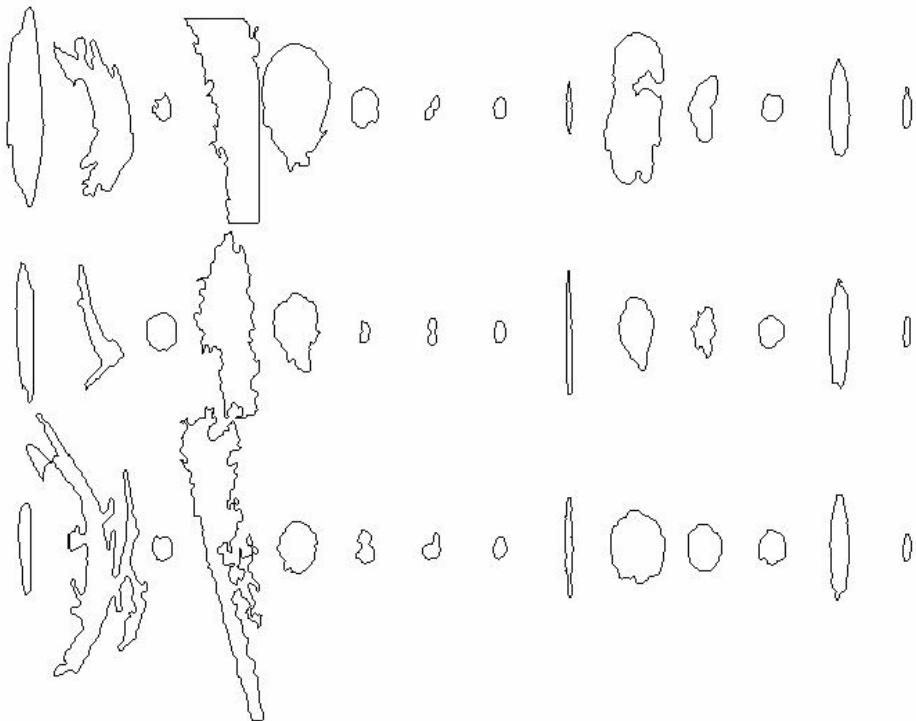
where  $L$  is a constant value that defines the feature vector length (dimensionality).

## 3 Experiments

In this paper, we make experiments to demonstrate the effect of the zero-padding on the shape classification performance. In this experimental part we use a database of industrial defect shapes as a test set.

### 3.1 Testing Database

For testing purposes, we used defect images that were collected from a real industrial process using a paper inspection system [14]. A reason for collecting defect image databases in process industry is a practical need for controlling the quality of production [14]. When retrieving images from a database, the defect shape is one essential property describing the defect class. Therefore, effective methods for the shape representation are necessary. The defects occurring in paper can be for example holes, wrinkles or different kinds of thin or dirt spots. The test set consisted of 1204 paper defects which represented 14 defect classes with each class consisting of 27-103 images. Three example contours of each defect class are presented in figure 1. Within each class, there are defects of different size and orientation. Furthermore, in some classes the boundaries are very varying and sometimes distorted (classes 2, 4 and 10, for example).

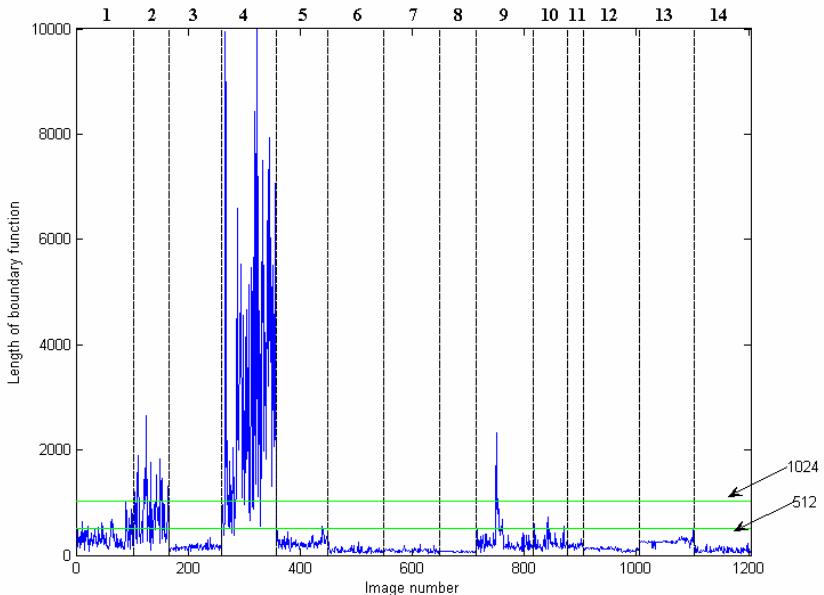


**Fig. 1.** Three example contours of each 14 paper defect class in the testing database

### 3.2 Classification

The feature vectors for the shapes in the database were calculated for ordinary *Contour Fourier* descriptors as well as for the same descriptors using zero-padding. The lengths of the boundary lines of the defect shapes were very varying, which can be seen in figure 2. In this figure, the lengths of each 1204 defect boundaries are presented. In our experiments, the boundaries of lower lengths than  $N_{zp}$  were inserted with zeros. Two values of  $N_{zp}$  were used, namely 512 and 1024. However, preliminary experiments have showed that in some cases zero-padding also decreases the Fourier-based shape distinction. This can be avoided by emphasizing the zero-padding only to the shortest boundaries in the test set. Therefore, we made an additional experiment, in which only very short boundaries whose length was less than 100 points, were used. These boundaries were zero-padded quite strongly, to 1024 points. The length of the Fourier descriptor ( $L$ ) was selected to be 16 in all the experiments. It is important to note that the use of zero-padding has no influence on the descriptor dimensionality.

In classification, we used 5-nearest neighbor classifier and leave-one-out validation. The distance metrics was selected to be Euclidean distance, which is a standard



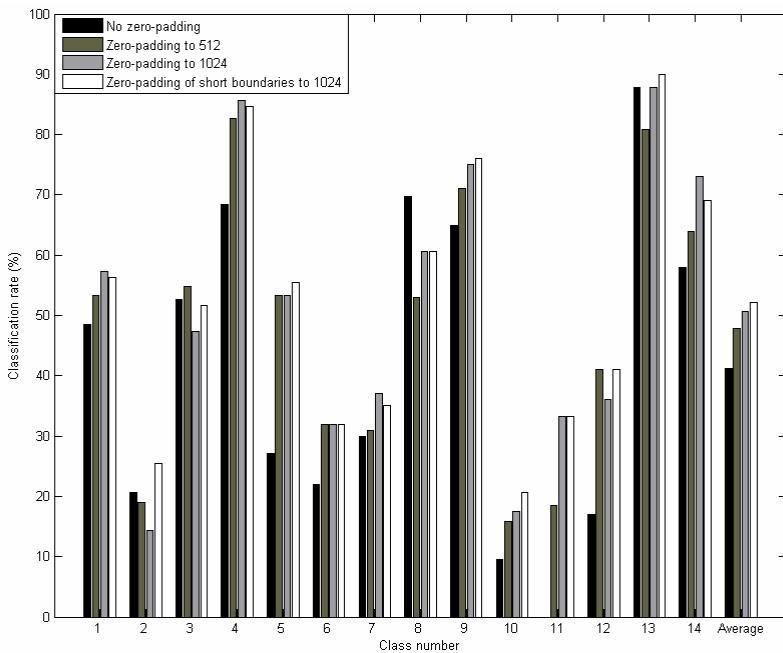
**Fig. 2.** The lengths of the boundary functions of 1204 paper defects in the testing database. The numbers above the plot correspond to the classes

approach with Fourier descriptors. We carried out the classification using four selected methods, which were ordinary *Contour Fourier*, *Contour Fourier* with zero-padding to 512 or 1024 points, and *Contour Fourier* with zero padding of very short boundaries to 1024 points. Average classification rates of the descriptors are presented for each 14 defect class in figure 3.

### 3.3 Results

The results presented in figure 3 show that the zero-padding method is able to improve the shape classification rates in most of the defect classes. Particularly, in the classes containing short boundaries the overall result was improved. Therefore, the application of the zero-padding was capable of increasing the average classification performance from 41.2 % to 52.2 %. The best average result was achieved using the adaptive zero-padding of very short boundaries to 1024 datapoints. However, all the zero-padding approaches were able to improve the classification results, especially in the classes with short boundaries. For example, in class 11 the classification rate was improved from zero to over 30 %.

According to the obtained results, it seems that the most recommendable approach is to use the zero-padding method only to the shortest boundaries, though the other presented approaches produce clear improvement in classification performance as well.



**Fig. 3.** The average results of 5-NN classification in each 14 defect class

## 4 Discussion

In this paper, a method for improving Fourier-based shape description was presented. The proposed method is a simple and fast tool for improving the shape distinction ability of Fourier descriptors. It is particularly beneficial in the case of short boundary functions, which do not produce adequate frequency resolution to their Fourier spectrum. The zero-padding method as itself is not a novel method, because it has been applied to different kinds of signal processing problems before. However, the new way in which it has been employed to improve shape description has a certain practical value in different shape classification problems.

For experimental purposes, we used industrial defect images, which are quite complicated shape classification task. This is due to the irregularities and variations in the defect shapes. Some of the classes are also somewhat overlapping. It is essential to note that in real defect image classification the shape is not the only classifying feature, because also texture and gray level distribution play a role in defect image description.

The experimental results obtained from this real-world shape classification problem show that using zero-padding the classification performance can be significantly improved. This, on the other hand, does not increase the computational cost, because the dimensionality of the feature vectors remains the same. Zero-padding method

does not require additional computational capacity in feature extraction, either. This is due to the advanced FFT algorithms [12].

In conclusion, the zero-padding method has proved to be an effective tool for enhancing Fourier-based shape description. It is especially effective with short boundary functions of complicated shapes.

## Acknowledgment

The authors wish to thank ABB Oy (Mr. Juhani Rauhamaa) for the paper defect image database used in the experiments.

## References

1. Costa, L.F., Cesar, R.M.: *Shape Analysis and Classification, Theory and Practice*, CRC Press, Boca Raton, Florida (2001).
2. Dubois, S.R., Glanz, F.H.: An Autoregressive Model Approach to Two-Dimensional Shape Classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 8 (1986) 55-66
3. Freeman, H., Davis, L.S.: A Corner Finding Algorithm for Chain Coded Curves, *IEEE Transactions on Computers* Vol. 26 (1977) 297-303
4. Granlund, G.H.: Fourier Preprocessing for Hand Print Character Recognition, *IEEE Transactions on Computers* Vol. C-21, No. 2 (1972) 195-201
5. Hu, M.K.: Visual Pattern Recognition by Moment Invariants, *IRE Transactions on Information Theory* Vol. IT-8 (1962) 179-187
6. Iivarien, J., Visa, A.: An adaptive texture and shape based defect classification, *Proceedings of the 14<sup>th</sup> International Conference on Pattern Recognition*, Vol. 1, (1998) 117-122
7. Kauppinen, H., Seppänen, T., Pietikäinen, M.: An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2 (1995) 201-207
8. Kunttu, I., Lepistö, L., Rauhamaa, J., Visa, A.: Multiscale Fourier Descriptor for Shape-Based Image Retrieval, *Proceedings of 17<sup>th</sup> International Conference on Pattern Recognition*, Vol. 2 (2004) 765-768
9. Loncaric, S.: A survey of shape analysis techniques, *Pattern Recognition* Vol. 31, No. 8 (1998) 983-1001
10. Mehtre, B.M., Kankanhalli, M.S., Lee, W.F.: Shape Measures for Content Based Image Retrieval: A Comparison, *Information Processing Management*, Vol. 33, No 3, (1997) 319-337
11. Mokhtarian, F., Mackworth, A.K.: Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1 (1986) 34-43
12. Orfanidas, S.J.: *Introduction to Signal Processing*, Prentice Hall (1996)
13. Persoon, E., Fu, K.: Shape Discrimination Using Fourier Descriptors, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 7 (1977) 170-179
14. Rauhamaa, J., Reinius, R.: Paper Web Imaging with Advanced Defect Classification, *Proceedings of the 2002 TAPPI Technology Summit* (2002)

15. Teague, M.R.: Image Analysis via the General Theory of Moments, *Journal of Optical Society of America*, Vol. 70, No. 8 (1980) 920-930
16. Zhang, D., Lu, G.: A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-Based image Retrieval, *Journal of Visual Communication and Image Representation*, Vol. 14, No. 1 (2003) 41-60
17. Zhang, D., Lu, G.: Review of shape representation and description techniques. *Pattern Recognition*, Vol. 37, No. 1, (2004) 1-19

# Color-Based Classification of Natural Rock Images Using Classifier Combinations

Leena Lepistö, Iivari Kunttu, and Ari Visa

Tampere University of Technology, Institute of Signal Processing,  
P.O. Box 553, FI-33101 Tampere, Finland  
[{Leena.Lepisto, Iivari.Kunttu, Ari.Visa}@tut.fi](mailto:{Leena.Lepisto, Iivari.Kunttu, Ari.Visa}@tut.fi)  
<http://www.tut.fi/>

**Abstract.** Color is an essential feature that describes the image content and therefore colors occurring in the images should be effectively characterized in image classification. The selection of the number of the quantization levels is an important matter in the color description. On the other hand, when color representations using different quantization levels are combined, more accurate multilevel color description can be achieved. In this paper, we present a novel approach to multilevel color description of natural rock images. The description is obtained by combining separate base classifiers that use image histograms at different quantization levels as their inputs. The base classifiers are combined using classification probability vector (CPV) method that has proved to be an accurate way of combining classifiers in image classification.

## 1 Introduction

Image classification is an essential task in the field of image analysis. The classification is usually based on a set of visual features extracted from the images. These features may characterize for example colors or textures occurring in the images. Most of the real-world images are seldom homogenous. Especially, different kinds of natural images have often non-homogenous content. The division of natural images like rock, stone, clouds, ice, or vegetation into classes based on their visual similarity is a common task in many machine vision and image analysis solutions. In addition to non-homogeneities, the feature patterns can be also noisy and overlapping. Due to these reasons, different classifiers may classify the same image in different ways. Hence, there are differences in the decision surfaces, which lead to variations in classification accuracy. However, it has been found that a consensus decision of several classifiers can give better accuracy than any single classifier [1],[8],[9]. This fact can be easily utilized in the classification of real-world images.

The goal of combining classifiers is to form a consensus decision based on opinions provided by different base classifiers. Duin [5] presented six ways, in which consistent set of base classifiers can be generated. In the base classifiers, there can be differences in initializations, parameter choices, architectures, classification principle, training sets, or feature sets. Combined classifiers have been applied to several classification tasks, for example to face recognition [15], person identification [3] and fingerprint verification [7]. Theoretical framework for combining classifiers is provided in [8].

In the image classification, several types of classifier combination approaches can be used. In our previous work [11],[13] we have found that different feature types can be easily and effectively combined using classifier combinations. In practice, this is carried out by making the base classification for each feature type separately. The final classification can be obtained based on the combination of separate base classification results. This has proved to be particularly beneficial in the case of non-homogenous natural images [11],[13]. Hence, the non-homogenous properties of individual features do not necessarily affect directly on the final classification. In this way, each feature has its own affect on the classification result.

Rock represents typical example of non-homogenous natural image type. This is because there are often strong differences in directionality, granularity, or color of the rock texture, even if the images represented the same rock type [11]. Moreover, rock texture is often strongly scale-dependent. Different spatial multiscale representations of rock have been used as classification features using Gabor filtering [12]. However, the scale dependence of the rock images can be used also in another way, using color quantization. It has been found that different color features can be found from the rock images using different numbers of quantization levels. Hence, by combining the color representation at several levels, a multilevel color representation can be achieved. For this kind of combination, a classifier combination method can be used. In this paper, we present our method to make a classifier combination that is used to produce this multilevel color representation. The rest of this paper is organized as follows. Section two presents the main principle of classifier combinations as well as our method for that purpose. In section three, the principle of multilevel color representation is presented. The classification experiments with natural rock images are presented in section four. The obtained results are discussed in section five.

## 2 Classifier Combinations in Image Classification

The idea of combining classifiers is that instead of using single decision making theme, classification can be made by combining opinions of separate classifiers to derive a consensus decision [8]. This can increase classification efficiency and accuracy. In this section, methods for combining separate classifiers are presented. Furthermore, we present our approach to make a probability-based classifier combination.

### 2.1 Methods for Combining Classifiers

The general methods for combining classifiers can be roughly divided into two categories, voting-based methods and the methods based on probabilities. The voting-based techniques are popularly used in pattern recognition [10],[14]. In the voting-based classifier combinations, the base classifier outputs vote for the final class of an unknown sample. These methods do not require any additional information, like probabilities from the base classifiers. Voting has proved to be a simple and effective method for combining classifiers in several classification problems. Also in the comparisons with the methods presented by Kittler et al., voting-based methods have given relatively accurate classification results [8]. Lepistö et al. [11] presented a method for combining

classifiers using classification result vectors (CRV). In this approach, the class labels provided by the base classifiers are used as a feature vector in the final classification, and hence the result is not based on direct voting. CRV method outperformed voting method in the classification experiments. In [13] an unsupervised variation of CRV was presented and compared to other classifier combinations.

Recently, the probability-based classifier combination strategies have been popularly used in pattern recognition. In these techniques, the final classification is based on the a posteriori probabilities of the base classifiers. Kittler et al. [8] presented several common strategies for combining base classifiers. These strategies are e.g. product rule, sum rule, max rule, min rule, and median rule. All these rules are based on the statistics computed based on the probability distributions provided by the base classifiers. In [8], the best experimental results have been obtained using sum and median rules. Theoretical comparison of the rules has been carried out in [9]. Also Alkoot and Kittler [1] have compared the classifier combination strategies.

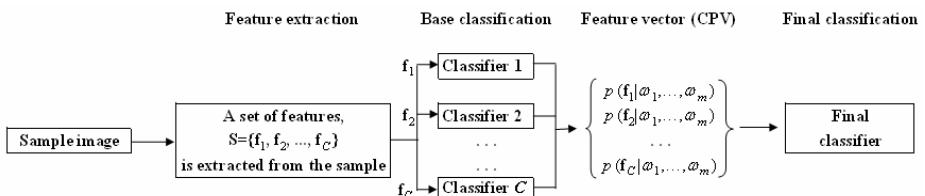
## 2.2 Classification Probability Vector Method

Our previous method for combining classifiers combined the outputs of the base classifiers into a feature vector that is called classification result vector (CRV) [11]. However, CRV method uses only the class labels provided by the base classifiers, and ignores their probabilities. In this paper, we use the probability distributions of the separate base classifiers as features in the final classification.

In general, in the classification problem a pattern  $S$  is to be assigned to one of the  $m$  classes  $(\omega_1, \dots, \omega_m)$  [8]. We assume that we have  $C$  classifiers each representing a particular feature type and we denote the feature vector used by each classifier by  $f_i$ . Then each class  $\omega_k$  is modeled by the probability density function  $p(f_i | \omega_k)$ . The priori probability of occurrence of each class is denoted  $P(\omega_k)$ . The well-known Bayesian decision theory [4],[8] defines that  $S$  is assigned to class  $\omega_j$  if the a posteriori probability of that class is maximum. Hence,  $S$  is assigned to class  $\omega_j$  if:

$$P(\omega_j | f_1, \dots, f_C) = \max_k P(\omega_k | f_1, \dots, f_C) \quad (1)$$

However, the probabilities of all the other classes than  $\omega_j$  have also significance in classification. They are particularly interesting when the pattern  $S$  is located near the decision surface. Therefore, we focus on the whole probability distribution  $p(f_i | \omega_1, \dots, \omega_m)$  provided by each classifier. Hence, if the probability is defined for each  $m$  class, the obtained probability distribution is a  $C$  by  $m$  matrix for each pattern  $S$ . This matrix is used as a feature vector in the final classification, and it is called



**Fig. 1.** The outline of the CPV classifier combination method

classification probability vector (CPV). In the final classification, the images with similar CPV's are assigned into same classes. The outline of the CPV method is presented in figure 1.

The common probability-based classifier combinations [8] are used to calculate some statistics based on the probability distributions provided by the base classifiers. In contrary to them, CPV method uses the whole probability distribution as a feature vector in the final classification. The CPV method utilizes the fact that the separate base classifiers classify similar samples in the similar way, which leads to a similar probability profile. The final classification is based merely on the similarity between the probabilities of the base classifiers. Hence, in contrary to voting, in the CPV method the base classifier outputs (class labels) do not directly affect the final classification result.

When image classification is considered, the CPV method has several advantages. CPV method considers each visual feature of the images in the base classifiers separately. In the final classification, the probability distributions are employed instead of features. This way, the individual features do not directly affect the final classification result. Therefore, classification result is not sensitive to variations and non-homogeneities of single images.

### **3 Multilevel Color Representation Using Quantization**

#### **3.1 Color Image Representation**

In digital image representation, an image has to be digitized in two manners, spatially (sampling) and in amplitude (quantization) [6]. The use of spatial resolutions is common in different texture analysis and classification approaches, whereas the effect of quantization is related to the use of image color information. The quantization can be applied to different channels of a color image. Instead of using common RGB color space, the use of HSI space has found to be effective, because it corresponds to the human visual system [16].

A common way of expressing the color content of an image is the use of image histogram. Histogram is a first-order statistical measure that expresses the color distribution of the image. The length of the histogram vector is equal to the number of the quantization levels. Hence the histogram is a practical tool for describing the color content at each level. Histogram is also a popular descriptor in color-based image classification, in which images are divided into categories based on their color content.

#### **3.2 Multilevel Classification**

The classifier combination tools presented in section two provide a straightforward tool for making a histogram-based image classification at multiple levels. Hence the histograms at selected quantization levels and color channels are used as separate input features. Each feature is then classified separately at base classification. After that, the base classification results are combined to form the final classification. This way the final classifier uses multilevel color representation as classifying feature.



**Fig. 2.** Three example images from each rock type in the testing database

## 4 Experiments

In this section, we present the classification experiments using rock images. The purpose of the experiments is to show that an accurate multilevel color representation is achievable using classifier combinations. We also compare the CPV method to other classifier combination approaches.

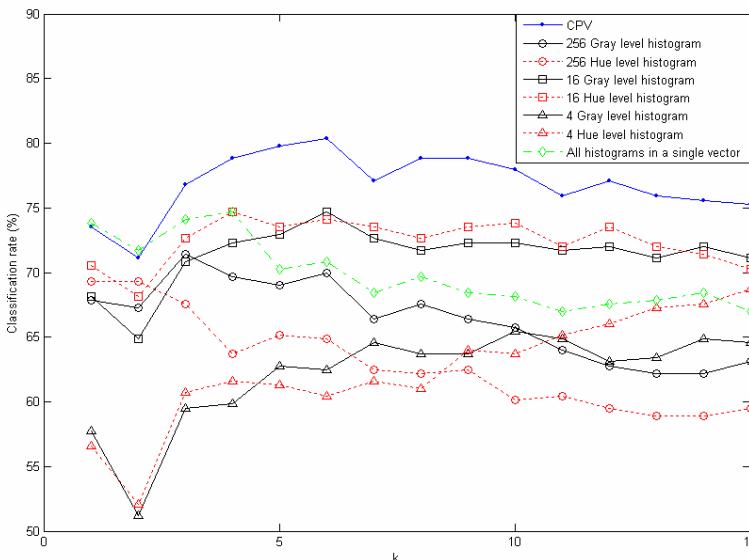
### 4.1 Rock Images

The experiments in this paper are focused on non-homogenous natural image data that is represented by a database of rock images. There is a practical need for methods for classification of rock images, because nowadays rock and stone industry uses digital imaging for rock analysis. Using image analysis tools, visually similar rock texture images can be classified. Another application field for rock imaging is geological research work, in which the rock properties are inspected using borehole imaging. Different rock layers can be recognized and classified from the rock images based on e.g. the color and texture properties of rock. The degree of non-homogeneity in rock is typically overwhelming and therefore, there is a need for an automatic classifier that is capable of classifying the borehole images into visually similar classes. The testing database consists of 336 images that are obtained by dividing large borehole images into parts. These images are manually divided into four classes by an expert. In classes 1-4, there are 46, 76, 100, and 114 images in each class, respectively. Figure 2 presents three example images of each four class.

## 4.2 Classification Experiments

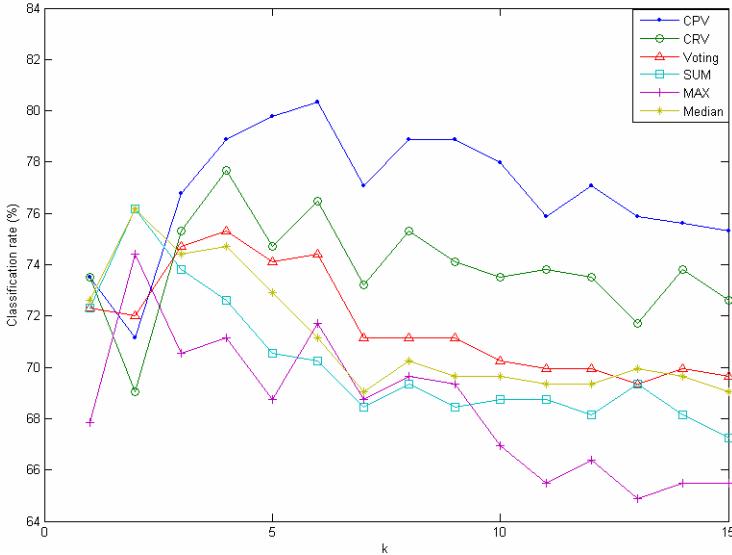
In the experimental part, the principle for classification was selected to be  $k$ -nearest neighbor ( $k$ -NN) method in base classification and final classification. Barandela et al. [2] have proved that nearest neighbor principle is efficient and accurate method to be used in classifier combinations. Classification results are obtained using leave one out validation principle [4]. The distance metrics for the comparison of histograms in the base classification was selected to be  $L_1$  norm. In the CPV method, the final classifier used  $L_2$  norm (Euclidean distance) to compare CPV:s.

The histograms were calculated for the database images in HSI color space. In the classification experiments we used hue (H) and intensity (I) channels, which have been proved to be effective in color description of rock images [12]. The hue and intensity histograms were calculated for the images quantized to 4, 16, and 256 levels. Hence the number of the features was six. The classification was carried out using values of  $k$  varying between 1 and 15. In the first experiment, the classification rate of CPV method was compared to that of separate base classifiers which use different histograms. In this comparison, also the classification accuracy all the histograms combined into a single feature vector was tested. The average classification rates are presented in figure 3 as a function of  $k$ . The second experiment measured the classification accuracy of different classifier combination strategies compared to CPV method. The idea of this experiment was to combine the six base classifiers that use different histograms as input features. In this case, CPV was compared to the most usual probability-based classifier combinations, sum, max and median rules [8]. Product rule is not included into comparison, because the probability estimates of



**Fig. 3.** The average classification rates of the rock images using base classifiers that use different histograms and the classifier combination (CPV)

$k$ -NN classifiers are sometimes zero, which may corrupt the result. In addition to the selected probability-based combination methods, we used also majority voting and our previously introduced CRV method [11] in the comparison. Figure 4 presents the results of this comparison with  $k$  varying between 1 and 15.



**Fig. 4.** The average classification rates of the rock images using different classifier combinations

#### 4.3 Results

The results presented in figure 3 show that using CPV the classification accuracy is clearly higher than that of any single base classifier. The performance of CPV is also compared to the alternative approach, in which all the histograms are collected into a single feature vector. This vector is very high dimensional (552 dimensions), and its performance is significantly lower than in the case of CPV. This observation gives the reason for the use of classifier combinations in the image classification. That is, different features can be combined by combining their separate base classifiers rather than combining all the features into a single high dimensional feature vector in classification. This way, also the “curse of dimensionality” can be avoided.

The results of the second experiment presented in figure 4 show that CPV method outperforms the other classifier combinations in the comparison with a set of rock images. Also the CRV [11] method gives relatively good classification performance. Only with small values of  $k$  CPV is not accurate one. This is due to the probability distributions used by CPV are able to effectively distinguish between image classes only when more than three nearest neighbors are considered in  $k$ -NN algorithm.

## 5 Discussion

In this paper, we presented a method for combining classifiers in the classification of real rock images. Due to their non-homogenous nature, the classification of them is a difficult task. We presented a method for an effective multilevel color representation using our classifier combination strategy, classification probability vector (CPV). In CPV method, the feature vector that describes the image content is formed using the probability distributions of separate base classifiers. The probabilities provided by the base classifiers form a new feature space, in which the final classification is made. Hence the final classification depends on the metadata of the base classification, not the image features directly. This way the non-homogeneities of individual features do not have direct impact on the final result.

In the color-based image classification, like in image classification in general, it is often beneficial to combine different visual features to obtain the best possible classification result. Therefore, classifiers that use separate feature sets can be combined. In this study this feature combination approach was applied to color histograms with different numbers of bins. By combining the histograms using classifier combinations, a multilevel color representation was achieved. The experimental results showed that this representation outperforms any single histogram in classification. Furthermore, CPV method also gives better classification accuracy than any other classifier combination in the comparison.

## Acknowledgement

The authors wish to thank Saanio & Riekola Oy for the rock image database used in the experiments.

## References

1. Alkoot, F.M., Kittler, J.: Experimental evaluation of expert fusion strategies, *Pattern Recognition Letters*, Vol. 20 (1999) 1361-1369
2. Barandela, R., Sánchez, J.S., Valdovinos, R.M.: New applications of ensembles of classifiers, *Pattern Analysis & Applications*, Vol. 6 (2003) 245-256
3. Brunelli, R., Falavigna, D.: Person Identification Using Multiple Cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17 (1995) 955-966
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2<sup>nd</sup> ed., John Wiley & Sons, New York (2001)
5. Duin, R.P.W.: The Combining Classifier: to Train or Not to Train, In: *Proceedings of 16<sup>th</sup> International Conference on Pattern Recognition*, Vol. 2 (2002) 765-770
6. Gonzales, R.C., Woods, R.E.: *Digital Image Processing*, Addison Wesley, 1993.
7. Jain, A.K., Prabhakar, S., Chen, S.: Combining Multiple Matchers for a High Security Fingerprint Verification System, *Pattern Recognition Letters*, Vol. 20 (1999) 1371-1379
8. Kittler, J., Hatef, M., Duin, R.P.W., Matas J.: On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20 (1998) 226-239
9. Kuncheva, L.I.: A Theoretical Study on Six Classifier Fusion Strategies, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 (2002) 281-286

10. Lam, L., Suen, C.Y.: Application of majority voting to pattern recognition: An analysis of the behavior and performance, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 27 (1997) 553-567
11. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Classification of Non-homogenous Textures by Combining Classifiers, *Proceedings of IEEE International Conference on Image Processing*, Vol. 1 (2003) 981-984
12. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Classification Method for Colored Natural Textures Using Gabor Filtering, In: *Proceedings of 12<sup>th</sup> International Conference on Image Analysis and Processing* (2003) 397-401
13. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Combining Classifiers in Rock Image Classification – Supervised and Unsupervised Approach, In: *Proceedings of Advanced Concepts for Intelligent Vision Systems*, (2004) 17-22
14. Lin, X., Yacoub, S., Burns, J., Simske, S.: Performance analysis of pattern classifier combination by plurality voting, *Pattern Recognition Letters*, Vol. 24 (2003) 1959-1969
15. Lu, X., Wang, Y., Jain, A.K.: Combining Classifiers for Face Recognition, In: *Proceedings of International Conference on Multimedia and Expo*, Vol. 3 (2003) 13-16
16. Wyszecki, G., Stiles, W.S.: *Color Science, Concepts and Methods, Quantitative Data and Formulae*, 2<sup>nd</sup> Edition, John Wiley & Sons (1982)

# Fast Guaranteed Polygonal Approximations of Closed Digital Curves

Fabien Feschet

LLAIC1 Laboratory

IUT Clermont-Ferrand, Campus des Cézeaux 63172 Aubière Cedex, France  
[feschet@llaic3.u-clermont1.fr](mailto:feschet@llaic3.u-clermont1.fr)

**Abstract.** We present in this paper a new non-parametric method for polygonal approximations of digital curves. In classical polygonal approximation algorithms, a starting point is randomly chosen on the curve and heuristics are used to ensure its effectiveness. We propose to use a new canonical representation of digital curves where no point is privileged. We restrict the class of approximation polygons to the class of digital polygonizations of the curve. We describe the first algorithm which computes the polygon with minimal Integral Summed Squared Error in the class in both linear time and space, which is optimal, independently of any starting point.

## 1 Introduction

Approximation of digital curves or shapes is an important task in pattern recognition, digital cartography, data compression... Polygonal models are still easier to handle than digital curves. The problem is usually defined by an error criterion and the constraint that the vertices of the polygonal model are points of the original digital curve. In the present work, we only use the Integral Summed Squared Error (ISSE) criterion. We refer to the thesis of Marji [Mar03] which contains more than 100 algorithms, methods and measures. Polygonal approximations are closely related to dominant [TC89] and corner [MS04] point detection. The polygonal approximation problem consists in finding the real polygon such that the ISSE with the digital curve is minimized. This problem is ill-posed since the polygon obtained by linking every point of the digital curve clearly reaches the minimal value of 0. This fact leads to the  $\min - \varepsilon$  problem which corresponds to the minimization of the ISSE for polygons with a fixed number of edges, or to the non-parametric algorithms where the number of edges is automatically defined. Two classes of algorithms have been proposed: graph-theoretical [II88] and optimization (dynamic programming) [PV94]. Best complexities are  $O(n^2 \log n)$  for the former and closed to  $O(n^2)$  time and  $O(Mn)$  space where  $M$  is the fixed number of edges [Sal01, Sal02] for the latter and  $n$  is the number of points. Beside this, suboptimal methods have also been proposed [DP73, RR92, MS04, KF04]. They have the advantage of having low complexity, but they lack guarantee for high quality results.

In almost all previous works, digital curves were considered opened and so for closed curves an initial starting point was randomly chosen. Heuristics have been proposed to overcome the problem [PV94, HL02]. We propose in this paper a non-parametric approach to the problem by combining graph-theoretic and optimization approaches in order to solve the initial point problem. The result is an algorithm which does not depend on any point of the digital curve. Our algorithm is based on digital lines - partially used by Cornic [Cor97] -, a circular arc-graph canonical representation of digital curves [Fes05] and efficient error computation following the ideas in [HL02]. The resulting algorithm is linear time and linear space and is thus optimal. Starting with digital lines, we construct a graph representation. To obtain an efficient algorithm, we restrict the class of allowed polygonal models to the class of digital polygonalizations. We solve the polygonal approximation problem using the graph structure. Our strategy does not suffer from the tendency of the ISSE to favor polygonal model with a high number of edges since it is known [FT03] that there exists only two lengths differing by only one for the whole set of polygonalizations of any digital curve.

In Section 2, we present the notion of digital lines and the graph representation of digital curves known as the tangential cover. The method is presented in Section 3 and experimental validation in Section 4. Final conclusions and extensions end the paper.

## 2 Representation of Closed Digital Curves

A closed digital curve  $C$  is a list of connected points  $p_i = (x_i, y_i)$  of  $\mathbb{Z}^2$ . The allowed connectivities are the standard four and eight connectivities corresponding to the points with  $L_1$  and respectively  $L_\infty$  distance of one of a given point. Self-intersections are allowed as soon as points are duplicated in the list. For closed digital shapes, any boundary tracking algorithm produces the required list and the choice of the point  $p_0$  is arbitrary. If  $C = (p_i)_{0 \leq i < n}$ , we call  $n$  the length of the curve. The curve is closed if  $p_n = p_0$ . The order given by the usual order of the indices defines the orientation of the curve. In the paper, left and right as well as next and previous are defined relatively to the orientation of  $C$ . Indices are intended modulo  $n$ . A polygonal approximation of the curve  $C$  is a real polygon whose vertices belongs to  $C$ . It has the same orientation than the curve  $C$ . The goal of this section is to construct a geometrical representation suitable for computing polygonal approximations.

### 2.1 Digital Lines

Our representation has been introduced in [FT99], used in [FT03] and fully exploited in [Fes05]. It is based on digital lines (we refer to Rosenfeld and Klette [RK04] for definitions and main properties of digital lines). A digital segment is a finite connected subset of a digital line. Given a list of points extracted from a digital curve  $C$ , there exist algorithms to incrementally recognize if the list is a digital segment or not. When the answer is positive, the algorithms output the

parameters of a digital line containing the digital segment. The points can be taken both in positive or negative orientation of the curve  $C$ . The complexity of the recognition is linear in the number of points of the list. Given a subset of  $C$  which is a digital segment, it is called maximal if and only if it cannot be extended over  $C$  to another digital segment. Maximality of digital segments can be checked during their recognition within the same complexity bound.

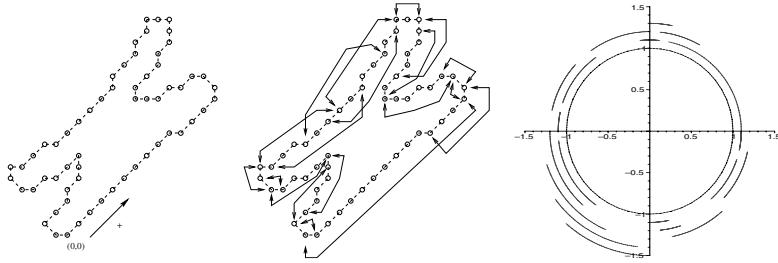
## 2.2 Tangential Cover

We now briefly present the construction given in [Fes05]. It is based on the notion of discrete (or digital) tangent at a point  $p_i$  of  $C$ . A digital tangent is a subset  $(p_{i-l}, \dots, p_i, \dots, p_{i+r})$  which is a maximal segment. The construction is as follows: points are alternatively added to the right and to the left of  $p_i$  while the resulting subset is a digital segment. When one side cannot be further extended, the addition of points are pursued at the other side while the subset is a digital segment. The resulting set forms the digital tangent of  $C$  at  $p_i$  and is as symmetric as possible. However, symmetry is not imposed and  $l$  and  $r$  are usually different. Any digital tangent is a maximal digital segment. We denote by  $T_{p_i}$  the digital tangent at  $p_i$ .

As explained in [Fes05], it often happens that  $T_{p_i} = T_{p_{i+1}}$ . This property was exploited in [FT99, Fes05] to built an efficient algorithm to construct the set of all digital tangents  $\mathcal{T}(C) = \{T_{p_i}, p_i \in C\}$ . The set  $\mathcal{T}(C)$  is called the tangential cover of  $C$ . The complexity of its construction is proved to be  $O(n)$ . The tangential cover has the property that it contains all maximal digital segments which can be constructed on the curve  $C$  with connected subsets.

Each digital tangent can be represented by its left and right ending points  $p_{i-l}$  and  $p_{i+r}$  and by the parameters of the associated digital segment. It is straightforward to see that the series of  $l$  and  $r$  are increasing series and that two consecutive tangents must overlap. Moreover due to the maximality of the discrete segments, no tangent can be contained in another one. To represent the tangential cover, we use the concept of circular arc graphs [Sch03]. A digital tangent can be represented by an interval  $[i - l, i + r]$ . We associate to it the arc between the angles  $2(i - l)\pi/n$  and  $2(i + r)\pi/n$ . We obtain a circular arc graph (see Fig. 1, right). We have increased or decreased the radius of each arc to distinguish between overlapping arcs. It must be said that each point can be viewed as a radial line from the center of the representation to one point of the boundary of the central unit circle. All the intersected arcs by this line correspond to the digital tangents containing the point. The horizontal axis pointing to the right corresponds to the points  $(0, 0)$  of the left image. It is clear from the previous embedding that the tangential cover is canonical meaning that the arbitrary starting point does not influence the resulting graph.

We put on the graph the orientation of the curve  $C$ . For a tangent  $T_{p_i}$ , there exists a finite subset of tangents  $T_{p_j}$  such that  $T_{p_i}$  and  $T_{p_j}$  overlap and  $i$  is at the left of  $j$ . We construct the function  $F(\cdot)$  by mapping  $T_{p_i}$  to the overlapping tangent  $T_{p_j}$  with maximal  $j$  intended modulo  $n$ , that is following the orientation of the graph. A digital polygonalization of  $C$  is a succession of digital segments



**Fig. 1.** (left) the chromosome shape (middle) maximal digital segments (right) its tangential cover

covering  $C$ , sharing only their ending points - the ending point of a digital segment is the beginning point of the next segment - and such that all but the last segment are maximal. This concept is well used in shape description [ZL04] or shape evolution [LL99]. Polygonalizations can be deduced from the function  $F(.)$  [FT03],

**Property 2.1.**  $T \in \mathcal{T}(C) \iff T = F(T)$

Thus using iterates of the function  $F(.)$  and the maximality of any tangent, it is possible to build any polygonalizations of the curve. For instance on the chromosome shape of Fig. 1 containing 60 points, points from 59 to 13 forms a maximal digital segment. The maximal segment starting at 59 ends at 13 and the maximal digital segment starting at 13 ends at 15 and so on.

We now introduce the tangent  $F^*(T)$  for any tangent  $T$  defined as the tangent such that:  $F^*(T) = F^j(T)$  for some  $j$  and there exists a digital tangent in  $\mathcal{T}(C)$  containing both the ending point of  $F^*(T)$  and the beginning point of  $T$ . In other words,  $F^*(T)$  is the last maximal digital segment of the polygonalization starting at the beginning point of  $T$ .

### 3 Polygonal Approximations

#### 3.1 Context

The class of allowed polygonal approximations is exactly the class of digital polygonalizations of the curve  $C$  from any of the starting point of the digital tangents of the tangential cover  $\mathcal{T}(C)$ . We believe that this class is sufficiently rich. Moreover all other points of  $C$  appear inside straight parts and do not appear to be dominant or corner points.

To measure the error between a polygonal approximation and the original curve  $C$ , we use the classical ISSE criterion. We consider two points  $p_i = (x_i, y_i)$  and  $p_j = (x_j, y_j)$  of  $C$ . We introduce  $a = y_j - y_i$ ,  $b = x_j - x_i$  and  $c = x_j y_i - y_j x_i$

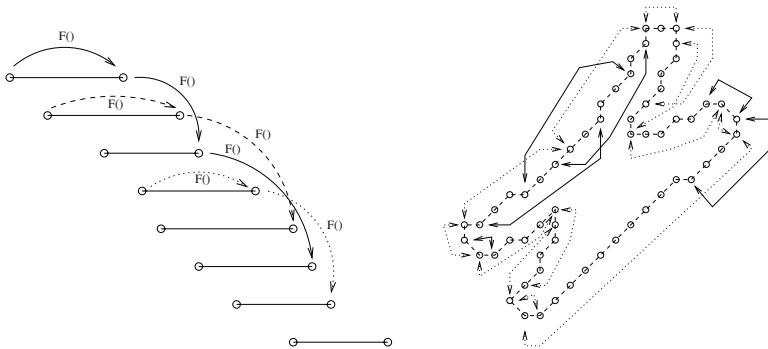
such that all points  $(x, y)$  of the real line passing through  $p_i$  and  $p_j$  satisfies  $ax - by + c = 0$ . Each point  $p_k$  of  $C$  with  $i \leq k \leq j$  are projected onto this line using perpendicular projection and the resulting distance  $d_{ij}(p_k)$  is  $d_{ij}^2(p_k) = (ax_k - by_k + c)^2 / (a^2 + b^2)$ . The ISSE value between  $p_i$  and  $p_j$  is  $\text{ISSE}(p_i \rightarrow p_j) = \sum_k d_{ij}^2(p_k)$  such that

$$(a^2 + b^2) \text{ISE}(p_i \rightarrow p_j) = a^2 \sum_k x_k^2 + b^2 \sum_k y_k^2 + c^2 \sum_k 1 + 2ac \sum_k x_k - 2bc \sum_k y_k - 2ab \sum_k x_k y_k \quad (1)$$

If the sums of eq. (1) can be computed in linear time then so is the ISE criterion. The ISSE of a complete polygonal approximation is defined as the sum of the ISE for each segment of the real polygon.

### 3.2 Method

To compute a polygonal approximation of  $C$ , we start by computing the tangential cover  $\mathcal{T}(C)$ . We call  $\mathcal{P}(C)$  the class of all allowed polygonal approximations. Our goal is to compute the element  $P$  of  $\mathcal{P}(C)$  such that  $\text{ISSE}(P) = \min\{\text{ISSE}(Q), Q \in \mathcal{P}(C)\}$ .



**Fig. 2.** (left) The iterates of  $F(\cdot)$  (right) cycles and paths for the chromosome shape

The first step of the algorithm consists in computing the function  $F(\cdot)$  (the algorithm is described in the next subsection). Having the function  $F(\cdot)$ , we now consider a polygonalization, depicted in Fig. 2 (left). To compute the ISE value of a polygonalization, we must compute two partial ISE values: the first one is the ISE of each tangent and the second one is the ISE between a tangent  $T$  and the tangent  $F(T)$ . Those two computations are done in the second step of the method. Note that in fact we compute the six sums of equation (1).

The third step of the method is a decomposition of the graph of the tangential cover into cycles and paths. To introduce those notions, let us fix an initial

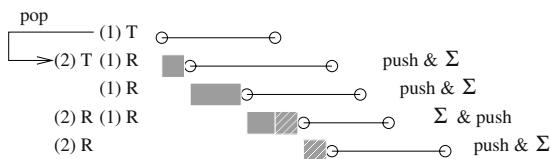
tangent for instance  $T_0 = T_{p_0}$ . If we consider the tangents  $T_j = F^j(T_0)$ , then the list is obviously finite. Hence, there exists a minimal iterates  $j_0$  such that  $F^{j_0+1}(T_0) = F^k(T_0)$  with  $k < j_0 + 1$ . The set of tangents from  $T_0$  until  $T_{j_0}$  forms a cycle in the graph. Let us consider now a tangent  $T'$  not belonging to the previous cycle. Then either the iterates  $F^j(T')$  create a new cycle or they merge with a previously computed cycle and the portion of the graph from  $T'$  to the cycle is a path in the graph. For each path, we compute the label of the cycle it merges with. Those information are stored in each tangent. For the chromosome shape (see Fig. 2 right), there exists only one cycle, depicted in dotted lines, and paths merging to this cycle. The merging process implies that the end of the polygonalization starting at one tangent in a path must end with a tangent of the cycle. So, the function  $F^*(\cdot)$  is defined on a cycle and also concerns all tangents belonging to paths merging with the cycle.

Hence the fourth step consists in computing the  $F^*(\cdot)$  values. To do this, we rely on the merging labels previously computed. We first pick a tangent in a cycle and compute its  $F^*(\cdot)$  value using iterates of  $F(\cdot)$ . Then, we move along the tangential cover graph and update  $F^*(\cdot)$  simply by checking if there still exists one tangent overlapping the current tested tangent and the current  $F^*(\cdot)$  tangent. So in one pass of the tangential cover graph, to each tangent is associated an  $F^*(\cdot)$  tangent.

To complete the computation of ISSE, we must end each polygonalization by the portion of the digital segment linking each tangent  $T$  and  $F^*(T)$ . This is the fifth step of the algorithm. How to do all steps in linear time is explained in next subsection.

### 3.3 Intermediate Constructions and Complexity

In first step, we compute  $F(\cdot)$ . We start with  $T = T_{p_0}$ . By moving to the right, we determine  $F(T)$  simply by checking the overlappings. This step obviously can be done in linear time with one complete turn over the tangential cover graph.



**Fig. 3.** ISE computation of each tangent

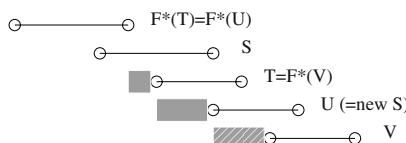
The computation process of the ISE from the beginning point of a tangent to its ending point is given in Fig. 3.  $T$  is the tangent being computed and  $R$  is a moving tangent. The numbers in parentheses mark which  $R$  are associated to which  $T$ . We use a FIFO (First-In First-Out) list. The sum symbol means summation in a global counter. The algorithm proceed by summing from the

beginning of  $T$  to the beginning of  $R$  (excluded) and push each portion in the list. When  $R$  reaches the next tangent of  $F(T)$ , the accumulation buffer contains the value of the six sums of the ISE computation. Then, we do not push since the computation is only partial at this step but we move  $T$  and pop from the list. The pop values are subtracted to the accumulation buffer. The process goes on but summation and push are inverted to push a complete part in the list. By moving all along the tangential cover graph, all partial ISE are computed, since we use the same strategy for the link between  $T$  and  $F(T)$ . The complexity is  $O(\#T(C))$  where  $\#$  denotes cardinal.

To compute cycles and paths, the algorithm works in two steps. First, we initialize the cycle mark to 1 and mark  $T_{p_0}$  with it. We also mark  $F(T_{p_0})$ . We now consider  $T$  to be the next tangent of  $T_{p_0}$ . If it is not marked, we increase the cycle mark by one, mark  $T$  and  $F(T)$  with the mark. If it is already marked, we propagate its mark to  $F(T)$  if this last one is not already marked. At each tangent when a merging is detected, we store it in a lookup table. After a complete turn over the tangential cover, each tangent has a mark and all merging have been stored in the lookup table. However, it might happen that a path merges with another path before merging with a cycle. Hence, the lookup table must be modified in order to have a cycle mark for each path. For instance, let us consider the following lookup table [1, 1, 3, 3, 2, 4, 4] which means that path 7 (cell with index 7 in the array) merges with path 4 and path 4 with cycle 3. The two cycles are 1 and 3 since the numbers in these cells correspond to the indices of the cells. We start with path 7 and move in the lookup table until we reach a cycle then we mark each element with the mark of the cycle. We then proceed with another unmodified element in decreasing order. In one pass, we get the correct lookup table [1, 1, 3, 3, 1, 3, 3]. Then each time we consider a mark, we access to the lookup table to get the merging cycle. So step 3 is also done in linear time.

To compute the  $F^*(\cdot)$  values, we first consider  $T_{p_0}$  and find  $F^*(T_{p_0})$  and store it in a lookup table. We then consider the tangent  $T$  next to  $T_{p_0}$  in the tangential cover. If it does not have a  $F^*(\cdot)$  tangent, we look in the lookup table if an  $F^*(\cdot)$  has already been computed for its cycle. If not, we compute it and store in the lookup table. If there exists a previously computed  $F^*(\cdot)$ , we move it to find  $F^*(T)$  and store it in the lookup table. After, one complete turn over the tangential cover, every tangent  $T$  has an  $F^*(T)$  tangent. The complexity is  $O(\#T(C))$ .

The last step of the algorithm is similar to the previous one. But summation must be done to complete the computation of the ISE of each polygonalization.



**Fig. 4.** ISE computation for the end of the polygonalizations

The process is depicted in Fig. 4. We suppose that  $T$  and  $F^*(T)$  have been computed. To end the computation of the ISE measure, we sum the portion of the curve in grey level and store it in  $T$ . When going to the next tangent  $U$ , we discover that  $F^*(U) = F^*(T)$  since  $S$  also covered  $U$ . Hence, we accumulate the summation in the buffer and store it in  $U$ . Next tangent is  $V$  which has a different  $F^*$ (). So we set the buffer to zero and do the summation. In one turn of the tangential cover using the lookup table for the currently computed  $F^*$ (), we can complete the computation of the ISSE measure for each polygonalization. To see that the whole complexity is  $O(n)$ , we must ensure that the number of time a point is computed is independant of  $n$ . This fact can be deduced from the following remark: when  $F^*$ () change this means that the new  $S$ , equals to  $U$  in Fig. 4, must be strictly at the right of  $S$  such that when the buffer is set to zero, the new accumulation covers points not accumulated before. Hence, we obtain a linear time complexity.

The following theorem summarizes the previous complexity analysis.

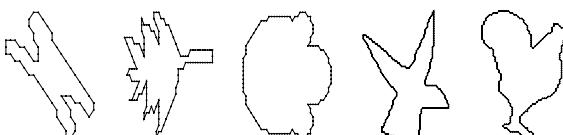
### Theorem 3.1.

$$O(n)$$

$$C \leq n$$

## 4 Experiments

Experiments were performed on different shapes part of which is given on Fig. 5. The first three are taken from [TC89] and the last two from [MS04]. Results are given in Table 1. Several facts appear. First, the method performs well since the numbers of detected vertices are not too high and the ISSE values are relatively small. Second, the quality of the results of our strategy is very stable. Its non-parametric nature must be kept in mind when comparing the results with other methods. Moreover, the complexity of our method is linear time and space. Our method also controls the  $L_\infty$  error measure since it is based on digital segments. When using eight connected digital segments, we can guarantee that the  $L_\infty$  error is strictly below 1. But, thickness of digital lines might be increased. For instance, for the Bird shape, the use of four-connected digital lines leads to a solution in 20 vertices with an ISSE of 65.50. If we used the concept of blurred segments [DRFR05] which forms a fuzzy-like extension of digital segments, when modifying the width of digital segments we can reach a solution in only 19



**Fig. 5.** Different shapes used for experimentation

**Table 1.** Results of experiments

Shapes	Methods	ISSE	# vertices	Shapes	Methods	ISSE	# vertices
Chromosome	Optimal	5.82	12	Semicircle	Optimal	14.40	15
	Optimal	3.13	17		Optimal	2.64	30
	[Cor97]	9.57	12		[Cor97]	13.00	22
	[TC89]	7.2	15		[TC89]	20.61	22
	[RR92]	4.81	18		[RR92]	11.5	27
	[SRS03]	8.53	17		[SRS03]	7.29	30
	New	5.90	14		New	14.59	21
Leaf	Optimal	22.42	17	Bird	[MS04]	72.92	15
	Optimal	6.80	28		New	34.93	36
	[Cor97]	25.80	23	Cock	[MS04]	39.96	24
	[TC89]	14.96	29		New	25.53	39
	[RR92]	14.18	32				
	[SRS03]	59.92	32				
	New	13.77	25				

vertices with an ISSE of 54.70. Our algorithm applies without any modifications since it is only based on the graph structure. Moreover, another extension can be used to also increase the quality of our method by allowing any polygonalizations of  $C$ . We currently cannot prove that our method remains linear time but experiments show computing times coherent with a linear time complexity.

## 5 Conclusions

We have proposed a new non-parametric algorithm for polygonal approximation of digital curves. It is based on digital lines and a canonical graph representation. By restricting the polygonal models to the digital polygonalizations starting at beginning points of digital tangents, our algorithm finds the polygonal model with the minimal ISSE value. The algorithm is linear in time and space complexity. A work is in progress to extend our representation to thick digital lines in order to increase the size of the class of polygonal models. We believe that we will nearly always find the global optimal polygonal model in linear time.

## References

- [Cor97] P. Cornic. Another look at the dominant point detection of digital curves. *Pattern Recognition Letters*, 18:13–25, 1997.
- [DP73] D.H. Douglas and T.K. Peucker. Algorithm for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

- [DRFR05] I. Debled-Rennesson, F. Feschet, and J. Rouyer. Optimal blurred segments decomposition in linear time. In *Digital Geometry and Computer Imagery*, volume 3429 of *LNCS*, pages 371–382. Springer-Verlag, 2005.
- [Fes05] F. Feschet. Canonical representations of discrete curves. *Pattern Analysis and Applications*, 2005. Accepted for publication, to appear.
- [FT99] F. Feschet and L. Tougne. Optimal time computation of the tangent of a discrete curve: application to the curvature. In *DGCI*, volume 1568 of *LNCS*, pages 31–40. Springer-Verlag, 1999.
- [FT03] F. Feschet. and L. Tougne. On the Min DSS Problem of Closed Discrete Curves. In A. Del Lungo, V. Di Gesù, and A. Kuba, editors, *IWCIA*, volume 12 of *ENDM*. Elsevier, 2003.
- [HL02] J-H. Horng and J.T. Li. An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. *Pattern Recognition Letters*, 23:171–182, 2002.
- [II88] H. Imai and M. Iri. Polygonal approximations of a curve (formulations and algorithms). In G.T. Toussaint, editor, *Computational Morphology*, pages 71–86. North-Holland, 1988.
- [KF04] A. Kolesnikov and P. Fränti. Reduced-search dynamic programming for approximation of polygonal curves. *Pattern Recognition Letters*, 24:2243–2254, 2004.
- [LL99] L.J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454, 1999.
- [Mar03] M. Marji. *On the detection of dominant points on digital planar curves*. PhD thesis, Graduate School, Wayne State University, 2003.
- [MS04] M. Marji and P. Siy. Polygonal representation of digital planar curves through dominant point detection - a nonparametric algorithm. *Pattern Recognition*, 37(11):2113–2130, 2004.
- [PV94] J.-C. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15:743–750, 1994.
- [RK04] A. Rosenfeld and R. Klette. Digital Straightness – a review. *Discrete Applied Math.*, 139(1–3):197–230, 2004.
- [RR92] B.K. Ray and K.S. Ray. An algorithm for detecting dominant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, 13:849–856, 1992.
- [Sal01] M. Salotti. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22:215–221, 2001.
- [Sal02] M. Salotti. Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. *Pattern Recognition*, 35:435–443, 2002.
- [Sch03] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Berlin: Springer-Verlag, 2003.
- [SRS03] B. Sarkar, S. Roy, and D. Sarkar. Hierarchical representation of digitized curves through dominant point detection. *Pattern Recognition Letters*, 24:2869–2882, 2003.
- [TC89] C.-H. Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11(8):859–872, 1989.
- [ZL04] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1–3):1–19, 2004.

# Fast Manifold Learning Based on Riemannian Normal Coordinates

Anders Brun<sup>1,3</sup>, Carl-Fredrik Westin<sup>3</sup>, Magnus Herberthson<sup>2</sup>, and Hans Knutsson<sup>1</sup>

<sup>1</sup> Department of Biomedical Engineering,  
Linköpings Universitet, Linköping, Sweden  
[{andbr, knutte}@imt.liu.se](mailto:{andbr, knutte}@imt.liu.se)

<sup>2</sup> Department of Mathematics,  
Linköpings universitet, Linköping, Sweden  
[maher@mai.liu.se](mailto:maher@mai.liu.se)

<sup>3</sup> Laboratory of Mathematics in Imaging,  
Harvard Medical School, Boston, MA, USA  
[westin@bwh.harvard.edu](mailto:westin@bwh.harvard.edu)

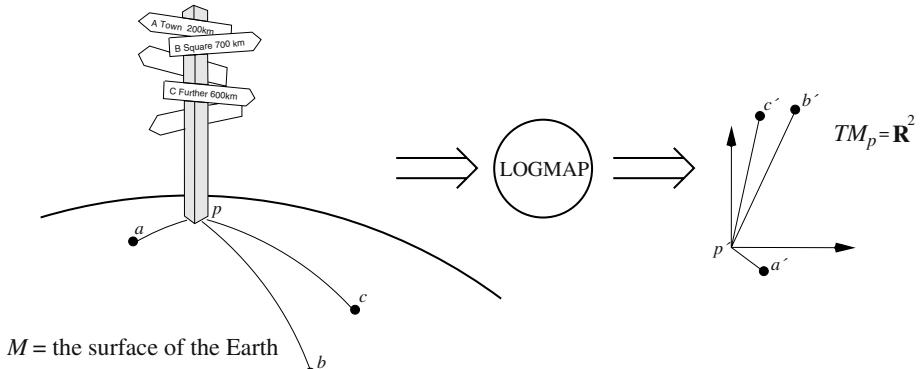
**Abstract.** We present a novel method for manifold learning, i.e. identification of the low-dimensional manifold-like structure present in a set of data points in a possibly high-dimensional space. The main idea is derived from the concept of Riemannian normal coordinates. This coordinate system is in a way a generalization of Cartesian coordinates in Euclidean space. We translate this idea to a cloud of data points in order to perform dimension reduction. Our implementation currently uses Dijkstra's algorithm for shortest paths in graphs and some basic concepts from differential geometry. We expect this approach to open up new possibilities for analysis of e.g. shape in medical imaging and signal processing of manifold-valued signals, where the coordinate system is "learned" from experimental high-dimensional data rather than defined analytically using e.g. models based on Lie-groups.

## 1 Introduction

A manifold can be seen as a generalization of a surface to higher dimensions. Locally a manifold looks like a Euclidean space,  $\mathbb{R}^N$ , but on a global scale it may be curved and/or compact, like a sphere or a torus. A manifold with a metric tensor defined at each point is called a Riemannian manifold.

Recent developments in so called manifold learning has opened up new perspectives in non-linear data analysis. Classical methods such as Principal Components Analysis (PCA, a.k.a. the Karhunen-Loeve transform) and Multidimensional Scaling (MDS) efficiently finds important linear subspaces in a set of data points. Methods within the field of manifold learning are however able to identify non-linear relations as well. In this paper we present a new tool for data analysis of this kind, based on the concept of Riemannian normal coordinates.

Manifold learning has become an established field of research, Kohonen's Self Organizing Maps (SOM) [5] being an important early example. Characteristic for the newest generation of manifold learning techniques is efficiency and global convergence,



**Fig. 1.** Traveling along a geodesic, starting at a specific location in a specific direction, will eventually take you to any place on the surface of the Earth. Riemannian normal coordinates captures this information, mapping points on the sphere to  $\mathbb{R}^2$  in a way that direction and geodesic distance from the origin to any point is preserved. Riemannian normal coordinates are therefore quite natural to use for navigation on a manifold, at least in the close vicinity of a point. Also note that geodesics on a manifold  $M$  (left) are mapped to lines in Riemannian normal coordinates (right)

in particular many of them are based on the solution of very large eigenvalue problems. This include for instance the recent Kernel PCA [8], Locally Linear Embedding [7], ISOMap [11], Laplacian Eigenmaps [1] and Hessian Eigenmaps [2].

Manifolds arise in data for instance when a set of high-dimensional data points can be modeled in a continuous way using only a few variables. A typical example is a set of images of a 3-D object. Each image may be represented as a very high-dimensional vector, which depends on the scene and a few parameters such as relative camera orientation, camera position and lighting conditions. Camera orientation itself is a good example of a non-linear manifold. The manifold of orientations,  $SO(3)$ , can be represented by the set of all rotation matrices. While the manifold-valued parameter space is equivariant to important features of the data, namely camera- and lighting information, it should also be invariant to uninteresting things such as noise from the image sensors.

In the following sections we present a novel technique for manifold learning based on the concept of Riemannian normal coordinates. We have translated this technique from its original setting in differential geometry, to the task of mapping a set of experimental high-dimensional data points, with a manifold-like structure, to a low-dimensional space. An intuitive explanation of Riemannian normal coordinates is given in figure 1. They contain information about the direction and distance from a specific point on a manifold to other nearby points. The usefulness of such information for navigation is obvious, not only for navigating on the Earth, but also for creating user interfaces to navigate in manifold-valued data in general. The Riemannian normal coordinates are also closely related to geodesics and the exponential and logarithmic maps of Lie-groups, which have been used recently for the analysis of shape in medical images [4] and to perform time-dependent signal processing of orientation data [6].

## 2 Theory

In this section we briefly review some basic concepts of differential geometry necessary to understand the method we propose.

To each point  $p$  on a manifold  $M$  there is a associated tangent space,  $T_p M$ , consisting of a Euclidean space tangential to  $M$  at  $p$ . Derivatives at  $p$  of smooth curves passing through a point  $p$  belongs to  $T_p M$ .

A special kind of curves defined on Riemannian manifolds are the geodesics, i.e. a length minimizing curve on  $M$ . These define a metric  $d(x, y)$  on a manifold derived from the length of a geodesic passing through  $x$  and  $y$ .

The Riemannian exponential map,  $\exp(v) \in M$ ,  $v \in T_p M$ , is a function which maps points in the tangent space of  $p$ , to points on  $M$ . If  $H(t)$  is the unique geodesic, starting at  $p$  with velocity  $v$ , then  $\exp(v) = H(1)$ . Intuitively this can be thought of as walking with constant velocity in particular direction on the manifold, from a point  $p$ , during one time unit. This mapping is one-to-one in a neighborhood of  $p$  and its inverse is the log map.

The set of points on  $M$  for which there exists more than one shortest path from  $p$  is called the *cut locus* of  $p$ . The cut locus of a point on a sphere is for instance its antipodal point. Some manifolds, such as  $\mathbb{R}^2$ , lack a cut locus. Other manifolds, such as the torus, have a quite complex looking cut locus.

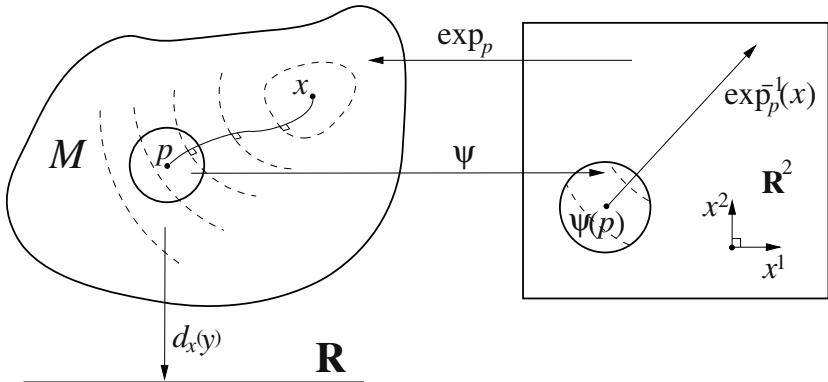
Given a point  $p$  and an orthonormal basis  $\{\hat{\mathbf{e}}_i\}$  for the tangent space  $T_p M$ , a Riemannian normal coordinate system is provided by the exponential mapping. A point  $x \in M$  gets the coordinate  $(x^1, \dots, x^N)$  if  $x = \exp(x^i \hat{\mathbf{e}}_i)$ .

The gradient of a scalar function  $f$  is a dual vector field which components are simply the partial derivatives (in the induced basis).

## 3 Method

Given a basis point  $p$  from a data set  $X$  and an orthonormal basis of the tangent space at  $p$  to a thought manifold  $M$ , we would like to, via the log map into  $T_p M$ , express all data points  $x \in X$  using Riemannian normal coordinates. Due to the properties of Riemannian normal coordinates, this is equivalent to measuring the distance and direction from  $p$  to every other point in the data set. We choose to call this framework LOGMAP:

1. From a set of data points,  $X$ , sampled from a manifold  $M$ , choose a base point  $p \in X$ .
2. To determine the dimension of  $M$ , select a ball  $B(p)$  of the  $K$  closest points around  $p$ . Then perform standard PCA in the ambient space for  $B(p)$ . This will give us  $T_p M$ , with  $\dim T_p M = N$ , where we choose any suitable ON-basis  $\{\hat{\mathbf{e}}_i\}$ . All  $y \in B(p)$  are mapped to  $T_p M$  by projection on  $\{\hat{\mathbf{e}}_i\}$  in the ambient space. This is the  $\Psi$ -mapping in figure 2.
3. Approximate distances on  $M$ . In the current implementation we do this by defining a weighted undirected graph, with each node corresponding to a data point and with edges connecting each node to its  $L$  closest neighbors. Let the weights of these



**Fig. 2.** A schematic illustration of a geodesic from  $x$  to  $p$  in a manifold  $M$ . Dashed curves correspond to iso-levels of  $d_x^2(y) = d^2(x, y)$ . These iso-curves are perpendicular to every geodesic passing through  $x$ . The ball around  $p$  and the mapping  $\Psi$  defines a chart that maps a part of  $M$  to  $\mathbb{R}^2$ . The domain of  $\exp$  is actually the tangent space of  $M$  at  $p$ , and it is natural to identify vectors in  $\mathbb{R}^2$  with  $TM_p$

edges be defined by the Euclidean distance between data points in the ambient space. We then use Dijkstra's algorithm for finding shortest paths in this graph, to approximate the geodesic distances in  $M$ . This gives estimates of  $d(x, y)$  for all  $(x, y) \in X \times B(p)$ .

4. To calculate the direction from  $p$  to every point  $x \in X$ , estimate  $\mathbf{g} = \sum g^i \hat{\mathbf{e}}_i = \nabla_y d^2(x, y)|_{y=p}$  numerically, using the values obtained in the previous step. While we only have values of  $d^2(x, y)$  for  $y \in B(p)$ , we must interpolate this function in  $T_p M$ , e.g. using a second order polynomial, in order to calculate the partial derivatives at  $\Psi(p)$ .
5. Estimates of Riemannian normal coordinates for a point  $x$  are then obtained as  $x^i = d(x, p) \frac{g^i}{|\mathbf{g}|}$ .

In step 4) above, the numerical calculation of the gradient at  $p$  uses the squared distance function. The reason for not just taking the gradient at  $p$  of the plain distance function from  $x$ , which is known to point in the direction of the geodesic connecting  $p$  and  $x$ , is that it is not smooth for  $p \approx x$ . Using the square of the distance function, which is much easier to interpolate, solves this problem while giving a gradient in the same direction. However, when  $x$  is close to the cut locus of  $p$ , even the squared distance function becomes non-smooth. In the experiments shown in the next section, we have actually used a slightly more robust scheme to estimate the gradient for points close to the cut locus. This was done by using the RANSAC algorithm [3] for selecting only a small number of points around  $p$  to use in the interpolation step of the squared distance function.

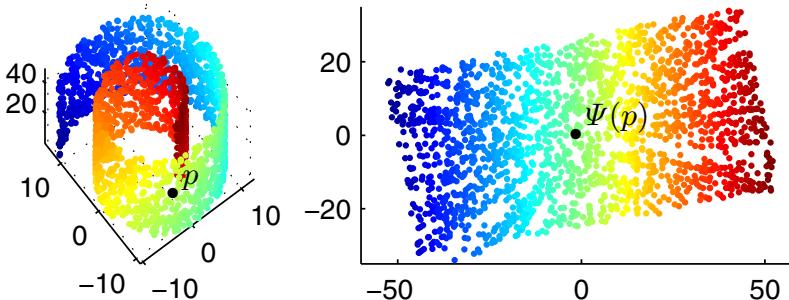
## 4 Experiments

The LOGMAP method was evaluated using Matlab. The most critical part of the algorithm, the calculation of shortest paths, was borrowed from the ISOMAP implementation of Dijkstra's shortest paths [11]. In the LOGMAP implementation, the selection of  $p$  was made interactively by the click on the mouse and the resulting log map was calculated almost in real time.

Three experiments on synthetic data are presented here to illustrate the behavior of the algorithm. In each of the experiments we have assumed knowledge of how to choose  $L$ , the number of neighbors for building the graph, and  $K$ , which determines the size of the neighborhood used for dimension estimation and later the estimation of gradients. It is important to point out that selection of these parameters is actually non-trivial for many data sets, e.g. when noise is present. We will not go further into the details of choosing these constants in this paper however.

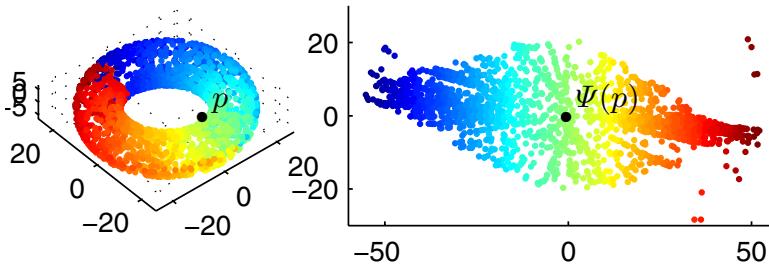
### 4.1 The Swiss Roll

In the first experiment we use the “Swiss roll” data set, consisting of points sampled from a 2-D manifold, embedded in  $\mathbb{R}^3$ , which looks like a roll of Swiss cheese. It has been used before to illustrate methods for manifold learning, see e.g. [11, 7], and we include it mainly as a benchmark. A set of 2000 points from this data set were used in the experiment and the results are presented in figure 3. The experiment shows that the LOGMAP method correctly unfolds the roll and maps it to Riemannian normal coordinates in  $\mathbb{R}^2$ .



**Fig. 3.** A set of 2000 points from the “Swiss roll” example [11]. Colors correspond to the first Riemannian normal coordinate derived from the method. **Left:** The original point cloud embedded in 3-D. **Right:** Points mapped to 2-D Riemannian normal coordinates

It is important to note that the resulting mapping in the Swiss roll example is more or less isometric, which is expected for simple flat manifolds. This is similar to the behavior of ISOMAP. On the other hand, both ISOMAP and LOGMAP would fail to produce isometric embeddings if we would introduce a hole in the Swiss roll data set. This particular problem is solved by Hessian Eigenmaps for flat manifolds.



**Fig. 4.** A set of 2000 points from a torus embedded in 3-D. Colors correspond to the first Riemannian normal coordinate derived from the method. **Left:** The original point cloud embedded in 3-D. Notice the discontinuity (red-blue) in the coordinate map, revealing a part of the “cut locus” of  $p$ . **Right:** Points mapped to 2-D Riemannian normal coordinates. Because the metric of a torus embedded in 3-D is not flat, the manifold is not mapped to a perfect rectangle. Some outliers are present, due to incorrect estimation of the gradient for points near the cut locus

#### 4.2 The Torus

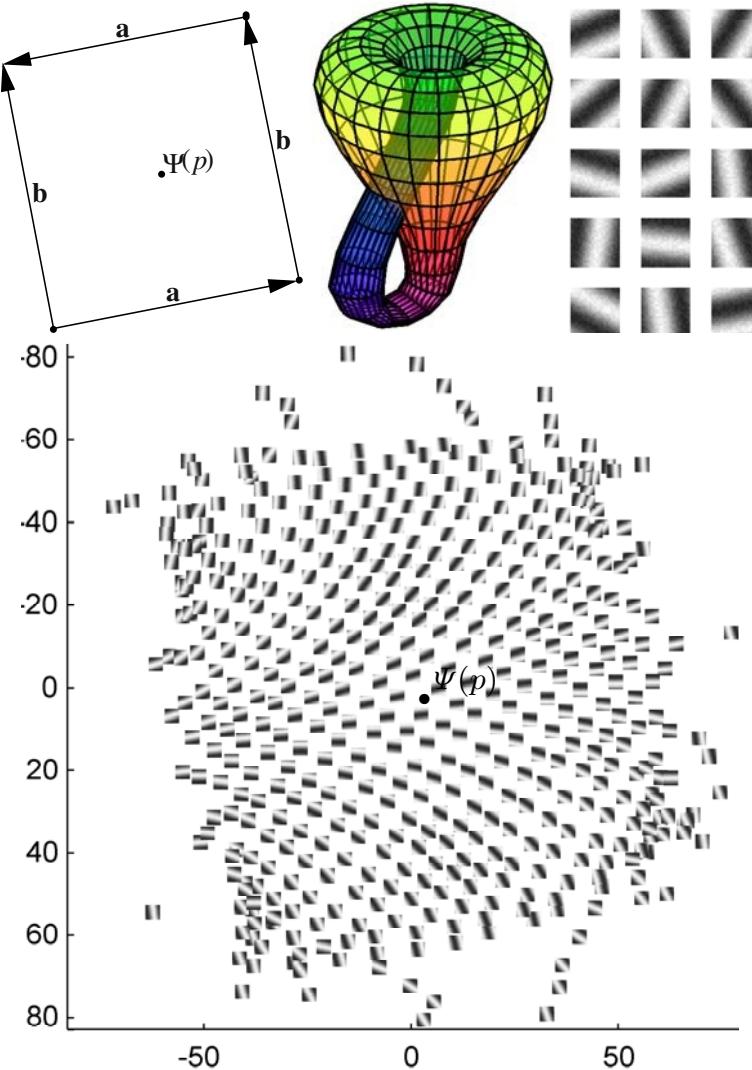
In the second experiment we tested the method on a data set consisting of 2000 points from a torus embedded in 3-D. The results in figure 4 illustrate how the method cuts the coordinate chart at the cut locus of the point  $p$ . This particular behavior of “cutting up” the manifold allows us to save one dimension in this particular example. There is no embedding of the torus into  $\mathbb{R}^2$ . Any standard method for dimension reduction, e.g. LLE, Laplacian Eigenmaps or ISOMAP, would embed this manifold into  $\mathbb{R}^3$  at best. However, the automatic introduction of a cut by the LOGMAP method makes it possible to make a one-to-one mapping of this manifold to  $\mathbb{R}^2$ .

#### 4.3 The Klein Bottle

The third experiment finally, shown in figure 5, tests the method on truly high-dimensional data. The data set consists of  $21 \times 21$  pixel image patches. Each of the 2-D image patches were rendered as a 1-D sine wave pattern with a specific phase and orientation. A small amount of normal distributed white noise was also added to the images. The resulting data set consisted of 900 data points, in a 441-dimensional space, representing image patches sampled uniformly from all possible parameters of phase and orientation. It is natural to assume that the intrinsic dimensionality of this data set is 2, since the variables phase and orientation adds one degree of freedom each.

The mapping of image patches to  $\mathbb{R}^2$  is visualized by simply using the image patches as glyphs, placed at various locations in the plane. We observed slightly different shapes of the cut locus, i.e. the border of the resulting map, depending on the choice of base point  $p$ . Even though the data set seems to be highly symmetric in terms of orientation and phase, the patches themselves have a square shape. This lack of rotation invariance will in turn affect the shape of the manifold of all image patches due to lack of rotation invariance.

By carefully identifying the edges of the cut locus, we manually obtain an interpretation of the mapping shown in the top left of figure 5. The resulting directed labeled



**Fig. 5.** To test the proposed method on a high-dimensional data set, a set of 900 image patches, each being of  $21 \times 21$  pixels with a characteristic orientation/phase, were generated and mapped to Riemannian normal coordinates. This experiment reveals the Klein bottle-structure of local orientation/phase in 2-D image patches! **Top left:** An idealized Klein bottle aligned to the mapping below. Edges correspond to the cut locus of  $p$  and should be identified according to the arrows. **Top middle:** An immersion of the Klein bottle in 3-D. **Top right:** 15 random examples of image patches used in the experiment. **Bottom:** The mapping of image patches to Riemannian normal coordinates using the proposed method

graph reveals that the topology of this particular image manifold is actually the well known Klein bottle [12]. Similar conclusions for the topology of local descriptions

of phase and orientation has previously been described in [10, 9], where the topology of Gabor filters is derived from theoretical investigations. Our investigation is on the contrary experimental, and to the best of our knowledge it is a new example of how manifold learning can be used to experimentally infer the topology of a data set.

## 5 Discussion

The presented LOGMAP method is rather different from many other methods for manifold learning and dimension reduction, both in terms of the output and in terms of algorithmic building blocks.

The possibility of a cut, a discontinuity in the mapping at the so called cut locus could be a problem in some applications, but it can also be seen as a feature. For instance it allows the torus and Klein bottle to be visualized using only a two-dimensional plot. Other methods, see for instance [11, 7, 8, 1, 2], tries to find a continuous embedding of the manifold, and for that at least 3 dimensions are needed for the torus and the Klein bottle needs at least 4 dimensions. (The top middle illustration in figure 5 is actually an example of an immersion and not an embedding of the Klein bottle in 3-D, meaning roughly that it intersects itself at some points.)

The use of other criteria for assigning a global coordinate system to a manifold could also be considered, for instance conformal mappings of 2-D manifolds. In almost every case when mapping a manifold to a low-dimensional space, some kind of distortion is introduced while some features of the original manifold will be preserved. For most manifolds, Riemannian normal coordinates create a very distorted mapping far away from the base point  $p$ , in some cases they even introduce a cut. However, they also preserve all geodesic distances and angles from  $p$  to other points on the manifold, which makes this mapping quite intuitive and particularly useful for the purpose of navigating inside a manifold. At least this is true in the close vicinity of the base point  $p$ .

In this paper we have chosen examples of manifolds with low intrinsic dimensionality, mainly to illustrate the method, but in principle the method works for manifolds of higher dimensionality too. In the examples we have also used only little or no noise. While this can be seen as very optimistic assumptions about the data, we would like to stress the main algorithmic building blocks of LOGMAP:

1. Approximation of distances on a manifold given a set of sampled data points.
2. Calculation of gradients on manifolds, from a set of function values defined at the sampled data points.

For the first block we have used Dijkstra's method, mainly inspired by the ISOMAP implementation. This method has obvious problems to truthfully approximate distances, because distances are measured along zigzag trajectories in a graph. One obvious way to make LOGMAP more accurate is therefore to switch to a more accurate method based on higher order approximations of the manifold.

The second building block, which is about calculating gradients, could also be improved a lot compared to the current implementation. Measuring gradients for smooth functions is not a problem, but for points close to the cut locus the distance function will introduce a discontinuity which makes the problem quite delicate. The difficulties

of gradient estimation manifest itself by producing spurious points in the mapping, most easily seen in the torus and Klein bottle example, close to the cut locus.

It is also important to stress that the LOGMAP method does not try to explicitly deal with noise. Instead we have moved that problem into the rather challenging task of measuring distances on the manifold.

Finally it is important to mention the efficiency of the LOGMAP method. First of all it does not involve the solution of any large eigenvalue problem, in contrast to many other methods for manifold learning. Instead it relies totally on the ability to fast approximate distances on the manifold and calculate gradients. The key observation is that distances  $d(x, y)$  are only calculated for pairs  $(x, y) \in X \times B(p)$ . This is far less demanding than calculating the distance for all pairs  $(x, y) \in X \times X$ !

In summary, we have introduced a novel method for manifold learning with interesting mathematical and computational properties. We have also provided an example of how manifold learning can assist in identifying a rather non-trivial manifold, in this case a Klein bottle, from a high-dimensional data set. We believe this to be of general interest to people within the fields of manifold learning and medical image analysis, to for instance develop better tools for shape analysis, and to inspire the future development of manifold learning and manifold-valued signal processing in general.

## Acknowledgements

We gratefully acknowledge the support of the Manifold Valued Signal Processing project by the Swedish Research Council (Vetenskapsrådet, grant 2004-4721) and the SIMILAR network of excellence (<http://www.similar.cc>), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609).

## References

1. Belkin, M., Niyogi., P: Laplacian eigenmaps and spectral techniques for embedding and clustering, in T. G. Dietrich, S. Becker, and Z. Ghahramani (eds:) *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, 2002.
2. Donoho, D. L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data PNAS, vol. 100, no. 10, pp. 5591–5596, May 13, 2003.
3. Fischler, M. A., Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.
4. Fletcher, P.T., Joshi, S., Lu, C., Pizer, S.: Gaussian Distributions on Lie Groups and Their Application to Statistical Shape Analysis, In proc. of Information Processing in Medical Imaging (IPMI), pages 450–462, 2003.
5. Kohonen, T.: Self-organized formation of topologically correct feature maps, Biological Cybernetics, vol. 43, pp. 59–69, 1982.
6. Lee, J., Shin, S. Y.: General construction of time-domain filters for orientation data, IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 2, pp. 119–128, 2002.
7. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500): 2323–2326, 22 December 2000.

8. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* 10 (5): 1299–1319, 1998.
9. Swindale, N.V.: Visual cortex: Looking into a Klein bottle, *Current Biology* 6(7), pp. 776–779, 1996.
10. Tanaka, S.: Topological analysis of point singularities in stimulus preference maps of the primary visual cortex, *Proc. R. Soc. Lond. B*, 261: 81–88, 1995.
11. Tenenbaum, J. B., de Silva, V., Langford, J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290 (5500): 2319–2323, 22 December 2000.
12. Weisstein, E. W.: "Klein Bottle." From MathWorld—A Wolfram Web Resource.  
<http://mathworld.wolfram.com/KleinBottle.html>

# TIPS: On Finding a Tight Isothetic Polygonal Shape Covering a 2D Object

Arindam Biswas<sup>1</sup>, Partha Bhowmick<sup>1</sup>, and Bhargab B. Bhattacharya<sup>2</sup>

<sup>1</sup> Computer Science and Technology Department,  
Bengal Engineering and Science University, Shibpur, Howrah, India  
`{abiswas, partha}@becls.ac.in`

<sup>2</sup> Center for Soft Computing Research,  
Indian Statistical Institute, Kolkata, India  
`bhargab@isical.ac.in`

**Abstract.** The problem of constructing a tight isothetic outer (or inner) polygon covering an arbitrarily shaped 2D object on a background grid, is addressed in this paper, and a novel algorithm is proposed. Such covers have many applications to image mining, rough sets, computational geometry, and robotics. Designing efficient algorithms for these cover problems was an open problem in the literature. The elegance of the proposed algorithm lies in utilizing the inherent combinatorial properties of the relative arrangement of the object and the grid lines. The shape and the relative error of the polygonal cover can be controlled by changing the granularity of the grid. Experimental results on various complex objects with variable grid sizes have been reported to demonstrate the versatility, correctness, and speed of the algorithm.

## 1 Introduction

The problem of finding an optimal outer (or inner) polygonal envelope, imposed by isothetic grid lines, for an object, is a grave and critical one, and its solution can be useful to many interesting applications, such as image mining [5], grasping objects by a robot [2, 3, 6], deriving free configuration space (path-planner) for robot navigation [4], lower and upper approximations in rough sets [8, 9], VLSI layout design [7], etc.

The challenge of the problem lies in the fact that, at each grid point, a decision has to be made for the next path to be followed. It may so happen that a current decision based on the local arrangement, may eventually lead to a situation where no further advancements can be made. As a result, it may become mandatory to revoke the earlier decisions in order to backtrack and branch out iteratively until the final solution is found, which would incur excessive computational cost.

Proposed in this paper is a fast, efficient, and elegant algorithm for finding the optimal isothetic inner and outer polygons of an object, where the object can have any arbitrary shape. The elegance and novelty of our algorithm lies in the fact that it takes into account the spatial arrangement of the grid lines

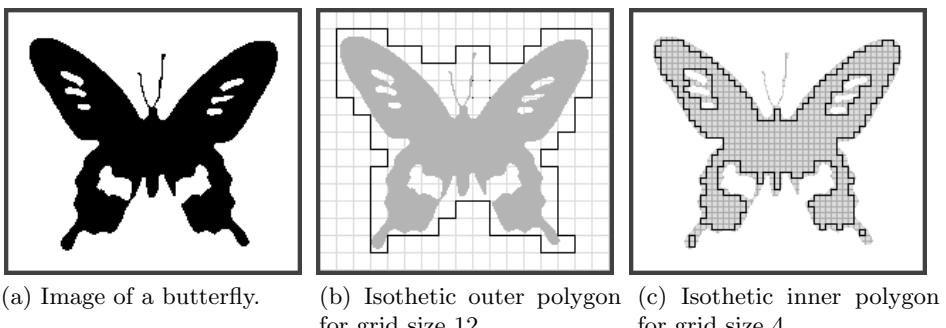
with respect to the object, and constructs the polygon to avoid backtracking completely. Further, this algorithm, with suitable modifications, will also work if non-uniform isothetic grid lines are imposed on the object plane.

It may be mentioned that, in contrast to the classical safety zone problem [7], which computes a minimum area safety region for an input polygon using the Minkowski sum, and also to the problem on inner and outer approximations of polytopes [1], which approximates a convex polytope by a collection of (hyper)boxes, the proposed algorithm can be applied on an arbitrarily shaped object/image, independent of whether or not it is a polygon. Since it works on isothetic grid lines, it can be used for VLSI design rule checking by adjusting the grid space as dictated by the minimum clearance zone required to be maintained. In the present form, this algorithm works on uniform grid spacing; however, this can be easily extended to non-uniform grid spacing which delineates the outer approximation of rough sets of very complex types. Furthermore, this algorithm, in a converse form, can extract the area-maximized isothetic inner polygon, thereby enabling the determination of inner approximation and boundary region of rough sets.

In this paper, after stating the problem definition in Sec. 2, the major steps of the algorithm to find the isothetic outer polygon (area-minimized) are narrated and explained in Sec. 3. In Sec. 4, we have shown experimental results on several objects. Since the algorithm of finding the inner polygon of an object will be very much similar to that of finding the outer one, we have not discussed the details of finding the inner polygon; however, results have been shown for both inner and outer polygons.

## 2 Problem Definition

Given a region (object)  $\mathcal{R}$  defined in the two-dimensional real plane  $\mathbf{R}^2$ , and a set of uniformly spaced horizontal and vertical grid lines,  $\mathcal{G} = (\mathcal{H}, \mathcal{V})$ , where  $\mathcal{H}$  and  $\mathcal{V}$  represent two sets of equispaced horizontal and vertical grid lines respectively



**Fig. 1.** A sample 2D object and its isothetic polygons

(uniform grid), the problem is to construct the corresponding isothetic polygonal envelope,  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , such that the following conditions are satisfied:

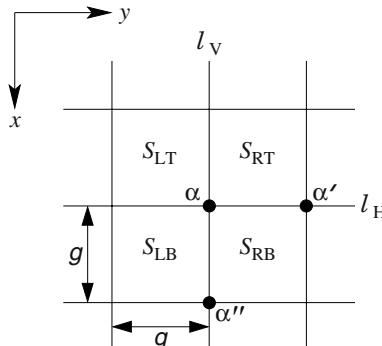
- (c1)  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  should not have any self-intersection and should not contain any hole (although  $\mathcal{R}$  may be self-intersecting and may contain holes);
- (c2) no point  $p \in \mathbf{R}^2$ , lying in the region  $\mathcal{R}$ , should lie outside  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ ;
- (c3) each vertex of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  is the point of intersection of some line in  $\mathcal{H}$  and some line in  $\mathcal{V}$ ;
- (c4) area of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  is minimized.

Before going into the discussion about the algorithm, a sample object  $\mathcal{R}$ , mapped to the 2D discrete plane, has been shown in Fig. 1(a), and the corresponding isothetic outer polygon, completely “containing”  $\mathcal{R}$ , has been displayed in Fig. 1(b). In Fig. 1(c), the entire set of isothetic inner polygons, that completely “fills”  $\mathcal{R}$ , has been shown.

### 3 Proposed Algorithm

Let  $\mathcal{I}$  be the smallest two-dimensional image plane that contains the entire object  $\mathcal{R}$ . Let  $g$  be the underlying grid size, which is equal to the length (i.e. height or width) of each unit grid square in  $\mathcal{G}$ , defined over  $\mathcal{I}$  (Fig. 2). It may be noted that the height  $h$  and the width  $w$  of the plane  $\mathcal{I}$  are chosen appropriately to suit the requirement that  $g$  divides both  $h$  and  $w$ . Let  $\alpha(i, j)$  be the point of intersection of the horizontal grid line  $l_H : x = i$  and the vertical grid line  $l_V : y = j$ , where,  $l_H \in \mathcal{H}$  and  $l_V \in \mathcal{V}$ . It may be observed that, since  $\alpha(i, j)$  is a point on grid with grid size  $g$ ,  $g$  always divides  $i$  and  $j$ , i.e.,  $i \pmod g = 0$  and  $j \pmod g = 0$ . Let  $S_{LT}(i, j)$ ,  $S_{RT}(i, j)$ ,  $S_{LB}(i, j)$ , and  $S_{RB}(i, j)$  represent the four unit grid squares surrounding the common grid node  $\alpha$ , and lying at the left-top, right-top, left-bottom, and right-bottom block respectively, as shown in Fig. 2.

We define a function  $\varphi$  to construct a binary matrix  $M_e$  (Fig. 2) that stores the information regarding the intersection of each of the unit grid



**Fig. 2.** Four unit grid squares with common vertex  $\alpha$

edges (of length  $g$ ) with the object  $\mathcal{R}$ . It may be noted that the total number of unit horizontal edges is  $\frac{w}{g}(\frac{h}{g} + 1)$ , and total number of unit vertical edges is  $\frac{h}{g}(\frac{w}{g} + 1)$ , which togetherly decide the size of the unit edge matrix,  $M_e$ . Let  $\alpha'(i, j+g)$  be the grid point lying immediate right of  $\alpha(i, j)$ , and  $e(\alpha, \alpha')$  be the unit horizontal grid edge connecting  $\alpha$  and  $\alpha'$  (Fig. 2). Similarly, let  $\alpha''(i+g, j)$  be the grid point lying immediate below  $\alpha(i, j)$ , and  $e(\alpha, \alpha'')$  be the unit vertical grid edge connecting  $\alpha$  and  $\alpha''$ . Then the function  $\varphi$ , defined as follows, indicates the entry place,  $\langle i_{e(\alpha, \beta)}, j_{e(\alpha, \beta)} \rangle$ , in  $M_e$  where the binary information about the intersection of the unit edge  $e(\alpha, \beta)$  with the object  $\mathcal{R}$  is stored:

$$\varphi : e(\alpha, \beta) \mapsto \begin{cases} \langle 2i/g, j/g \rangle, & \text{if } \beta = \alpha'; \\ \langle i/g, 2j/g \rangle, & \text{if } \beta = \alpha''. \end{cases} \quad (1)$$

Depending on the intersection of the edge  $e(\alpha, \beta)$  with the object  $\mathcal{R}$ , the corresponding entry  $M_e[i_{e(\alpha, \beta)}][j_{e(\alpha, \beta)}]$  is decided as follows:

$$M_e[i_{e(\alpha, \beta)}][j_{e(\alpha, \beta)}] = \begin{cases} 1, & \text{if edge } e(\alpha, \beta) \text{ intersects } \mathcal{R}; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Now from the unit edge matrix,  $M_e$ , we construct another binary matrix, called the, ...,  $M_s$ , having  $h/g$  rows and  $w/g$  columns, as follows.

$$M_s[i_s][j_s] = \begin{cases} 1, & \text{if } ((M_e[2i_s][j_s] \text{ OR } M_e[2i_s + 1][j_s] \text{ OR } M_e[2i_s + 1][j_s + 1] \text{ OR } M_e[2i_s + 2][j_s]) = 1) \\ & \text{OR} \\ & \text{OR} \\ & \text{OR} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

It may be observed that, if an entry in the unit square matrix,  $M_s$ , is unity, then the corresponding unit square grid in  $\mathcal{I}$  contains some part of the object  $\mathcal{R}$ . More precisely, if  $M_s[i_s][j_s] = 1$ , then the unit grid square, formed by the four grid lines (two horizontal and two vertical), namely  $x = gi_s$ ,  $x = g(i_s + 1)$ ,  $y = gj_s$ , and  $y = g(j_s + 1)$ , contains some part of  $\mathcal{R}$ . Furthermore, as evident from Eqn. 3,  $M_s[i_s][j_s] = 1$  if and only if at least one among  $M_e[2i_s][j_s]$ ,  $M_e[2i_s + 1][j_s]$ ,  $M_e[2i_s + 1][j_s + 1]$ , and  $M_e[2i_s + 2][j_s]$  is unity, since the object  $\mathcal{R}$  must penetrate at least one of the four edges of the unit grid square in order that it lies inside the concerned square.

### 3.1 Finding the Vertices of $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$

In the proposed algorithm, the candidature of  $\alpha$  as a vertex of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  is checked by looking at the combinatorial arrangements (w.r.t. object containments) of the four unit grid squares having common vertex  $\alpha$  (Fig. 2). It may be noted that, each of these four unit squares has a binary entry at the corresponding locations in the, ...,  $M_s$ , which, in turn, is derived from Eqns. 1, 2, 3, as discussed above. These four entries together form a  $2 \times 2$  submatrix in  $M_s$ . Now, there exist  $2^4 = 16$  different arrangements of these 4 unit grid squares, since each square have 2 possibilities ('0'/'1'). These 16 arrangements

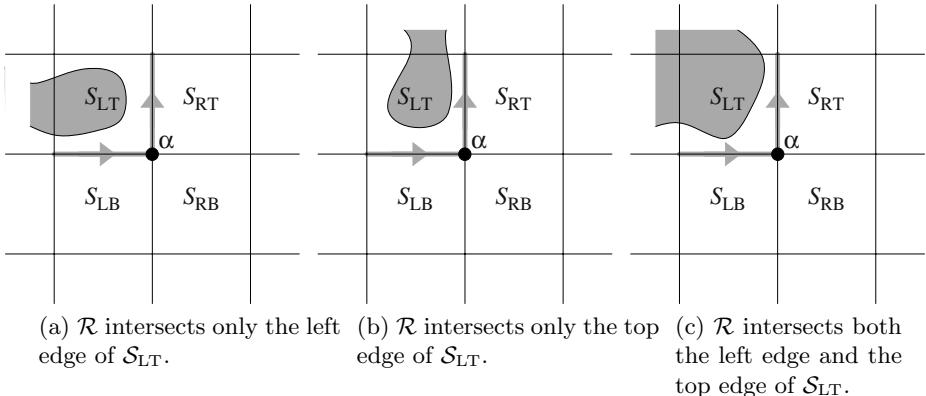
have been further reduced to 5 cases in this algorithm, where, a particular case  $\mathbf{C}_q$ ,  $q = 0, 1, \dots, 4$ , includes all the arrangements where exactly  $q$  out of these 4 squares have object containments (i.e. contain parts of the object  $\mathcal{R}$ ), and the remaining (i.e.  $4 - q$ ) ones have not. That is, the case in which the sum of the 4 bits in the corresponding entries in  $M_s$  is equal to  $q$  is represented by  $\mathbf{C}_q$ . Further, out of these 5 cases, only two cases, namely  $\mathbf{C}_1$  and  $\mathbf{C}_3$ , always represent vertices of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , and one case, namely  $\mathbf{C}_2$ , may conditionally give rise to a vertex of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , as discussed below.

### Case $\mathbf{C}_1$ :

In this case, exactly one of the four unit grid squares contains some part of the object  $\mathcal{R}$ . W.l.g., let  $S_{LT}(i, j)$  be the unit grid square that contains some part of  $\mathcal{R}$ , as shown in Fig. 3. Hence the isothetic envelope,  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , will have its one horizontal edge ending at  $\alpha$  and the next vertical edge starting from  $\alpha$ , if we consider traversal along the edges of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  in a way such that the region/object  $\mathcal{R}$  always lies to the left of each edge while being traversed (shown by the respective arrows in Fig. 3). This argument holds for each of the 4 arrangements where exactly one of the corresponding four binary entries in  $M_s$  is unity and the remaining three is zero. This observation leads to the fact that  $\alpha$  is a  $90^\circ$  vertex (i.e. a vertex with  $90^\circ$  internal angle) of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , if and only if  $q = 1$ .

### Case $\mathbf{C}_2$ :

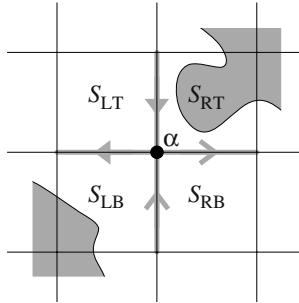
Here, exactly two of the four unit grid squares contain parts of  $\mathcal{R}$ . If the two unit grid squares, having object containments, do not have any unit grid edge in common, only then  $\alpha$  will be vertex of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ . It is easy to observe that, in case  $\mathbf{C}_2$ , only two different arrangements are possible for which  $\alpha$  is a vertex;



**Fig. 3.** Three possible instances for one of the 4 arrangements of case  $\mathbf{C}_1$ , where  $\alpha$  is a  $90^\circ$  vertex. The edges (right edge and bottom edge of  $S_{LT}$ ), which would belong to  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , have been highlighted and directed to show their directions of traversal with  $\mathcal{R}$  always lying at the left

these are: (i)  $S_{RT}(i, j)$  and  $S_{LB}(i, j)$  contain object parts, and (ii)  $S_{LT}(i, j)$  and  $S_{RB}(i, j)$  contain object parts. An instance of arrangement (i) is shown in Fig. 4. It can be shown that, in each of these two arrangements,  $\alpha$  becomes a  $270^0$  vertex, since  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$  is considered to be non-self-intersecting, as stated in condition **(c1)** in Sec. 2. It should be carefully observed in Fig. 4 that two different styles of arrow heads indicate the two possibilities of formation of edges of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , since  $\alpha$  may be visited along either of the two possible paths during construction of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , and only the traced one is included in the final solution of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ .

For all other (four) arrangements with  $q = 2$ ,  $\alpha$  is just an ordinary point lying on some edge of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ .



**Fig. 4.** An instance of one of the 2 arrangements of case **C<sub>2</sub>**, where  $\alpha$  is a  $270^0$  vertex

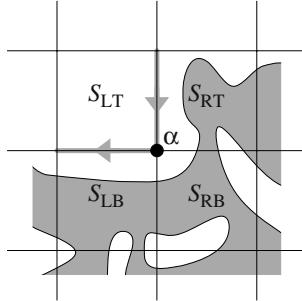
### Case C<sub>3</sub>:

If  $q = 3$ , then out of the four unit grid squares, only one square is free, which will have 4 different arrangements. In each of these arrangements, one of which is shown in Fig. 5,  $\alpha$  would be a  $270^0$  vertex (i.e. a vertex with  $270^0$  internal angle).

For the two other cases, namely case **C<sub>0</sub>** and case **C<sub>4</sub>**, it can be proved that  $\alpha$  can never be a vertex of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ . Case **C<sub>0</sub>** implies that  $\alpha$  is just an ordinary grid point that lies in  $\mathcal{I} \setminus \mathcal{R}$ , and can be shown to lie outside  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ . Case **C<sub>4</sub>** indicates that  $\alpha$  is a grid point that either lies in  $\mathcal{R}$ , or lies in a hole of  $\mathcal{R}$  and is surrounded by parts of  $\mathcal{R}$  in all four unit grid squares with common vertex  $\alpha$ , whence  $\alpha$  can be shown to be a grid point lying inside  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ .

### 3.2 Storing the Vertices of $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$

It is easy to see that there are two types of vertices of the isothetic polygon,  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ :  $90^0$  vertex (obtained from case **C<sub>1</sub>**), and  $270^0$  vertex (obtained from case **C<sub>2</sub>** and case **C<sub>3</sub>**), whose nature are discussed in Sec. 3.1. During the process of extraction of these vertices, each of them is dynamically inserted simultaneously in two temporary link lists,  $L_x$  and  $L_y$ , such that  $L_x$  always remains



**Fig. 5.** A typical instance of one of the 4 arrangements of case **C<sub>3</sub>**, where  $\alpha$  is a  $270^0$  vertex

lexicographically sorted in an increasing order w.r.t.  $x$  (primary key) and  $y$  (secondary key), and  $L_y$  always lexicographically sorted in an increasing order w.r.t.  $y$  (primary key) and  $x$  (secondary key). In addition to the grid coordinates of these vertices, one bit ( $\cdot, \cdot$ ) is stored for each vertex  $v$  in  $L_x$  and in  $L_y$  to denote its type ('0' denotes a  $90^0$  vertex and '1' denotes a  $270^0$  vertex). Furthermore, when the first  $90^0$  vertex ( $v_1^{(0)}$ ) is detected, the way in which its two edges should be traversed, such that the object  $\mathcal{R}$  lies left during traversal, is decided and stored accordingly. After extraction of all the vertices, the link lists  $L_x$  and  $L_y$  are processed, starting from any one of the  $90^0$  vertices, in order to construct the isothetic polygonal envelope,  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , as discussed in Sec. 3.3.

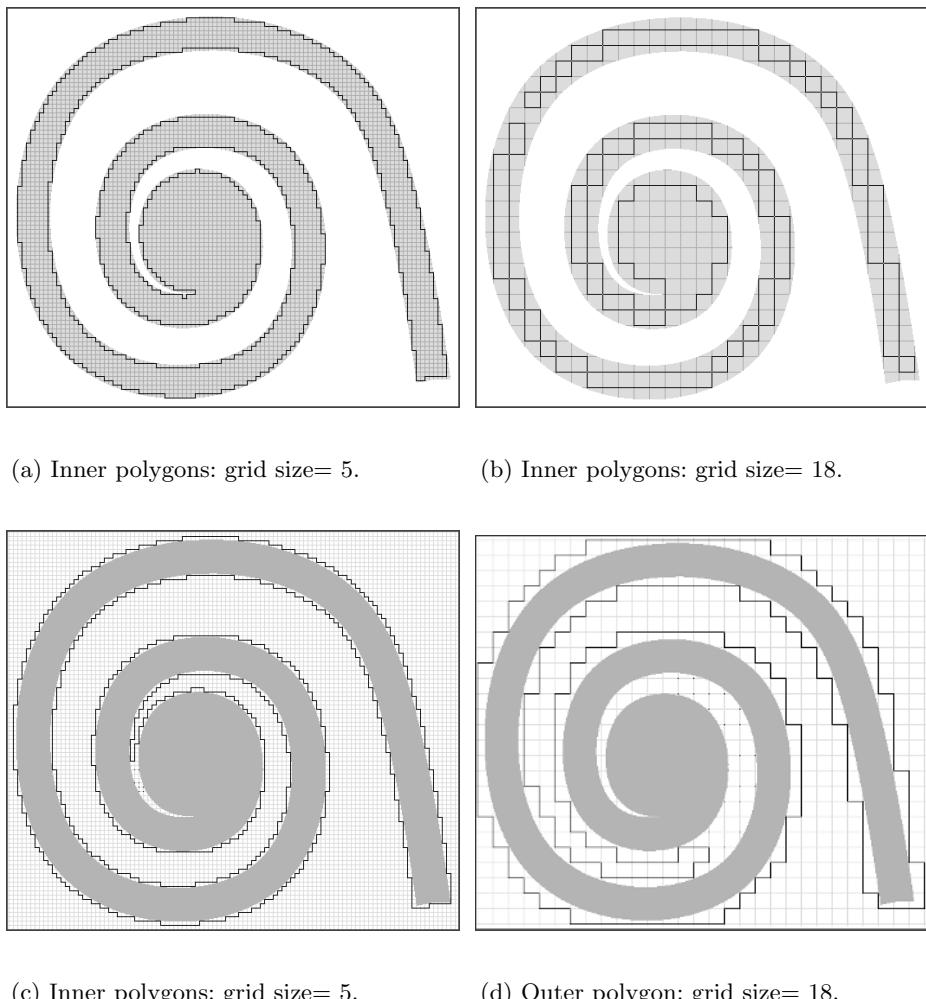
### 3.3 Construction of $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$

The construction of the isothetic polygonal envelope,  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , starts from  $v_1^{(0)}$ , which is the start vertex, as discussed in Sec. 3.2. It may be noted that, considering any  $270^0$  vertex as a start vertex is risky, since that vertex may be derived as a result of case **C<sub>2</sub>**, which has a dubious nature, as discussed in Sec. 3.1 and illustrated in Fig. 4. Further, if, for example, the isothetic envelope is merely a rectangle, then there exist  $90^0$  vertices and there does not exist any  $270^0$  vertex. Hence a vertex with internal angle  $90^0$  should always be considered as a start vertex.

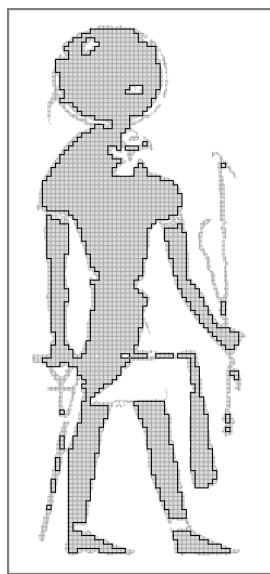
Now, if the outgoing edge from  $v_1^{(0)}$  is vertical and directed towards top (as shown in Fig. 3, then the preceding vertex in the list  $L_x$  is the next vertex,  $v_{next}$ , of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ , since the points in  $L_x$  are ordered w.r.t.  $x$ . Similarly, if the outgoing edge from  $v_1^{(0)}$  is vertical and directed towards bottom, then the succeeding vertex in the list  $L_x$  becomes the next vertex,  $v_{next}$ , of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ . For the other two possible arrangements, the decisions are similarly taken, and  $v_{next}$  is obtained using  $L_y$ . Once  $v_{next}$  is found, then depending on the type-bit ('0' or '1') of  $v_{next}$ , the outgoing edge from  $v_{next}$  is decisively selected using the rule that  $\mathcal{R}$  always lies left during traversal, and the process continues until the start vertex  $v_1^{(0)}$  is reached (after traversing the incident edge of  $v_1^{(0)}$  as the last edge of  $\mathcal{P}_{out}(\mathcal{R}, \mathcal{G})$ ).

## 4 Results

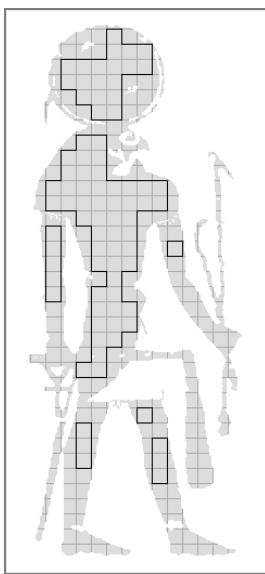
The above algorithm for constructing the outer polygon of any object in a 2D real plane has been tested on several objects of various shapes and sizes in 2D discrete plane. The algortihm requires slight modification for finding the inner polygons. By definition (Sec. 2), for a single object, we will always have a single outer polygon. However, if holes and self-intersections of outer polygon are allowed, the algorithm can be modified to produce the desired results. In the case of inner polygons, a single polygon may not be tight, i.e., complete “fill” the given object. Hence, we have allowed multiple inner polygons, whenever necessary. The CPU times on two typical sample images (shown in Figs. 6 and 7) have been given in



**Fig. 6.** Inner and outer polygons of a spiral



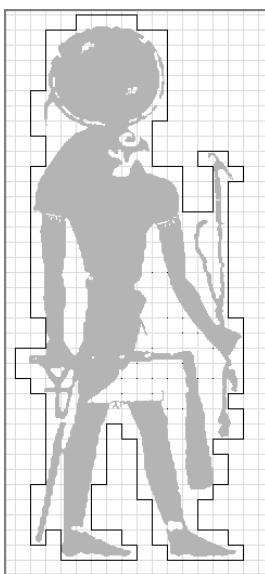
(a) Inner polygons: grid size= 4.



(b) Inner polygons: grid size= 14.



(a) Outer polygon: grid size= 4.



(b) Outer polygon: grid size= 14.

**Fig. 7.** Inner and outer polygons of a mythological figure

**Table 1.** CPU times in millisecs. for objects given in Figs. 6 and 7 for various grid sizes

Grid Size	CPU Time				
	spiral		myth. fig.		
	Inner	Outer	Inner	Outer	
1	895	687	1285	451	
2	235	201	280	203	
4	64	62	67	40	
8	15	20	13	15	
16	5	9	4	9	

Table 1 for varying grid sizes. From Table 1, it is evident that, there is a sharp decrease in CPU time with the increase in grid size.

## 5 Conclusion and Future Works

The proposed algorithm has been tested on several 2D objects in discrete domain, and has produced fast successful results in all cases. The proof of correctness of the algorithm is under preparation and will be reported in a subsequent paper. The algorithm will be tested on a non-uniform grid as a future work. Further, extension of the algorithm to 3D objects is also under our consideration.

## References

1. A. Bemporad, C. Filippi, and F. D. Torrisi, *Inner and outer approximations of polytopes using boxes*, Computational Geometry - Theory and Applications, Vol. 27, (2004) 151–178
2. L. Gatrell, *Cad-based grasp synthesis utilizing polygons, edges and vertices*, Proc. IEEE Intl. Conf. Robotics and Automation (1989) 184–189
3. Y. Kamon, T. Flash, and S. Edelman, *Learning to grasp using visual information*, Proc. IEEE Intl. Conf. Robotics and Automation (1995) 2470–2476
4. J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg, *Real-time Robot Motion Planning Using Rasterizing Computer*, Computer Graphics, ACM, Vol. 24(4), (1990) 327–335
5. M. Liu, Y. He, H. Hu, and D. Yu, *Dimension Reduction Based on Rough Set in Image Mining*, Intl. Conf. on Computer and Information Technology (CIT'04) (2004) 39–44
6. A. Morales, P. J. Sanz, and Á. P. del Pobil, *Vision-Based Computation of Three-Finger Grasps on Unknown Planar Objects*, IEEE Intl. Conf. on Intelligent Robots and Systems (2002) 1711–1716
7. S. C. Nandy and B. B. Bhattacharya, *Safety Zone Problem*, Journal of Algorithms, Vol. 37 (2000) 538–569
8. S. K. Pal and P. Mitra, *Case Generation Using Rough Sets with Fuzzy Representation*, IEEE Trans. on Knowledge and Data Engg., Vol. 16(3), (2004) 292–300
9. S. K. Pal and P. Mitra, *Pattern Recognition Algorithms for Data Mining*, Chapman and Hall/CRC Press, Bocan Raton, FL (2004)

# Approximate Steerability of Gabor Filters for Feature Detection

I. Kalliomäki and J. Lampinen

Laboratory of Computational Engineering,  
Helsinki University of Technology,  
P.O.Box 9203, FIN-02015 Espoo, Finland  
[{ilkka.kalliomaki, jouko.lampinen}@tkk.fi](mailto:{ilkka.kalliomaki, jouko.lampinen}@tkk.fi)

**Abstract.** We discuss the connection between Gabor filters and steerable filters in pattern recognition. We derive optimal steering coefficients for Gabor filters and evaluate the accuracy of the approximative orientation steering numerically. Gabor filters can be well steerable, but the error of the approximation depends heavily on the parameters. We show how a rotation invariant feature similarity measure can be obtained using steerability.

## 1 Introduction

Many different types of oriented filters have been proposed for low-level vision tasks. In this paper we consider only the suitability of filters to the task of generic visual feature detection. A central task in feature detection is to internally represent the local gray level structure of the image at specific fiducial locations in a robust and generic way. Of specific interest are systems such as the bunch graph matching model [12] which consists of local feature descriptors in a global graph structure, representing the current state of the art in face alignment and recognition.

Gabor filters are oriented filters which achieve the lower limit of joint uncertainty in spatial and frequency domains, described by the Heisenberg-Weyl (Gabor) uncertainty principle. In this sense they appear to be optimal for pattern recognition [2]. There exists also strong psychophysical evidence that mechanisms employing oriented linear filters are involved in mammalian vision, and they are well approximated with Gabor filters [1]. Despite being somewhat difficult to use in applications requiring reconstructions from the filter responses, Gabor filters can form a relatively good approximation of a tight wavelet frame and reconstructions using iterative methods are possible [7].

Steerable filters are a flexible and computationally efficient method to compute the response of an oriented filter in an arbitrary orientation [3]. They have found applications in many different low level vision tasks. The steerability property of the feature detectors facilitates strong rotational invariance, as orientation estimates can be computed from the filter responses. However, exactly steerable filters of the form  $P(x)G(\sqrt{x^2 + y^2})$ , where  $P(x)$  is a polynomial, such as the examples considered in [3] are less flexible, because orientation and radial frequency

selectivities of such filters cannot be easily chosen independently. Polar-separable oriented steerable filters were already proposed in [6] as generic image processing operators. Polar-separable oriented filters are typically constructed in frequency space, and have no closed-form expression in the spatial domain.

Using the theory of Lie groups, a steerable basis can be found for arbitrary parameter groups by representing or approximating the filters in an equivariant function space [8], [5]. For single-parameter 2D rotation expressed in polar coordinates  $(r, \theta)$ , this function space is  $\{f(r) \exp(ik\theta)\}, k \in \mathbb{Z}$ , i.e. complex harmonics together with an arbitrary (real-valued) radial component  $f(r)$  [10]. A viable approach for filter design is to start with an ideal filter prototype (for example, a Gabor filter) and approximate it in the appropriate equivariant function space, which is guaranteed to be closed under the same transformational group [11]. Before performing the approximation it is preferable to know the steering capability of the original Gabor filter.

## 2 Gabor Filters

The generic form of the Gabor filter with a non-spherical Gaussian envelope function is described by

$$f(\xi; \mu, S, R_\theta) = \frac{|\mu|^2}{\sqrt{\det(S)}} \exp\left(-\frac{|\mu|^2}{2}(R_\theta \xi)^T S^{-1} R_\theta \xi\right) \exp(i\mu^T R_\theta \xi) \quad (1)$$

where  $\xi = [x \ y]^T$  are the spatial coordinates, the wave vector  $\mu = [f_c \ 0]^T$  determines the center frequency  $f_c$  of the filter and also acts as a scaling factor in this parameterization,  $\theta$  determines the orientation of the filter via the rotation matrix

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (2)$$

and the covariance matrix

$$S = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (3)$$

determines the frequency bandwidths of the filter along the axes in Cartesian coordinates. Leaving the normalization term out, we can write the complex Gabor filter as a product of a Gaussian distribution modulated by a complex exponential,

$$f(\xi; \mu, S, R_\theta) \propto N(0, |\mu|^2 R_\theta^T S R_\theta) \cdot \exp(i\mu^T R_\theta \xi). \quad (4)$$

An important property of this parameterization is that the filters retain their shape regardless of rotation angle  $\theta$  and center frequency  $\mu$ .

The Fourier transform of a Gaussian function is also a Gaussian function (although no longer normalized), and modulation by complex plane wave corresponds to a shift from the origin in the Fourier plane by the amount described by  $R_\theta^T \mu$ . The rotation property of 2D Fourier transform states that rotations in the spatial plane correspond directly to rotations in the Fourier plane. As a

result, the Fourier transform of the complex Gabor filter is a single Gaussian function

$$\mathcal{F}\{f\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta). \quad (5)$$

We denote the real-valued even and odd Gabor filters with  $g = \text{Re}\{f\}$  and  $h = \text{Im}\{f\}$ , respectively, so that  $f = g + ih$ . Fourier transforms of real and imaginary parts of the complex filter are sums of two Gaussian functions,

$$\mathcal{F}\{g\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) \quad (6)$$

and

$$\mathcal{F}\{h\} \propto N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) - N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta). \quad (7)$$

The real and imaginary parts of the complex Gabor filter are approximately in quadrature, allowing independent analysis of signal amplitude and phase. Low values of the filter shape parameter  $\sigma_x$  cause a prominent DC component to the even filter, which makes the quadrature approximation inexact. In practice this means that the filter phase responses are directly dependent on the signal amplitude, which is usually undesirable. One way to remove DC component is to subtract it from the modulating complex exponential, giving the equation

$$N(0, |\mu|^2 R_\theta^T S R_\theta) \cdot \left( \exp(i\mu^T R_\theta \xi) - \exp\left(-\frac{\sigma_x^2}{2}\right) \right), \quad (8)$$

which is no longer exactly a Gabor filter. Because of brevity reasons we will not consider this filter type in this paper further, but only note that its behavior is highly similar to the results presented here for Gabor filters.

### 3 Steerability

Freeman originally derived the conditions for exact steerability by considering the Fourier series of the angular component of the filter in polar coordinates [3]. An alternative approach is to use linear algebra to find the optimal linear steering functions for an arbitrary set of filters [4], which we review here briefly.

Orientation steerability of a (real-valued) linear filter  $g$  means that arbitrary filter orientations can be computed (or at least approximated) by computing the sum of a set of basis filters  $\mathbf{g} = \{g_{\theta_1}, g_{\theta_2}, \dots, g_{\theta_N}\}$  weighted with steering coefficients  $\mathbf{k} = \{k_1(\theta), k_2(\theta), \dots, k_N(\theta)\}$ ,

$$g(\theta) \approx \sum_{j=1}^N k_j(\theta) g_{\theta_j} = \mathbf{k}^T \mathbf{g}. \quad (9)$$

In the case of complex-valued filters in quadrature, separate real-valued steering coefficients  $k_j(\theta)$  are needed for the real and imaginary parts of the filter. We assume that the basis filters  $g_{\theta_j}$  share the same shape parameters  $S$  and frequency  $\mu$ .

Let us define the inner product between real-valued functions  $u$  and  $v$  as  $\langle u, v \rangle = \int_{\omega \in R^2} u(\omega)v(\omega) d\omega$ . Let the functions  $u$  and  $v$  be normalized without

loss of generality so that  $\langle u, u \rangle = \langle v, v \rangle = 1$ . The optimal steering coefficients  $k$  can be solved analytically by minimizing the L2 norm of the error  $e = g(\theta) - \mathbf{k}^T \mathbf{g}$ ,

$$\begin{aligned} \arg \min_{\mathbf{k}} \|e\|^2 &= \arg \min_{\mathbf{k}} \langle g(\theta) - \mathbf{k}^T \mathbf{g}, g(\theta) - \mathbf{k}^T \mathbf{g} \rangle \\ &= \arg \min_{\mathbf{k}} \langle g(\theta), g(\theta) \rangle - 2 \langle g(\theta), \mathbf{k}^T \mathbf{g} \rangle + \langle \mathbf{k}^T \mathbf{g}, \mathbf{k}^T \mathbf{g} \rangle. \end{aligned} \quad (10)$$

The minimum of this expression is obtained by differentiating Eq. 10 and setting the result to zero, leading to the matrix equation

$$\mathbf{Gk} = \gamma \quad (11)$$

where the matrix  $\mathbf{G}$  and vector  $\gamma$  have the elements  $\mathbf{G}_{i,j} = \langle g_{\theta_i}, g_{\theta_j} \rangle$  and  $\gamma_i = \langle g(\theta), g_{\theta_i} \rangle$ , respectively. The vector of optimal steering coefficients  $\mathbf{k}$  can be solved using straightforward matrix inversion of  $\mathbf{G}$ , which is constant relative to the steering angle  $\theta$ . Another possibility is to compute the Singular Value Decomposition of  $\mathbf{G}$  and use it to calculate the SVD inverse. Unlike previous approaches using the SVD approach ([4], [9]), we compute the inner products  $u(\theta) = u(\alpha - \beta) = \langle g_\alpha, g_\beta \rangle$  analytically.

## 4 Inner Product of Rotated Gabor Filters

The inner product integral  $u(\theta)$  of two even Gabor filters in the frequency space is

$$\begin{aligned} \langle g, g_\theta \rangle &= \langle \mathcal{F}\{g\}, \mathcal{F}\{g_\theta\} \rangle = \int (N(\mu, S^{-1}) + N(-\mu, S^{-1})) \\ &\quad \cdot (N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta)) d\omega \end{aligned} \quad (12)$$

which, using a symmetry argument, becomes

$$= 2 \int N(\mu, S^{-1}) N(R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) + N(\mu, S^{-1}) N(-R_\theta^T \mu, R_\theta^T S^{-1} R_\theta) d\omega. \quad (13)$$

The inner product of two Gaussian functions (the normalization constant of a product of two Gaussian functions) is also Gaussian with respect to the parameters of the functions,

$$\langle N(a, A) \cdot N(b, B) \rangle \propto \sqrt{\frac{|C|}{|A||B|}} \exp\left(-\frac{(a^T A^{-1} a + b^T B^{-1} b - c^T C^{-1} c)}{2}\right), \quad (14)$$

with  $C = (A^{-1} + B^{-1})^{-1}$  and  $c = CA^{-1}a + CB^{-1}b$ . Applying this result gives after some manipulation

$$\langle g, g_\theta \rangle = \frac{\sqrt{|U|}}{Z_g} \exp\left(\frac{\nu^T (U + R_\theta^T U R_\theta) \nu}{2}\right) \cdot \cosh\left(-\frac{\nu^T (U R_\theta^T + R_\theta U) \nu}{2}\right) \quad (15)$$

where  $U = (S + R_\theta S R_\theta^T)^{-1}$ ,  $\nu = [\sigma_x^2 \ 0]^T$  and  $Z_g$  is a normalization factor. It is most conveniently computed by requiring that the inner product equals to one at  $\theta = 0$ , yielding the result  $Z_g = \frac{1}{2} \sigma_x^{-1} \sigma_y^{-1} \exp\left(\frac{1}{2} \sigma_x^2\right) \cosh\left(-\frac{1}{2} \sigma_x^2\right)$ .

Inner product function of two odd Gabor filters  $h$  and  $h_\theta$  is obtained similarly,

$$\langle h, h_\theta \rangle = \frac{\sqrt{|U|}}{Z_h} \exp\left(\frac{\nu^T(U + R_\theta^T U R_\theta)\nu}{2}\right) \cdot \sinh\left(-\frac{\nu^T(U R_\theta^T + R_\theta U)\nu}{2}\right), \quad (16)$$

with the normalization factor  $Z_h = \frac{1}{2}\sigma_x^{-1}\sigma_y^{-1} \exp\left(\frac{1}{2}\sigma_x^2\right) \sinh\left(-\frac{1}{2}\sigma_x^2\right)$ .

The inner product values define the elements of  $\mathbf{G}$  and  $\gamma$ , and optimal steering coefficients  $\mathbf{k}$  for an arbitrary steering angle  $\theta$  can be computed via the matrix multiplication  $\mathbf{k}(\theta) = \mathbf{G}^{-1}\gamma(\theta)$ .

## 5 Steering Error

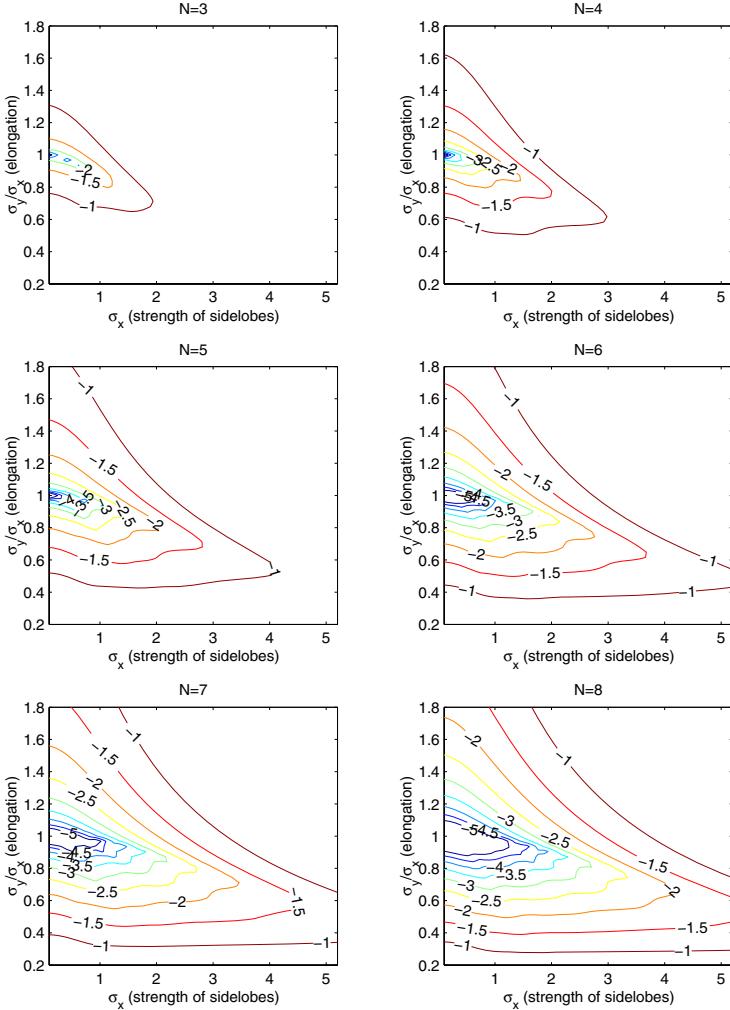
The steering property of Gabor-type filters is not exact, but only approximate. The goodness of the steering approximation depends heavily on the number of basis filters and shape parameters  $S$ . Let us define the measure for steering error by

$$E_s = \max_{\theta} \sqrt{\frac{\langle g(\theta) - \mathbf{k}(\theta)^T \mathbf{g}, g(\theta) - \mathbf{k}(\theta)^T \mathbf{g} \rangle}{\langle g(\theta), g(\theta) \rangle}}, \quad (17)$$

that is, the 2-norm distance of the maximum relative impulse response error. The same error measure was used in [4]. In an evenly spaced filter bank the maximum error occurs always exactly between known filter orientations, that is, if filters are in orientations  $\theta_i = \pi \frac{i}{N}$ ,  $i \in \{0, 1, \dots, N-1\}$ , maximum error is reached at  $\theta = \frac{\pi}{2N}$ . It is straightforward then to evaluate numerically the maximum steering error with different filter shape parameters  $S$ . Since we have separate steering functions  $\mathbf{k}(\theta)$  for even and odd filters of the quadrature pair, we define the total steering error as the average of the even and odd filter errors,

$$E_s^{avg} = \frac{E_s^{even} + E_s^{odd}}{2}. \quad (18)$$

The error behavior of Gabor filters in banks of three to eight filters is shown in Fig. 1. The overall effects of the filter shape parameters  $S$  are similar with any number  $N$  of basis filters. Because the angular bandwidth of Gabor filters depends nonlinearly on both  $\sigma_x$  and  $\sigma_y$ , there exists no single optimal value for the shape parameters, but rather only tradeoffs between approximate steerability, exactness of the quadrature pair and frequency resolution in the angular and radial domains. Steering error becomes progressively lower as more basis filters are present, and while the spatial domain sidelobes are not prominent. Most importantly, spherical Gabor filters are in general not optimal in terms of steering, and slightly flattened filters with  $\sigma_y/\sigma_x < 1$  have considerably better error performance. In other words, steerability is improved if the filters are less specific in the angular dimension than in the frequency dimension. This behavior is compatible with the properties of derivative of Gaussian filters, which are similarly flattened in the frequency space although their envelope function is a spherically symmetric Gaussian.



**Fig. 1.** Base-10 logarithm of steering error in impulse responses of Gabor filters. In typical applications 1% error (the -2 contour) may be considered acceptable

## 6 Rotation Invariant Feature Similarity

The Gabor transform  $G_I$  of the image  $I$  is defined by convolving the image with the filter bank  $\mathbf{f}$ ,

$$G_I = \mathbf{f} * I = \{f_{\theta_1} * I, f_{\theta_2} * I, \dots, f_{\theta_N} * I\}. \quad (19)$$

The  $J_{x,y} = G_I(x,y) = \{j_1, j_2, \dots, j_N\}$  of a feature is the Gabor transform vector at the feature location  $(x,y)$ . In feature detection, a  $\sim \sim \sim \sim \sim$  which assigns a numerical value for the similarity between two perceived

patterns, is needed. For example, the normalized inner product between the feature vectors consisting of filter responses can be used [12], defined as

$$S_1 = \frac{\langle J^{(1)}, J^{(2)} \rangle}{\|J^{(1)}\| \|J^{(2)}\|}. \quad (20)$$

A straightforward way to extend this similarity measure to be rotation invariant is to compute the inner product in all  $2N$  relative discrete rotation angles of the filter jets and choose the largest,

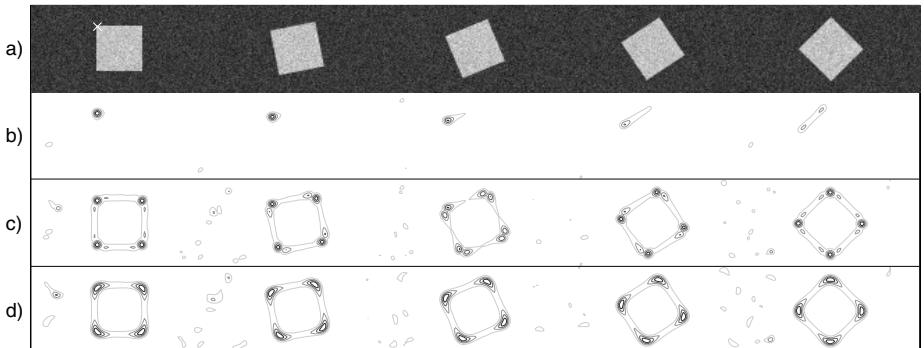
$$S_2 = \frac{1}{\|J^{(1)}\| \|J^{(2)}\|} \left( \max_{i=-N, \dots, N-1} \sum_{k=0}^{N-1} j_{k-i}^{(1)} j_k^{(2)} \right). \quad (21)$$

Here a negative index filter  $-i$  is interpreted as complex conjugate filter of the positive index  $N - i$  (this is because the response of the Gabor filter has equal amplitude and opposite phase when a 180 degree rotation is performed) and indices  $N + i, i \geq 0$  wrap around to  $-N + i$ .

This discrete scheme can be easily extended to continuous space by using steerability. Let us expand the discrete filter response  $J^{(1)}$  into a continuous one with rotation angle  $\phi$  using the steering coefficients  $\hat{k}_i(\phi)$ , so that the jet  $J^{(1)}$  consists of filter responses  $j_k^{(1)}(\phi) = \sum_i \hat{k}_i(\phi) j_{i+k}^{(1)}$ . Now we can compute the similarity between  $J^{(1)}$  and  $J^{(2)}$  in any relative orientation angle and choose the largest,

$$S_3 = \max_{\phi} \frac{1}{\|J^{(1)}\| \|J^{(2)}\|} \sum_k \left( \left( \sum_i \hat{k}_i(\phi) j_{i+k}^{(1)} \right) j_k^{(2)} \right). \quad (22)$$

The norm of the steered jet  $J^{(1)}$  is preserved exactly only if the steering is exact. Approximate steering causes slight variation in the norm, but normalization may



**Fig. 2.** Behavior of three different similarity functions with a synthetic test image *a*), with the test feature marked with a red cross. *b*) Normalized inner product similarity  $S_1$ . *c*) Discrete angle rotation invariant similarity  $S_2$ . *d*) Continuous angle rotation invariant similarity  $S_3$

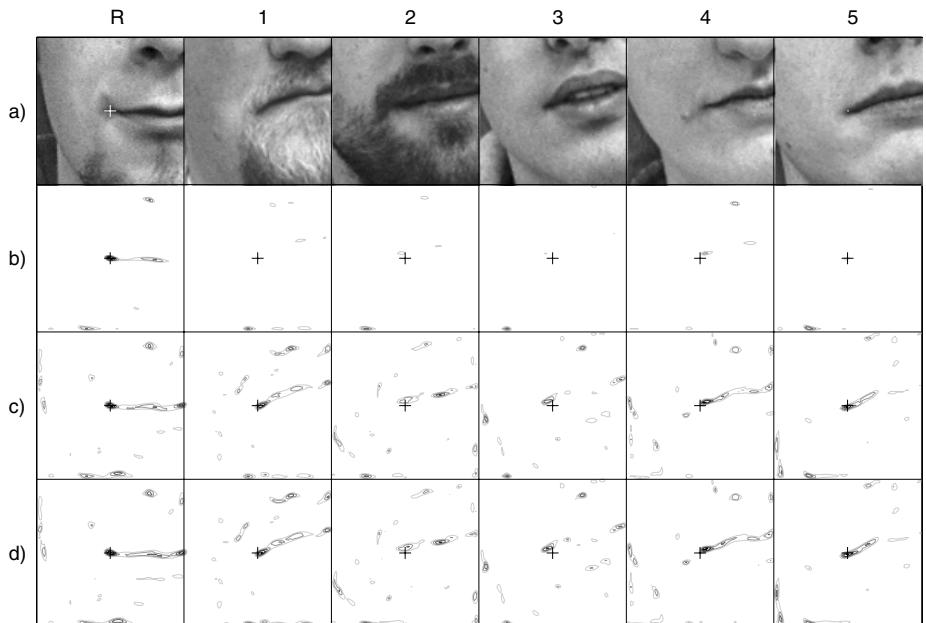
still be performed only after maximization to lessen the computational cost, if the maximum steering error is small.

We illustrate the behavior of the different similarity functions with two example images. Fig. (2) shows the similarity fields of a corner feature (marked with a white cross) in a synthetic test image, computed using Eqs. (20), (21) and (22). A filter bank with four orientations is used, with shape parameters  $\sigma = [2.5 \ 1.75]$ .

The similarity function  $S_1$  provides best localization of the correct feature, but withstands only small rotations. The similarity function  $S_2$  has generally an unequal response with respect to rotation, and only orientations which are present in the filter bank provide the correct response. The similarity function  $S_3$  has equal response in all orientation angles. However, as a consequence of rotation invariance, feature specificity is lower than with the two other measures, and the localization capability is worse.

The normalization factor  $\frac{1}{\|J^{(1)}\|\|J^{(2)}\|}$ , which is present in all three similarity measures, provides contrast invariance and causes medium similarity values to be found also among background noise.

Similarity functions are not highly sensitive to the filter shape parameter values, and while the filters used in this example do not retain the shape of their impulse responses very well under steering (error is approximately 6%), there



**Fig. 3.** Similarity of a reference mouth corner marked with a white cross, with five independent rotated test images. b) Normalized inner product similarity  $S_1$ . c) Discrete angle rotation invariant similarity  $S_2$ . d) Continuous angle rotation invariant similarity  $S_3$ . Even a small rotation in the image necessitates rotation invariance

is hardly any noticeable variation due to rotation in the similarity field of the function  $S_3$ .

Fig. (3) shows the similarity fields of  $S_1$ ,  $S_2$  and  $S_3$  of real-world face images. Here, eight filters with shape parameters  $\sigma = [5 \ 2.8]$  were used. The reference Gabor jet feature was obtained from the mouth corner of the leftmost face, marked again with a white cross. The five test images have been rotated 13 degrees, and the manually annotated feature locations are marked with crosses in their similarity fields. Eight basis filters are useful in making the features specific so that the mouth corners are recognized well, but without rotation invariance even small rotations cause the similarity to drop drastically, and detection is not possible. In contrast, similarity functions  $S_2$  and  $S_3$  provide very good feature localization, with a clear maximum near the annotated feature location in all test images. The differences between the performance of  $S_2$  and  $S_3$  are masked by the large variation occurring in natural images. The similarity measures alone do not suffice in solving the face alignment problem, and the ambiguities caused by feature variability have to be resolved by including the information in the relative locations of the detected features.

## 7 Conclusions

The main contribution of this work is to make the methods of steerable filters available when using Gabor filters. The advantages of Gabor filters are mainly that they have a simple analytical form, making the filter bank design problem easier, and that the filters can be tuned for different applications by adjusting the shape parameters.

We have shown that the methods for steerable filters are available also for some Gabor filters in applications which do not require exact steering. The steering error depends dramatically on the shape parameters, and non-spherical Gabor filters (with  $\sigma_x \neq \sigma_y$ ) have significantly better error performance than the more commonly used spherical Gabor filters.

Steerability allows a straightforward implementation of rotation invariance in the image plane. Without rotation invariance, features detected in high angular resolution become also very sensitive to their particular orientation. Rotation invariance removes this fundamental tradeoff, but makes the features also less specific using the same angular resolution. Demanding feature detection applications may require filter banks with high angular resolution. Our second example with natural face images implies that some form of rotation invariance is a necessity with such filters. The natural variation in the features of the test images is large enough to mask the errors of the discrete rotation invariance, and little additional gain is obtained with more accurate continuous rotation invariance.

## References

1. J. G. Daugman. Complete discrete 2-D gabor transformations by neural networks for image analysis and compression. *IEEE Trans. ASSP*, 36(7), 1988.
2. J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.*, 2(7):1160–1169, 1985.
3. W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. PAMI*, 13(9):891–906, 1991.
4. H. Greenspan, S. Belongie, P. Perona, R. Goodman, S. Rakshit, and C. Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *CVPR94*, pages 222–228, 1994.
5. Y. Hel-Or and P. C. Teo. Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9(1):83–95, 1998.
6. H. Knutsson, R. Wilson, and G. H. Granlund. Anisotropic Nonstationary Image Estimation and Its Applications: Part I—Restoration of Noisy Images. *IEEE Transactions on Communications*, COM-31(3):388–397, 1983.
7. Tai Sing Lee. Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.
8. M. Michaelis and G. Sommer. A Lie Group approach to steerable filters. *Pattern Recognition Letters*, 16(11):1165–1174, 1995.
9. G. Sommer, M. Michaelis, and R. Herpers. The SVD approach for steerable filter design. In *Proc. Int. Symposium on Circuits and Systems 1998*, volume 5, pages 349–353, Monterey, California, 1998.
10. P. C. Teo. *Theory and Applications of Steerable Functions*. Ph.D. thesis, Technical Report CS-TR-98-1604, March 1998.
11. P. C. Teo and Y. Hel-Or. Design of multi-parameter steerable functions using cascade basis reduction. *IEEE Trans. PAMI*, 21(6):552–556, 1999.
12. Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. In Gerald Sommer, Kostas Daniilidis, and Josef Pauli, editors, *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, number 1296, pages 456–463, Heidelberg, 1997. Springer-Verlag.

# Nonlinear Dimensionality Reduction Using Circuit Models

Fredrik Andersson and Jens Nilsson

Centre for Mathematical Sciences,  
Lund University/LTH,  
P.O. Box 118, S-22100 Lund, Sweden  
`{fa, jensn}@maths.lth.se`

**Abstract.** The problem addressed in nonlinear dimensionality reduction, is to find lower dimensional configurations of high dimensional data, thereby revealing underlying structure. One popular method in this regard is the Isomap algorithm, where local information is used to find approximate geodesic distances. From such distance estimations, lower dimensional representations, accurate on a global scale, are obtained by multidimensional scaling. The property of global approximation sets Isomap in contrast to many competing methods, which approximate only locally.

A serious drawback of Isomap is that it is topologically unstable, i.e., that incorrectly chosen algorithm parameters or perturbations of data may abruptly alter the resulting configurations. To handle this problem, we propose new methods for more robust approximation of the geodesic distances. This is done using a viewpoint of electric circuits. The robustness is validated by experiments. By such an approach we achieve both the stability of local methods and the global approximation property of global methods.

## 1 Introduction

The analysis of data sampled from manifolds embedded in higher dimensional spaces, is often referred to as manifold learning or nonlinear dimensionality reduction. Such situations arise frequently across diverse disciplines of science such as image analysis, signal processing, psychology and biology. The recent years have brought a growing interest in the development of methods for cases where data are sampled from curved manifolds and where standard, well-established methods like Principal Component Analysis (PCA) [10] and Multidimensional Scaling (MDS) [6] do not perform satisfactory.

... [12], ... [3] and ... [8] represent recent efforts to address such situations. Another example is the ... algorithm [13], which, adopting a graph-based approach, computes approximations of the geodesic distances on the manifold and uses them for producing lower dimensional representations of data. One problem that may arise in this process is that of topological instability — small perturbations on data points or minor changes in parameter

values may result in large changes in the constructed approximate distances; cf. [2]. In this paper we present algorithms for distance approximation that are more robust against issues of topological instability. We study the performance of the methods on the popular swiss roll data set and discuss the influence of different choices of parameter values on the results. We find that our proposed methods are substantially more robust than the Isomap algorithm while retaining good performance on a global scale.

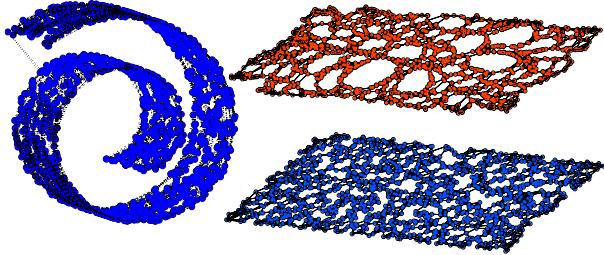
## 2 Manifold Learning

If  $\Phi$  is a linear isometric embedding, the standard methods of PCA and classical MDS are useful manifold learning tools. For more general mappings, other techniques are required. In [13] the ... algorithm is introduced, treating the case where  $\Phi$  is an isometry. Subsequently, ... [7] was proposed for conformal mappings.

Given a metric  $d_{\mathbb{R}^n}$  on  $\mathbb{R}^n$ , Isomap calculates approximations of the geodesic distances  $d_{\mathcal{X}}^{-1}$  on  $\mathcal{X}$ . An adjacency graph is constructed by connecting neighboring data points, where two points  $x_i$  and  $x_j$  are neighbors if either, given an  $\epsilon > 0$ ,  $d_{\mathbb{R}^n}(x_j, x_k) < \epsilon$ , or if, given an integer  $K$ , either one of  $x_j$  and  $x_k$ , has the other among its  $K$  closest points. As a next step, approximations  $d_{\text{ISO}}(x_j, x_k)$  of the geodesic distances  $d_{\mathcal{X}}(x_j, x_k)$  on  $\mathcal{X}$  are calculated by finding the shortest graph path between  $x_j$  and  $x_k$ . If the data point density is high enough,  $d_{\text{ISO}}$  approximates  $d_{\mathcal{X}}$  well. Since  $\Phi$  is an isometry,  $d_{\text{ISO}}$  also approximates the Euclidean distances  $d_{\mathcal{Y}}(y_j, y_k)$  in  $\mathcal{Y}$ . This motivates application of classical MDS to  $d_{\text{ISO}}(x_j, x_k)$ , yielding lower dimensional representations of data.

The performance of the Isomap algorithm depends on the density of data points; the curvature of the manifold  $\mathcal{X}$ ; the amount of noise; and the value of the neighborhood parameter ( $K$  or  $\epsilon$ ); cf. [4]. For improper parameter choices or in the presence of noise, shortcuts, not following the surface of the manifold, may appear in the graph, disturbing the ability of the algorithm to approximate geodesic distances; cf. Fig. 1. This effect is known as the problem of topological instability; cf. [2]. Another property of the Isomap algorithm is that it tends

<sup>1</sup> i.e., the distance travelled by an insect, taking the shortest path between two points on the manifold.



**Fig. 1.** To the left is shown 1500 uniformly distributed sample points ( $\hat{\mathcal{X}}_{1500}$ ) from a swiss roll manifold; the adjacency graph ( $K=10$ ) contains two shortcuts. To the right, the corresponding 1500 points in  $\hat{\mathcal{Y}}_{1500}$ , uniformly distributed in the plane (lower) and the corresponding Isomap projection (upper). The adjacency graph ( $K=5$ ) is drawn in the configurations. Random density fluctuations are amplified in the reconstruction

to cluster points in the resulting configurations. When the data point density on the manifold is finite, holes appear in the adjacency graph due to random fluctuations in local density, as illustrated in Fig. 1. For pairs of points on opposite sides of such holes the error term  $|d_{\text{ISO}} - d_{\mathcal{X}}|$  will become larger and consequently cause the holes to grow in the resulting Isomap projection. As a consequence, the projection might exhibit structures which are in fact amplifications of random fluctuations in data density. Note, however, that the clustering effect is of local character, in contrast to the error caused by topological instability.

Isomap and C-Isomap are examples of global manifold learning methods, attempting to reconstruct the configuration in  $\mathcal{Y}$  correctly on all scales. Local methods, on the other hand, attempt to create lower-dimensional representations that conserve similarities between nearby points only. Examples of local methods include [12], [3] and [8]. Similarly to Isomap, the Laplacian Eigenmaps algorithm constructs an adjacency graph, connecting neighboring points, as an initial step. The graph edges are then given weights  $w_{jk} = e^{-d_{\mathbb{R}^n}(x_j, x_k)^2/\sigma^2}$ , for some coupling parameter  $\sigma$ . Let  $W$  be the  $m \times m$  matrix containing the elements  $w_{jk}$ ,  $B$  be a diagonal matrix defined by  $B_{jj} = \sum_{k \neq j} w_{jk}$ , and let  $A = W - B$ , i.e., the . . . . . Briefly, the method of Laplacian Eigenmaps finds reconstruction embeddings by solving the generalized eigenvalue problem  $As = \lambda Bs$ .

### 3 Circuit Models for Distance Measures

Suppose that data points  $x_j \in \mathcal{X}$ ,  $j = 1, \dots, m$ , are given, along with a distance metric  $d_{\mathbb{R}^n}$ . The aim of this paper is to construct methods to approximate the geodesic distance  $d_{\mathcal{X}}(x_j, x_k)$  based only on the given data. Once good approximations are found, methods of MDS can be used to construct global methods for manifold learning. To this end, we employ models of electrical circuits, and use

the dynamics of such to construct new distance approximations. This approach offers a framework for the construction and understanding of the models, which are reminiscent of the methods based on the graph Laplacian.

### 3.1 A Passive Component Framework

Associate each data point  $x_j$  with a node  $n_j$  in an electrical circuit. To each node, attach a capacitor with capacitance  $c_j$  to earth. Neighboring<sup>2</sup> node pairs  $\{n_j, n_p\}$  are connected to each others by resistors  $r_{jp}$ , as in Fig. 2.a.

As a charge distribution is applied to the network, currents will move the charges until an equilibrium is reached, at which the voltages over the capacitors are all equal. The dynamic process of charging the capacitors will form the basis in our models for manifold learning, where we use charge times to construct new distance measures.

Denote the voltage over capacitor  $c_j$  by  $v_j(t)$ , let  $i_{jk}(t)$  denote the current from node  $n_j$  to  $n_k$ , and let  $i_{c_j}(t)$  denote the current from  $n_j$  to  $c_j$ . According to the Kirchhoff current law,

$$i_{c_j}(t) + \sum_{k=1, k \neq j}^m i_{jk}(t) = 0. \quad (1)$$

The current between two nodes  $n_j$  and  $n_k$  is given by

$$i_{jk}(t) = \frac{v_j(t) - v_k(t)}{r_{jk}}, \quad (2)$$

and for the current  $i_{c_j}$  the relation

$$c_j \frac{dv_j}{dt} = i_{c_j}(t), \quad (3)$$

holds true. By applying (2) and (3) we can rewrite (1) into

$$c_j \frac{dv_j}{dt} = \sum_{k=1, k \neq j}^m \frac{v_k(t)}{r_{jk}} - v_j(t) \sum_{k=1, k \neq j}^m \frac{1}{r_{jk}}. \quad (4)$$

Let  $V(t)$  be a column vector containing the elements  $v_j(t)$ ,  $j = 1, \dots, m$ . Then (4) can be expressed, in matrix form, as

$$C \frac{dV(t)}{dt} = AV(t), \quad (5)$$

where

$$A(j, k) = \begin{cases} -\sum_{k \neq j} r_{jk}^{-1}, & \text{if } j = k; \\ r_{jk}^{-1}, & \text{otherwise.} \end{cases}, \quad (6)$$

---

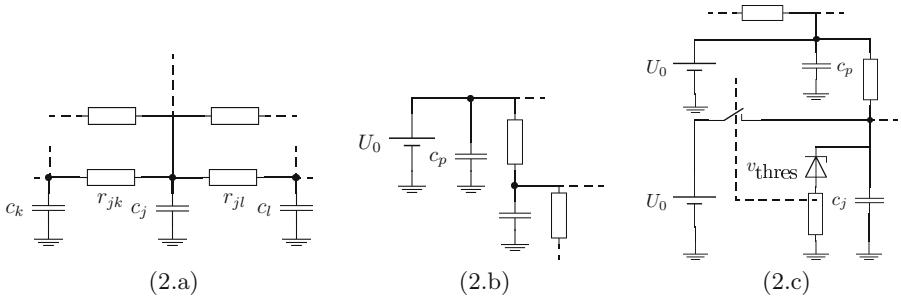
<sup>2</sup> with neighborhoods defined as in the Isomap algorithm, see Sect. 2.

and  $C$  is a diagonal matrix containing the capacities  $c_j$ ,  $j = 1, \dots, m$ . The solution to (5) is then given by

$$V(t) = e^{C^{-1}At}V_0, \quad (7)$$

where  $V_0$  is a column vector containing the initial voltages. It can be verified that  $C^{-1}A$  is negative semi-definite.

Typically, we choose the resistances  $r_{jk}$  to depend Gaussian on the distances  $D_{jk} = d_{\mathbb{R}^n}(x_j, x_k)$ , i.e.,  $r_{jk} = e^{d_{jk}^2/\sigma^2}$ , for some parameter  $\sigma$ . The object  $A$ , with this choice of resistances, is the same as the graph Laplacian of Laplacian Eigenmaps; cf. Sect. 2.



**Fig. 2.** Circuit models; a) The basic RC-circuit, b) The RC-circuit with constant voltage source, c) RC-unit with a voltage source connected through a zener diode switch

### 3.2 The RC and RCZ Models

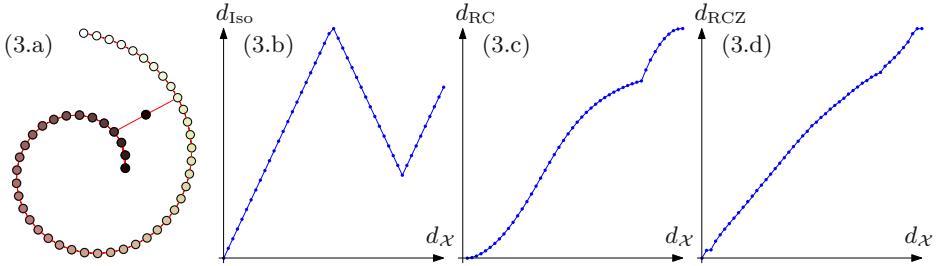
Instead of the basic model with charges diffusing from an initial state, we adopt a model where one of the nodes,  $n_p$ , is attached to a constant voltage source, as illustrated in Fig. 2.b. The voltages at the other nodes will then monotonically increase and reach the battery voltage at infinity. By setting the capacity  $c_p$  to infinity, we may use equation (5) to model the behavior of the circuit. The solution of the system is, analogously to (7), given by  $V = e^{C^\dagger At}V_p$ , where  $V_p(j) = \delta_{jp}$ , and  $C^\dagger(j,k) = c_j^{-1}$  if  $j = k \neq p$  and 0 otherwise. There are several ways to define the distance between points based on the voltages  $V$ . A straightforward way is to use the time it takes for a node  $n_j$  to reach a certain voltage level  $0 < v_{\text{thres}} < U_0$ , with a typical value  $v_{\text{thres}} = U_0/2$ . To assure symmetry we use the mean value of the time it takes to charge node  $n_j$  given a constant voltage source at node  $n_p$  and vice versa. These charge times need not fulfil the triangle inequality, why our dissimilarity measure is not necessarily a metric. This, however, does not disqualify its usefulness. For instance, the well known Mahalanobis measure is not metric; cf. [5]. In what follows, we will refer to the model above as the RC model, and denote the corresponding dissimilarity measure by  $d_{\text{RC}}$ . More sophisticated dissimilarity measures could, e.g., be constructed based on statistical methods; cf. [9, 11]. For our purposes, and in particular for what follows, the simpler variant is preferable.

Consider a set of samples  $\{x_i\}$  from a spiral  $\mathcal{X}$ , to which an outlier point is added, yielding a shortcut in the adjacency graph (Fig. 3.a.). Applying some global dimensionality reduction method, we would like the distances  $d_o(x_i, x_j)$  estimated by the particular method to be closely related to the geodesic distance  $d_{\mathcal{X}}(x_i, x_j)$  along the spiral. Ideally, there should be a linear relation,  $d_o = ad_{\mathcal{X}}$ . A more realistic criteria, though, would be to require a monotonically increasing relation  $d_o = f(d_{\mathcal{X}})$ . Panels b–d of Fig. 3 display distances from the outmost point in the spiral, computed using different methods, plotted against the true geodesic distances. Disturbed by the graph shortcut, Isomap fails to compute distances that relate monotonically to  $d_{\mathcal{X}}$  (Fig. 3.b). The RC method, on the other hand, succeeds with this despite the the shortcut (Fig. 3.c). However a shortcut effect still remains, and moreover, the relation between  $d_{RC}$  and  $d_{\mathcal{X}}$  deviates from a linear one. Points close to the source node are charged at a higher rate and during the time it takes for faraway points to reach threshold potential some charge will have trickled through the shortcut, thus explaining the observed shortcut influence. To avoid this, we introduce a model in which the nodes are charged through a moving front. In this model we connect nodes directly to the voltage source once the corresponding voltage reaches the threshold level  $v_{\text{thres}}$ . At this instance, we say that the node reaches . . . . , and we use the time it takes for the nodes to reach on-states as distance measure. In this way, nodes neighboring the front are charged directly by their neighbors, in contrast to being charged indirectly (through points in between) from the original source point. Electronically, we implement the moving front model by replacing the basic RC-unit with a slightly more sophisticated one; cf. Fig. 2.c. Each node is now equipped with a zener diode and a current controlled switch. As the voltage over the capacitor reaches  $v_{\text{thres}}$ , the zener diode moves into a conductive phase, turning the switch on, and the node enters the on-state. We refer to this model as the RCZ model, and denote the corresponding dissimilarity  $d_{\text{RCZ}}$ .

The purpose of the RCZ model is to make distances more uniformly distributed, since points far from the original source node sooner will reach on-state. Indeed, applying the RCZ model on the spiral data results in distances scaling linearly with  $d_{\mathcal{X}}$  showing just a minor influence from the shortcut (Fig. 3.d). For details on the numerical implementation of the RCZ model, see [1].

## 4 Results and Discussion

When both the coordinate space  $\mathcal{Y}$  and the input space  $\mathcal{X}$  are known (or, specifically,  $\hat{\mathcal{Y}}_m$  and  $\hat{\mathcal{X}}_m$  are given), we may evaluate the performance of dimensionality reduction methods by comparing the reconstructed configurations  $\hat{\mathcal{Z}}_m$  with  $\hat{\mathcal{Y}}_m$ . A reasonable requirement on any algorithm is that it performs well on affine subspaces. One way of verifying this requirement, is to apply the method using the embedding coordinates,  $\hat{\mathcal{Y}}_m$ , as input. If  $\Phi : \mathcal{Y} \longrightarrow \mathcal{X}$  is an isometry, the results should be identical when applying the algorithm to the real input coordinates  $\hat{\mathcal{X}}_m$  as when applying it to  $\hat{\mathcal{Y}}_m$ . Thus, the performance evaluation can

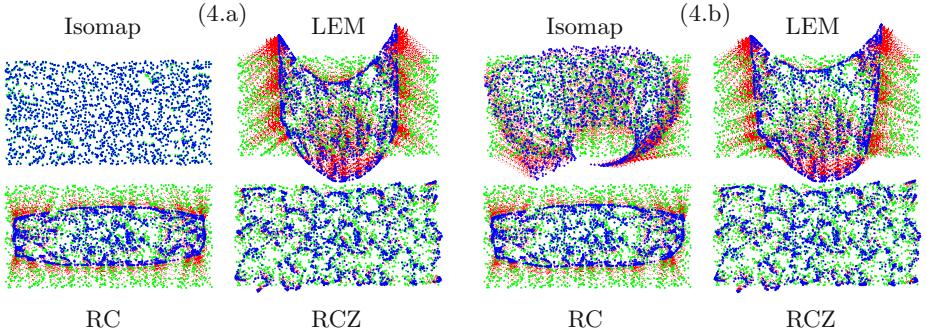


**Fig. 3.** Panel a shows perturbed spiral data. The remaining plots show the distances from the upper endpoint to the other points. Isomap distances (3.b) are severely distorted , while the RC (3.c) and the RCZ (3.d) distances deviate less from the geodesic

be divided with respect to two criteria — viewed as an operator, the method should mimic the identity operator when applied to  $\hat{\mathcal{Y}}_m$ , and the results when applied to  $\hat{\mathcal{X}}_m$  should not deviate too much from the results when applied to  $\hat{\mathcal{Y}}_m$ .

When comparing point configurations, only their actual shapes are interesting. Hence, we remove issues of scale, rotation, reflection and translation by fitting the configurations optimally (in a least square sense) to each other using such transformations. Subsequently, the root mean square (RMS) error is taken as a measure of similarity,  $E_{\text{RMS}} = \sqrt{\sum_{i=1}^m |z_i - y_i|^2/m}$ . In this section, we consider a set of 2000 data points, randomly sampled from a swiss roll manifold (cf. Fig. 1). The swiss roll data set contains coordinates for points both in  $\mathcal{Y} \subset \mathbb{R}^2$  and on  $\mathcal{X} \subset \mathbb{R}^3$ , and is therefore a suitable test set. The coordinate space of the swiss roll is a rectangle in the plane, so dimensionality reduction methods should ideally retrieve this structure. We study the performance of Isomap, Laplacian Eigenmaps, RC and RCZ applied to this data set.

First, we consider the case where the data points coincide with the embedding coordinates, i.e., when  $\hat{\mathcal{X}}_{2000} = \hat{\mathcal{Y}}_{2000} \subset \mathbb{R}^2$ . Concerning the parameters, we choose  $K = 14$ , a value where all four methods work well, and we choose the  $\sigma$ -values optimally for the Laplacian Eigenmaps and RCZ methods:  $\sigma_{\text{LEM}} = \infty$ ,  $\sigma_{\text{RCZ}} = 3\bar{d}$ , where  $\bar{d}$  is the average length in the adjacency graph, and use  $\sigma_{\text{RC}} = \sigma_{\text{RCZ}}$ . The resulting configurations are fitted to the embedding coordinates using similarity transformations, as described above, and the RMS errors of the respective reconstructions are calculated. Figure 4 a shows the embedding coordinates configurations ( $\hat{\mathcal{Y}}_{2000}$ ) by green/lighter points with the fitted reconstructions ( $\hat{\mathcal{Z}}_{2000}$ ) represented by blue/darker points, and with dotted (red) lines connecting the corresponding points in the configurations. On a global scale, Isomap and RCZ perform well while Laplacian Eigenmaps and RC suffer from some systematic errors. The RC reconstruction has an incorrect scale relation between the axes while Laplacian Eigenmaps produces a skewed reconstruction. Considering the local error structures, a slight clustering effect can be noticed in the Isomap reconstruction, while this effect is stronger in the other recon-



**Fig. 4.** Reconstructions with error structure: (a) for the planar case where the input coordinates and the embedding coordinates coincide; and (b) for the swiss roll case where the input coordinates are sampled from the swiss roll

structions. Clearly, Isomap produces the best reconstruction as also confirmed;  $E_{\text{RMS}}^{\text{Iso}} = 0.34$ ,  $E_{\text{RMS}}^{\text{LEM}} = 14$ ,  $E_{\text{RMS}}^{\text{RC}} = 6.8$ , and  $E_{\text{RMS}}^{\text{RCZ}} = 3.0$ .

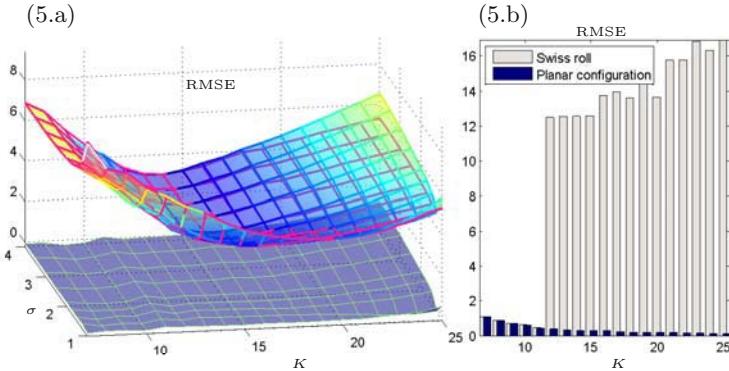
Laplacian Eigenmaps does not aim for globally correct reconstructions, hence the global distortion is not surprising here. The RC global error is less severe and parts of it is presumably due to the nonlinear relation between geodesic and approximated distances, as discussed in Sect. 3. The punishment of long distances with a higher resistance in the RC methods leads to the increased sensitivity to local clustering errors. Since Laplacian Eigenmaps and the RC methods rely on similar equations, Laplacian Eigenmaps also exhibits these effects.

Next, we consider the point configuration  $\hat{\mathcal{X}}_{2000}$  on the swiss roll manifold. Reconstructed embedding coordinates  $\hat{\mathcal{Z}}_{2000}$  are computed using input coordinates  $\hat{\mathcal{X}}_{2000} \subset \mathbb{R}^3$  on the swiss roll, and compared with  $\hat{\mathcal{Y}}_{2000}$  (Fig. 4.b). The largest difference compared to the previous analysis occurs in the Isomap reconstruction where the two ends of the rectangle have become connected. For the other three methods, however, the differences are small. In other words, the Laplacian Eigenmaps, RC and RCZ reconstructions are reasonably invariant under  $\Phi$  as required, while the Isomap reconstruction is not. The RMS errors are in this case  $E_{\text{RMS}}^{\text{Iso}} = 13$ ,  $E_{\text{RMS}}^{\text{LEM}} = 13$ ,  $E_{\text{RMS}}^{\text{RC}} = 6.4$ , and  $E_{\text{RMS}}^{\text{RCZ}} = 3.0$ , respectively.

Obviously, the results depend on the parameter values. For example, with  $K=10$ , no shortcuts appear in the adjacency graph on the swiss roll, and Isomap correctly reconstructs the rectangular configuration. In order to thoroughly investigate the method performances over a range of parameter values, we apply the algorithms using various  $\sigma \in [\bar{d}, 4\bar{d}]$  and  $K = 7, \dots, 25$ . Figure 5.a displays the approximation error for RCZ applied to the swiss roll (transparent surface), and the coordinate space rectangle (wire-frame mesh). Further, the difference between these two error matrices is shown by the lower surface plot, giving an idea of the fraction of error stemming from the geometrical change. The coordinate space error has a local minima around  $K = 14, \sigma = 3\bar{d}$ . The difference between the two errors behaves more or less monotonic with increased

$K$  and  $\sigma$ . Figure 5.b shows the error  $E_{\text{RMS}}^{\text{Iso}}$  of the Isomap reconstruction over different  $K$  for the rectangle and the swiss roll respectively. A sharp increase in error for the swiss roll appears at  $K=12$ , where shortcuts first appear in the graph.

The error for the rectangle configuration is highest at low  $K$  and low  $\sigma$ , where the local clustering effect is strongest — a low  $K$  gives a higher probability of holes in the adjacency graph, while a low  $\sigma$  gives a stronger punishment of long distances. At intermediate parameter values, the error is low, while it increases at higher  $K$  and  $\sigma$ . This slightly surprising behavior might be partly explained as an edge effect — at the edges the density of graph edges will be higher, causing a tendency of constricting points along the rectangle edge. This effect grows with  $K$  and  $\sigma$ . The fact that the residual error between the rectangle and the swiss roll grows with  $K$  and  $\sigma$  is explained by the increasing risk of shortcuts at larger  $K$  and the decreasing capability of down-weighting them at larger  $\sigma$ .



**Fig. 5.** a)RCZ: RMS errors for the planar and swiss roll configurations, illustrated by the colored surface and red wire-frame, respectively, with their difference, displayed by the lower surface; b) Isomap: RMS errors for the planar and swiss roll configurations

The results from the swiss roll data set illustrate that the Isomap algorithm is the most accurate, both on local and global scales — it works, that is. Due to topological instability, it is less robust than the other methods. Being a local method, Laplacian Eigenmaps is more stable than Isomap but does not control the global correctness. Furthermore, it suffers from a larger local clustering error — a problem shared with the RC and RCZ methods. Because of their similarity with Laplacian Eigenmaps, we may view the RC methods as being akin to Laplacian Eigenmaps with global control added. Compared to Isomap we may regard the RC methods as more robust relatives who pay for the increased robustness with a larger local error.

## 5 Conclusions

The present work addresses an alternative way of computing distances to avoid the problem of topological instability which the Isomap algorithm suffers from. These distances are obtained from solutions of ordinary differential equations, inspired by models of electric circuits. We demonstrate performance on typical data sets and discuss the relations between choice of parameter values and performance for the proposed and competing manifold learning methods, specifically under conditions of sparse or noisy data. The experiments show that the proposed methods may be seen as 'robustizations' of Isomap or 'globalizations' of Laplacian Eigenmaps.

## Acknowledgments

The authors would like to thank Per Broberg and Magnus Fontes. The research of J.N. was partially supported by the Swedish Knowledge Foundation (KK-stiftelsen), and AstraZeneca.

## References

1. Andersson, F., Nilsson, J.: Circuit models for manifold learning. Tech. rep., Lund University (2005)
2. Balasubramanian, M., Schwartz, E. L., Tenenbaum, J. B., de Silva, V., Langford, J. C.: The Isomap algorithm and topological stability. *Science* **295** (2002)
3. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation* **15** (2003) 1373–1396
4. Bernstein, M., de Silva, V., Langford, J. C., Tenenbaum, J. B.: Graph approximations to geodesics on embedded manifolds. Tech. rep., Stanford University (2000)
5. Chatfield, C., Collins, A. J.: Introduction to multivariate analysis. Chapman & Hall, London (1980)
6. Cox, T. F., Cox, M. A. A.: Multidimensional scaling. Vol. 59 of Monographs on Statistics and Applied Probability. Chapman & Hall, London (1994)
7. de Silva, V., Tenenbaum, J.: Global versus local methods in nonlinear dimensionality reduction. *Neural Information Processing Systems* **15** (2003), 705–712
8. Donoho, D., Grimes, C.: Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA* **100**(2003) 5591–5596
9. Ham, J., Lee, D. D., Mika, S., Schölkopf, B.: A kernel view of the dimensionality reduction of manifolds. In: ICML '04: Twenty-first international conference on Machine learning. ACM Press (2004)
10. Jolliffe, I. T.: Principal component analysis. Springer-Verlag, New York (2002)
11. Lafon, S.: Diffusion maps and geometric harmonics. Doctorate thesis, Yale University (2004)
12. Roweis, S. T., Saul, L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **290** (2000) 2323–2326
13. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290** (2000) 2319–2322

# Mapping Perceptual Texture Similarity for Image Retrieval

Janet S. Payne<sup>1</sup> and John Stonham<sup>2</sup>

<sup>1</sup> Centre for Applied Computing,

Faculty of Technology,

Buckinghamshire Chilterns University College,

High Wycombe, HP11 2JZ, UK

[janet.payne@bcuc.ac.uk](mailto:janet.payne@bcuc.ac.uk)

<http://www.bcuc.ac.uk/>

<sup>2</sup> Dept of Electronic and Computer Engineering,

Brunel University,

Uxbridge, UB8 3PH

[john.stonham@brunel.ac.uk](mailto:john.stonham@brunel.ac.uk)

<http://www.brunel.ac.uk/>

**Abstract.** Images are being produced and made available in ever increasing numbers; but how can we find images "like this one" that are of interest to us? Many different systems have been developed which offer content-based image retrieval (CBIR), using low-level features such as colour, texture and shape; but how can the retrieval performance of such systems be measured? We have produced a perceptually-derived ranking of similar images using the Brodatz textures image dataset, based on a human study, which can be used to benchmark retrieval performance. In this paper, we show how a "mental map" may be derived from individual judgements to provide a scale of psychological distance, and a visual indication of image similarity.

## 1 Measuring the Effectiveness of Image Retrieval

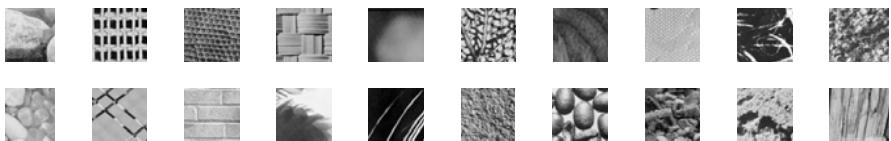
Digital images are being produced at a greater rate than ever before; digital cameras and cameraphones are widely available, and widely used. Art collections such as St Petersburg's Hermitage Museum or London's National Gallery, and photographic archives, continue to be digitised and made available via the Web. But how can we find images of interest to us among the petabytes of visual data? We have become used to efficient retrieval of text and web pages using keywords; but image retrieval remains a difficult task, in spite of the many research efforts applied over the past decade or more. Images for professional and technical uses, such as medical data, have standards that specify the metadata to be attached to each image [3]; however, the original annotator is unlikely to have anticipated all future potential uses of the images. Pictures taken for personal interest may also be of interest to others, both at the time, and for historical reasons in the future; experience shows that these are even less likely to have any kind of meaningful metadata [2], as a glance at the "properties" of any document file will demonstrate. Content-based image retrieval (CBIR) appears to be

essential, using low-level image features rather than relying on annotation to locate more images "like this one". Colour, texture and shape have all been used for this purpose, from IBM's QBIC onwards [7]; texture in particular has been used successfully for many different applications [3, 11].

### 1.1 How Should Similarity Be Judged for Images?

But how should image similarity be judged? Various measures for similarity of images have been proposed; these are typically based on distance measures in psychological space. Euclidean distance is perhaps the most widely used. However, Tversky (1977) criticizes the use of standard metrics, pointing out that similarity is often not symmetric [12]. As he points out, "a is like b" is generally not equivalent to "b is like a"; the more salient stimulus tends to be selected as the prototype or referent. Among the examples he used, geometric figures were considered to be more salient the better the "form" (regularity of outline); and where two figures were similar for this, the more complex one was considered to be more salient. We observed this asymmetry of similarity in this research, as figures 4 and 5 show.

Surely CBIR is intended for use by people interested in what the images resemble, rather than in learning a specialized database query language or an esoteric coding scheme. It seems appropriate to search on what the images actually "look like" to the potential users of such systems. In order to compare the performance of different CBIR systems, some standard image dataset is required. For texture, the most widely used one is still the Brodatz photographic album "Textures" [1], as used for example by Nunes *et al* [8]. Some example tiles from the Brodatz album are shown in fig. 1. Typically, a retrieval technique is judged by how many other tiles of the same texture it returns when presented with a tile from each texture as a query image [6]. This fails to take into account images which are not homogenous, nor does it consider different textures which the human observer would consider to be similar. In our research, we have developed a perceptually-derived ranking of similar images from the Brodatz dataset which can be used to evaluate retrieval performance [10]. Earlier studies of perceptual similarity have tended to use only a subset of the Brodatz textures; we have used the full set of 112 images.



**Fig. 1.** Examples from the Brodatz album

## 2 Texture Similarity Derived from Perceptual Judgements

The participants in our study were asked to select up to four images, in order, which they considered most like each of the full set of the 112 Brodatz images, viewed on a

computer screen [9]. The display showed 15 textures, arranged as three rows of five images, with the query image placed in the middle. Thirty people took part in this study, aged between 18 and 54, with a median age of 26; ten were female, twenty male. They included an art historian, a chemist, several computing students, a physicist, a psychology student, and a zoologist. It was noted that individuals tended to agree on similarity for regular textures such as reptile skin (eg D3, or D22), or pebbles (eg D30), or fine textures such as D4. Irregular textures such as D62, or D87, showed less agreement. Although individuals were free to select up to four "most like" images from the set displayed on the screen, it would have been surprising if there had been no agreement between them. Correlation measures this, and can be used to calculate the principal components of the responses.

For example, consider the selections made for D30 (table 1): a value of **5** is assigned to the query image, **4** to each individual's first choice, **3** to their second choice, and so on. In this case, only seven other images out of the possible fourteen were considered to be similar by the participants. Most people made four choices, but not everyone did; for example, the second participant made only one selection.

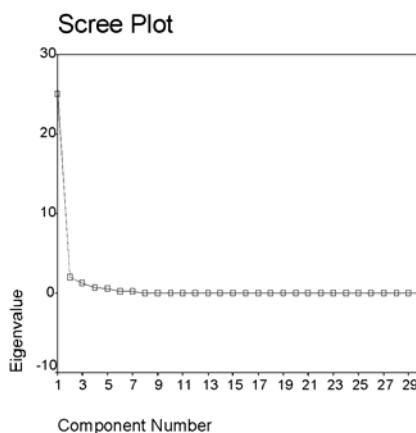
**Table 1.** Selections for D30 as query, by participant

D23	3	0	3	3	3	3	4	3	4	3	3	3	3	3	3	3	3	1	2	3	4	0	2	1	4	3	0	2	3
D27	1	0	1	0	2	1	2	1	1	0	1	0	0	0	0	1	1	3	0	1	2	0	1	2	1	1	0	1	2
<i>Query</i>	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
D31	4	4	4	4	4	4	3	4	2	4	4	4	4	4	4	4	4	2	4	4	4	3	4	4	4	2	4	4	4
D62	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	3	3	0	0	0	2	0	1
D66	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
D98	2	0	2	2	1	0	0	2	3	2	2	2	2	2	2	2	0	2	3	2	1	0	0	3	3	2	1	3	0
D99	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	3	0	0

## 2.1 Principal Components of the Perceptual Ranking

Principal components analysis (PCA), also known as the Karhunen-Loéve transform, is a form of data reduction; a mathematical procedure for transforming a number of (possibly) uncorrelated variables into a smaller number of uncorrelated variables, known as principal components [5]. The first principal component accounts for as much of the variability in the data as possible, and each following one for as much of the remaining variability as possible. Each component is orthogonal to the others; it can be considered as a translation, followed by a rotation, so that the data fits better around the axes, bringing out underlying patterns in the data.

Starting from the correlation matrix of the data, the eigenvectors (which correspond to the principal components) for the  $n$  biggest eigenvalues (the variances) are selected and transformed. The number of eigenvalues required to preserve most of the variance can be obtained from the shape of the "knee" in a plot of sorted normalised eigenvalues, known as a "scree diagram", from its shape (eg as in fig. 2, for D30, pebbles).



**Fig. 2.** Scree plot for PCA extraction, texture D30

The table below (Table 2) gives the PCA components for texture D30, shown to two decimal places: three components were extracted; the scree plot indicates that two were sufficient. “Subject” refers to the participants, listed alphabetically by their initials in the same order as in Table 1 above. For this texture, the 1<sup>st</sup> component accounts for 83.5% of the variance. In almost all cases for the full Brodatz image set and the selections made by the participants in our research, the 1<sup>st</sup> component accounts for over 90% of the variance.

**Table 2.** PCA components, for D30, alphabetical by participant

<b>Subject</b>	<b>Component</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
AH	0.99	-0.09	-0.05
AT	0.88	0.40	0.04
AW	0.99	-0.09	-0.05
BH	0.98	-0.05	-0.08
BT	0.96	-0.05	0.15
CD	0.89	0.24	0.31
ER	0.86	-0.20	0.45
FB	0.99	-0.09	-0.05
GS	0.87	-0.41	-0.10
HF	0.99	-0.09	-0.05
JG	0.96	-0.08	-0.02
JP	0.99	-0.09	-0.05
KM	0.97	0.04	-0.06
LT	0.96	-0.01	-0.12
MD	0.96	-0.01	-0.12

<b>Subject</b>	<b>Component</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
MG	0.98	-0.05	-0.08
MI	0.99	-0.09	-0.05
MM	0.58	-0.20	0.58
MT	0.85	0.09	-0.09
PB	0.93	0.10	-0.28
PH	0.99	-0.09	-0.05
RA	0.92	-0.24	0.23
RD	0.66	0.73	0.08
SF	0.80	0.45	0.28
SH	0.88	0.05	-0.29
SM	0.87	-0.41	-0.10
SP	0.99	-0.09	-0.05
ST	0.62	0.70	-0.17
TC	0.95	-0.03	-0.27
WH	0.91	0.11	0.33

### 3 A Mental Map of Perceived Similarity

A consensus view can be obtained by applying the first principal component to the matrix representing individual participants' selections; for example, subjects AH, AW, FB, HF, JP, MI, PH and SP all contributed 0.99 to this viewpoint, while MM was only 0.58. Multiplying out the responses gives the following values, shown to four decimal places, in Table 3. D31 has a value very close to that of the query image, D30; as Table 1 shows, all but five of the participants selected it as most like the query. D66 and D99 were only selected by three or four people respectively; the computed value is much lower.

**Table 3.** Computed values for D30

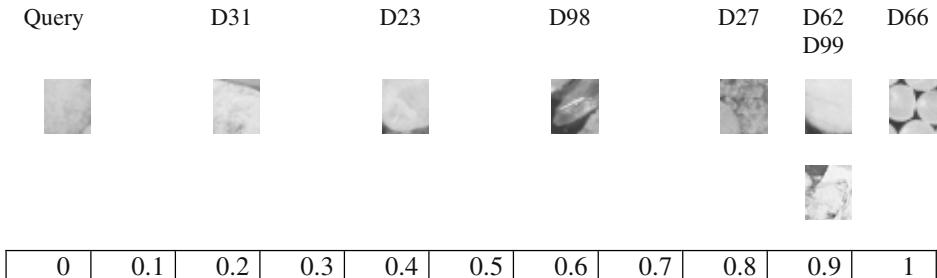
D23	72.5250
D27	24.5300
Query	136.0900
D31	102.4660
D62	10.2140
D66	4.1500
D98	45.2670
D99	4.4460

Scaling to a range of 0 for the query, 1 for the least like image, produces a scaled similarity vector (Table 4, shown to four decimal places).

**Table 4.** Scaled similarity for D30, from 1<sup>st</sup> principal component

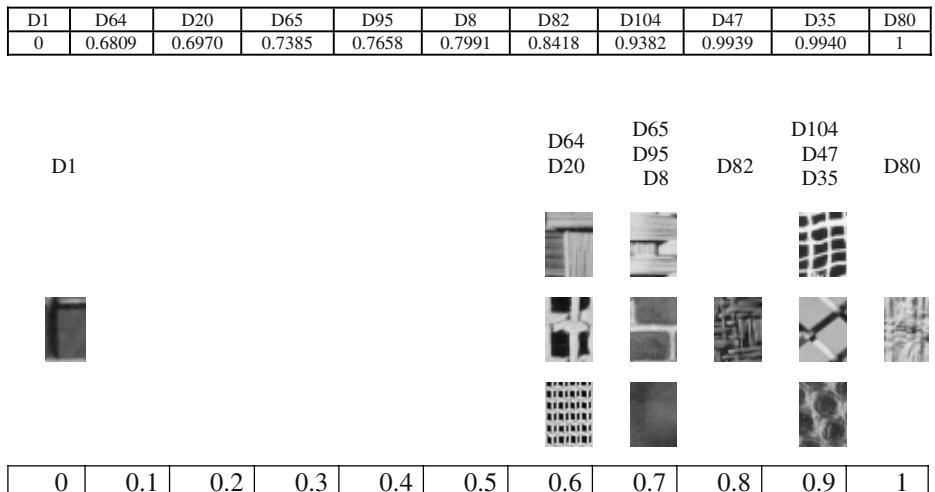
D30	D31	D23	D98	D27	D62	D99	D66
0	0.2548	0.4818	0.6884	0.8455	0.9540	0.9978	1

Gould and White (1986) show how individual ranking judgements can be transformed into such composite similarity scales [4]. Their examples are drawn from geography, and allow factors such as perceived attractiveness of different towns or regions of a country to be plotted, overlaid on a cartographic map; they call this approach "Mental Mapping". The same technique can be applied to the perceptually ranked images in our research. Using the scaled values of Table 4, a "mental map" can now be plotted, showing the ordering and relative spacing of the images; a distance measure has been derived corresponding to perceptual similarity. Fig. 3 indicates the relative distances, using the scaled values of Table 4, at 0.1 intervals, to provide a visual indication of the perceptually-derived ranking for this texture.

**Fig. 3.** Images along similarity scale, for D30

### 3.1 Similarity Is not Always Symmetric

Texture D1 is one of the easily identifiable images, of wire netting. However, although there are two other images, D46 and D47, also of wire netting, neither have the same orientation. The scaled similarity (table 5) shows that no other image was considered a close match; figure 4 plots the selected images along a scale from 0 to 1.

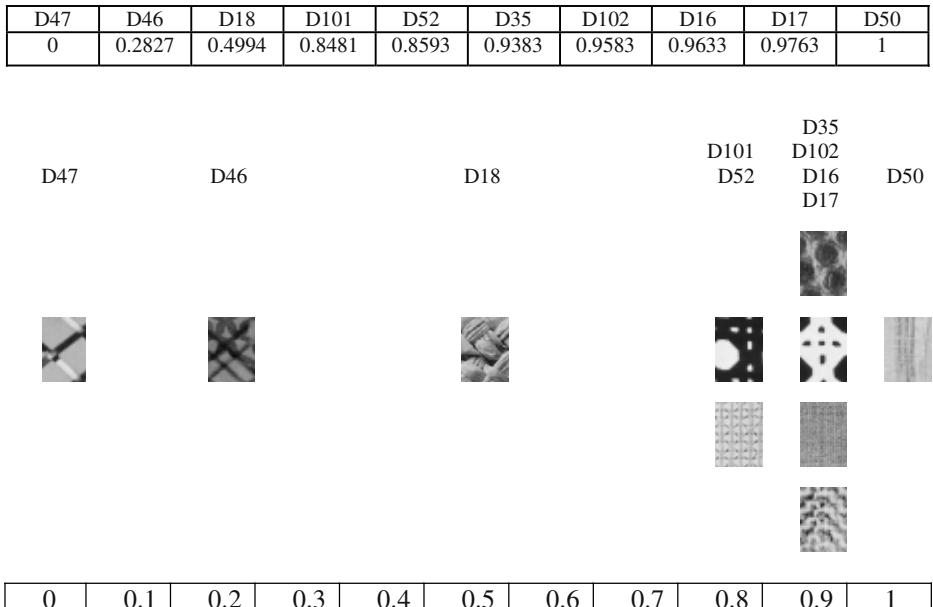
**Table 5.** Scaled similarity, from 1<sup>st</sup> principal component for D1**Fig. 4.** Images along similarity scale, for D1

The closest comparisons are to D64, D20, D65, D95, and D8, all of which show the same orientations at a similar scale. The next group includes D47, one of the other wire images, but this was considered less good a match. D46, the back-lit wire image, was not selected by any of the participants as a match for D1.

If D47 is next considered (table 6 and fig. 5), the lack of symmetry in perceptual similarity [12] is very noticeable. Here, the diagonal orientation is the most salient

feature, shown by the closeness of D46, then of D18. The remaining textures match on scale, or on grey level; D1, the only other image of wire, is not considered a match.

**Table 6.** Scaled similarity, from 1<sup>st</sup> PCA component for D47



**Fig. 5.** Images along similarity scale, for D47

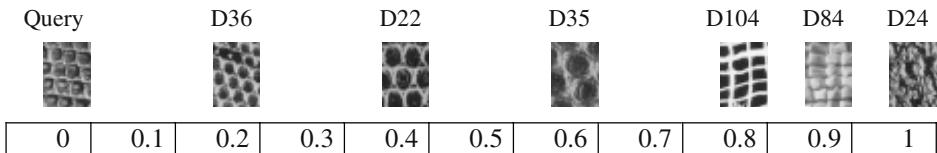
### 3.2 Salient Features

The reptile skin textures (D3, D22, D35 and D36) are very similar in regularity, and orientation (Table 7, and Fig 6). Taking D3 as the query image, the scale increases from D36, the most like, to D35, considered least like within this group.

**Table 7.** Scaled similarity, for D3

D3	D36	D22	D35	D104	D84	D24
0	0.2785	0.4161	0.6221	0.8885	0.9195	1

This demonstrates that human observers are not scale invariant, unlike some of the techniques proposed for image retrieval. Small scale is considered more salient here.

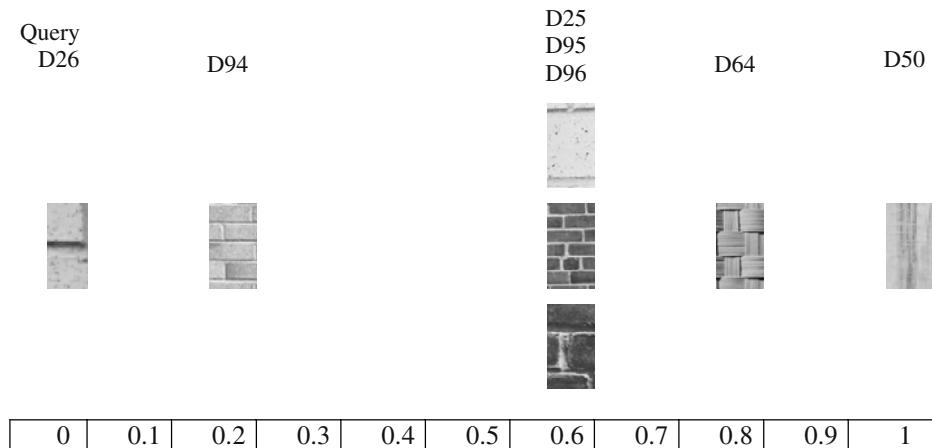
**Fig. 6.** Images along similarity scale, for D3

Considering now two of the "brick wall" textures, D26 and D94 (Table 9). D26 is considered to be most similar to D94, and *vice versa*, but the psychological distance between each as the query and the remaining brick textures, D25, D95 and D96, is noticeably different in each case.

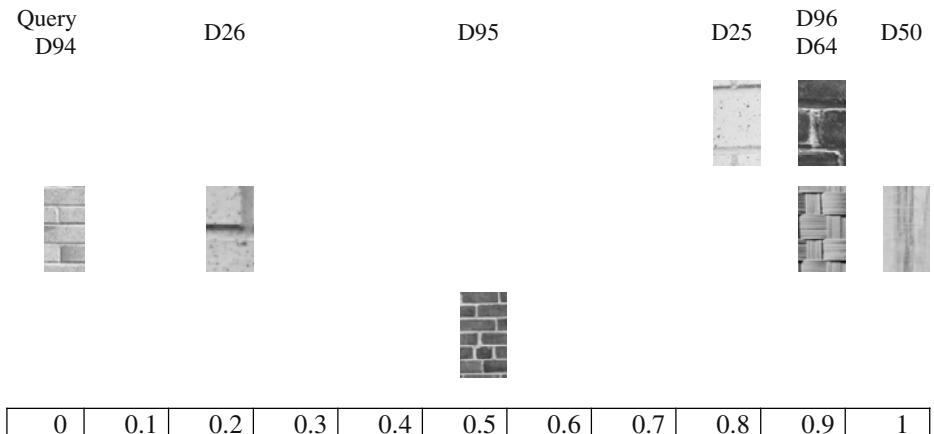
**Table 9.** Scaled similarity, for brick patterns D26, and D94

D26	D94	D25	D95	D96	D64	D50
0	0.2856	0.6277	0.6706	0.6953	0.8682	1
D94	D26	D95	D25	D96	D64	D56
0	0.2913	0.5917	0.8221	0.9030	0.9414	1

Figures 8 and 9 show this. Here, grey level appears to be the most salient feature, hence the closeness of D26 and D94 to each other. After this, scale applies, as it did for the reptile skin examples, all of which were of the same grey level and contrast.

**Fig. 8.** Images along similarity scale, for D26

D25 is considered to be similar to D26, and D95 to D94. Since Brodatz deliberately organised his photographs of textures to bring out similarities and contrasts, it would be expected that the participants in our study reached a similar ordering to his.



**Fig. 9.** Images along similarity scale, for D94

## 4 Conclusions

Images and searching for images continues to be an area of interest; individuals tend not to add metadata to their photographs; and even if they do, it is unlikely that the annotator will have considered all possible future uses of each image. Low-level features of the image itself can always be used, since they are an intrinsic part of the data itself; texture is particularly suitable, and applies equally to colour or "monochrome" originals.

However, if an image retrieval system is designed for end-users, rather than for specialists, it is unreasonable to expect such end-users to learn a specialised query language or interface; it is surely preferable to evaluate the retrieval performance of CBIR systems using perceptual judgements of image similarity. This research has shown how such a benchmark set may be obtained, working with the widely-used "Brodatz" textures image dataset.

We have shown how Gould and White's method of "mental mapping" may be applied to the individual perceptual ranking judgements. Correlation coefficients are calculated between each individual's ranking for each of the 112 Brodatz textures in turn; principal components analysis (PCA) is then applied to the correlation matrix for each texture. We found that the 1<sup>st</sup> principal component accounts for over 90% of the variance in almost all cases; for the example used in this paper, it accounted for 83.5%. This 1<sup>st</sup> principal component represents the weighting of each individual's view to the consensus view; multiplying out the response matrix with the 1<sup>st</sup> principal component and scaling the result, indicates psychological distance between similar images.

It was also observed that image similarity is not always symmetric, and that features such as scale, orientation, and contrast, affect the perceived saliency of such images.

## References

1. P. Brodatz, *Textures – A Photographic Album For Artists And Designers*, Dover (New York, 1966)
2. D.C.A. Bulterman, Is It Time for a Moratorium on Metadata?, *IEEE Multimedia*, Oct/Dec 2004, Vol 11, pp10-17
3. T. Glatard, J. Montagnat, I. E. Magnin, Texture Based Medical Image Indexing and Retrieval: Application to Cardiac Imaging, *Proceedings 6th ACM SIGMM International Workshop on Multimedia Image Retrieval (MIR 2004)*, New York, October 15-16, 2004, pp135-142
4. P. Gould and R. White, *Mental Maps* (2nd ed), Routledge (London, 1986, reprinted 2002)
5. I. T. Jolliffe, *Principal Component Analysis* (2nd ed), Springer (New York, 2002)
6. B. S. Manjunath and W. Y. Ma, Texture Features for Browsing and Retrieval of Image Data, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Aug 1996, Vol 18, No 8, pp837-842
7. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic and P. Yanker, The QBIC Project: Querying Images By Content Using Color, Texture & Shape, *IBM Research Report RJ9203*, 1 Feb 1993.
8. J. C. Nunes, O. Niang, Y. Bouaoune, E. Delechelle, and P. Bunel, Bidimensional Empirical Mode Decomposition Modified for Texture Analysis, In: *Proceedings of the 13th Scandinavian Conference on Image Analysis*, Göteborg, Sweden, June 2003, pp. 171-177.
9. J. S. Payne, L. Hepplewhite and T. J. Stonham, Evaluating Content-Based Image Retrieval Techniques Using Perceptually Based Metrics, In: *Proceedings of SPIE Electronic Imaging*, San Jose, USA, Jan 1999, vol. 3647, pp. 122-133.
10. J. S. Payne and T. J. Stonham, Can Texture and Image Content Retrieval Methods Match Human Perception?, In: *International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP'2001)*, Hong Kong, May 2001, pp. 154-157.
11. M. Pietikäinen, T. Ojala and O. Silven, Approaches to Texture-based Classification, Segmentation and Surface Inspection, In: *Handbook of Pattern Recognition and Computer Vision* (2nd ed), 1998, pp. 711-736.
12. A. Tversky, Features of Similarity, In: *Psychological Review*, July 1977, Vol 84 No 4, pp. 327-352

# Toward Automatic Motor Condition Diagnosis

J. Ilonen<sup>1</sup>, P. Paalanen<sup>1</sup>, J.-K. Kamarainen<sup>1</sup>, T. Lindh<sup>2</sup>, J. Ahola<sup>2</sup>,  
H. Kälviäinen<sup>1</sup>, and J. Partanen<sup>2</sup>

<sup>1</sup> Department of Information Technology,

<sup>2</sup> Department of Electrical Engineering,

Lappeenranta University of Technology,

P.O.Box 20, FI-53851, Lappeenranta, Finland

**Abstract.** In this study a method for automatic motor condition diagnosis is proposed. The method is based on a statistical discriminance measure which can be used to select the most discriminative features. New signals are classified to either a normal condition class or a failure class. The classification can be done traditionally using training examples from the both classes or using only probability distribution of the normal condition samples. The latter corresponds to typical situations in practice where the amount of failure data is insufficient. The results are verified using real measurements from induction motors in normal condition and with bearing faults.

## 1 Introduction

Automatic condition monitoring and diagnosis are important in modern industrial installations where a high degree of automation is desired. Automatic monitoring is needed to detect and recognize system faults, such as motor failures, where an early warning could prevent escalation of the problem. This is the case for example in motor bearing damage detection [9, 11]. There is a variety of sensor and control signals available, but current practice lacks general monitoring methods.

In this study, a diagnosis method is proposed to find discriminative regions, bands, from frequency content of two classes of signals (normal/failure) and to classify new measurements to these classes. The proposed method is useful in cases where there are measurements, but the physical characteristics of failures are not known. It is evident that a sufficient amount of measurements from the both classes are needed in order to find the most discriminative features, but the case where there are measurements mainly from the normal class is of special importance. In practice, measurements from failure conditions may not be sufficient due to economical reasons or difficulties to realize them.

In our experiments the proposed method was successfully applied to detection of bearing damages in induction motors based on their stator current content.

## 2 Discriminative Measures

### 2.1 Feature Extraction

A set of features must be selected and among them the best ones can be found by comparing their discriminative power.

Two sets of signals,  $x_k(t)$  and  $y_k(t)$ , represent examples from two classes,  $C_1$  and  $C_2$ , respectively. The sub-index  $k$  denotes a measurement number,  $k = 0, 1, \dots, N_1 - 1$  for  $C_1$  and  $k = 0, 1, \dots, N_2 - 1$  for  $C_2$ . It is assumed that the signals are measured during a stationary system mode, i.e., system parameters such as rolling speed and load are constant. Now, the discriminative information should be present at some frequency band and it is sufficient to apply a band-pass filter  $\psi(t)$ . In a stationary system mode the time information can be ignored and a global feature, such as a power spectrum

$$\int_{-\infty}^{\infty} |\psi(t) * x_k(t)|^2 dt, \quad (1)$$

can be utilized. The selection of the best features is reduced to finding the optimal values for the central frequency  $f$  and bandwidth  $\gamma$  of a band-pass filter. The normalized Gabor filter

$$\psi(t) = \frac{|f|}{\gamma\sqrt{\pi}} e^{-(\frac{f}{\gamma})^2 t^2} e^{j2\pi f t} \quad (2)$$

where  $f$  denotes the central frequency and  $\gamma$  controls the spatial sharpness and frequency bandwidth can be used as the band-pass filter.

### 2.2 Statistical Measures for Discriminative Power

If there are several frequency bands where the contents of the classes  $C_1$  and  $C_2$  are dissimilar, then the band where the separation of the classes is most evident should be selected. The first-order statistics approach is not sufficient since it simply selects the frequency band where the distance between the expectations is largest, but neglects the variance information, and thus, a significant overlap of the class probabilities may exist [5, 4]. In the second-order statistics typically an assumption must be made about forms of probability distributions  $p_x(n)$  and  $p_y(n)$  which describe the spread of features of the both classes.

It was assumed that the features are extracted from signals measured during a constant operation mode where variance in the measurements is supposed to be caused by a large number of unknown independent sources. It can be thus assumed that the form of the probability distributions is Gaussian. Now, both classes are uniquely defined by their expectations ( $\mu_x, \mu_y$ )

$$\mu_\xi = E \left[ \int_{-\infty}^{\infty} |\psi(t) * \xi_k(t)|^2 dt \right] \quad (3)$$

and variances ( $\sigma_x^2$ ,  $\sigma_y^2$ )

$$\sigma_\xi^2 = E \left[ \left( \int_{-\infty}^{\infty} |\psi(t) * \xi_k(t)|^2 dt - \mu_\xi \right)^2 \right]. \quad (4)$$

For Gaussians Fisher's discriminant ratio (*FDR*) can be used to measure the distance between the distributions [7]

$$FDR(p_x(n), p_y(n)) = \frac{(\mu_x - \mu_y)^2}{\sigma_x^2 + \sigma_y^2}. \quad (5)$$

Using the divergence measure in (5) the discriminative energy function can be defined as

$$E = \frac{1}{2} \left( \frac{(\mu_x - \mu_y)^2}{\sigma_x^2 + \sigma_y^2} \right)^2. \quad (6)$$

In this case it was possible to establish a formula for the discriminative energy, but in a more general cases other techniques, such as the AdaBoost boosting algorithm [2], can be used to select features.

### 2.3 Verifying Gaussianity

Since the measure is based on the Gaussianity assumption it must be verified. Kurtosis is a fourth-order statistic and has been used to examine the Gaussianity of distributions. A normalized kurtosis with adjusting for bias is [3]

$$k(x) = \frac{E \{(x - m)^4\}}{[E \{(x - m)^2\}]^2} - 3 \quad (7)$$

where  $m$  is the mean of  $x$ . The kurtosis is zero for a Gaussian distribution. Distributions having a negative kurtosis are subgaussian, e.g., uniform or multimodal distributions, and those with a positive kurtosis are supergaussian. A typical supergaussian distribution has a sharp peak and long tails.

### 2.4 Bayesian Classification and Confidence

Using the discriminative energy function (6) the frequency  $f$  and bandwidth  $\gamma$  of the band-pass filter in (2) can be optimized for features in (1). Single or several frequencies can be selected.

Since the normal distribution of features was assumed it is straightforward to establish estimates for means and variances and to classify new signals using the Bayesian decision making which is based on probability theory and the principle of choosing the most probable or the lowest risk (expected cost) option [10].

In this study we wanted to stress also the issue that quality or number of failure measurements is not sufficient or does not cover all failure states. In that case the classification should be based only on the probability distribution of normal condition measurements. This requirement can be established using an equiprobability surface where features inside the surface are classified as normal and outside as failure. The confidence value which defines the surface must

be selected manually or it can be computed from the measurements using rank order statistics, i.e., by selecting a fractile of training data which is covered by the smallest probability mass inside the surface. Algorithm 1 computes the probability limit value which can be used in classification. The confidence measure is more thoroughly examined from theoretical point of view in [6].

### Algorithm 1.

- 1:  $\bar{s} = \bar{p} = p_{i=1,\dots,N_1}, \dots, N_1, \dots, C_1$   
 2:  $\bar{s} = \bar{p} = p_{i=1,\dots,N_1}, \dots, N_1, \dots, C_1$   
 3:  $\bar{s} = \bar{p} = \{ \dots, \dots, \dots, \dots, \dots \}$   
 4:  $p_{eq} = *N \{ \dots, \dots, \dots, \dots, \dots \}$

### 3 Experiments

### 3.1 Induction Motor Bearing Damages

Induction motors have been a widely studied subject of the condition monitoring [11, 1]. An important sub-category of induction motor failures are bearing damages, which can be detected from vibration, acoustic noise, temperature, or stator current signals. Bearing damages are attractive for evaluating the proposed method since characteristic frequencies of damage appearance can be analytically solved and compared to automatically found frequencies.

The exact bearing fault characteristic frequencies calculated from the bearing geometry are valid only for ideal bearings. In practice the rolling elements also slide in addition to rotation, and thus, both in literature and practice approximate equations are often used [8]. The approximate equations for inner and outer race defects are

$$f_o = 0.4Nf_r \text{ and } f_i = 0.6Nf_r \quad (8)$$

where  $N$  is the number of balls or rollers and  $f_r$  is the rotational speed of the rotor.

Stator current is a potential source for detecting faults since rotor eccentricity causes changes to the current. It should be noted that there are always eccentricities in the rotors, but the eccentricities change in the case of a fault. Bearing defects establish new current components which are present at the frequencies

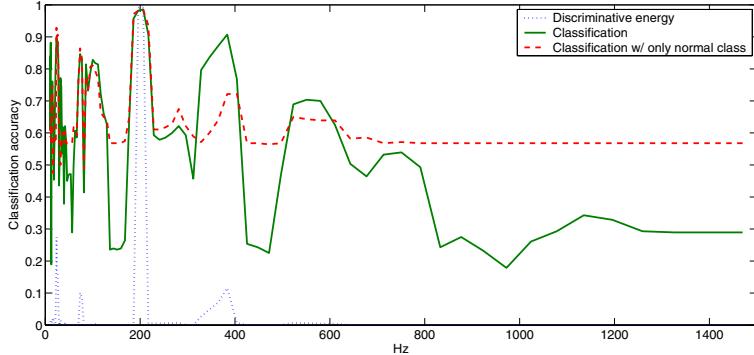
$$f_{ib} = |f \pm m \cdot f_{vb}|, m = 1, 2, 3, \dots \quad (9)$$

where  $f$  is the supply frequency and  $f_{vb}$  is the characteristic frequency of vibration caused by the fault [9].

### 3.2 Bearing Damage Detection Based on Stator Current

The stator current data consisted of stator current signals measured from motors in a normal condition ( $C_1$ ) and motors with a bearing damage ( $C_2$ ). The measurements contain two cases: no load connected to motors and with a full load.

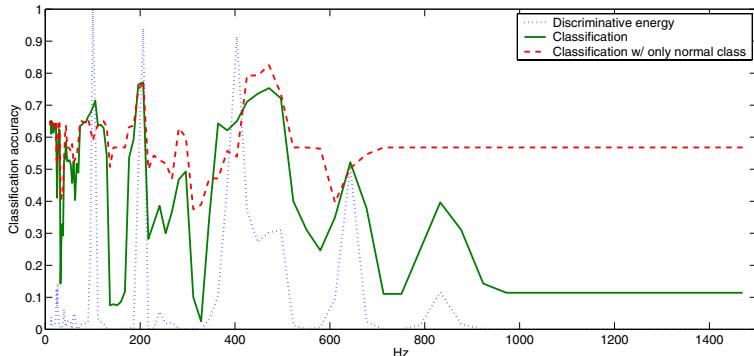
For motors with no load discriminative energy  $E$  and classification results are presented in Fig. 1. In this case discriminative energy had its maximum near the first harmonic (202 Hz) of the characteristic frequency (101 Hz). Also the both classification schemes had the maximal accuracy at the same frequency band.



**Fig. 1.** Discriminative energy and classification accuracies for motors with no load

For motors with a full load results are shown in Fig. 2. This was a more difficult situation since full load caused various disturbances, but still, the characteristic frequency (101 Hz) and some of its harmonics contained discriminative information. Classifications succeeded on the same frequencies, but due to disturbances the accuracy decreased.

In these experiments the classification was performed using the Bayesian decision making, which requires examples from the both classes, and using the fractile based limit (Algorithm 1), when failure measurements are not needed. The fractile was set to 100% which means that values within a confidence region that covers the whole training set are accepted as normal condition and all others



**Fig. 2.** Discriminative energy and classification accuracies for motors with full load

classified as a failure. In the both schemes practically the same classification accuracy was achieved at the most discriminative frequencies.

**Comparative Results.** The experiments were repeated by utilizing the characteristic frequencies in (8)–(9) as reported by Yazici and Kliman in [11]. For the characteristic frequencies and our approach the results are shown in Table 1. Classification was done using the three highest peaks of  $E$ , both separately and combined, and compared to the classification results with the characteristic frequencies. Characteristic frequencies provided an accuracy of 97.5% correct classification for a motor with no load. The most discriminative frequency of  $E$  provided the same accuracy, but 100% accuracy was achieved with a combination of the three most discriminative frequencies. For a motor with a full load classification with the characteristic frequencies provided an accuracy of only 66.8% while the three most discriminative frequencies provided a slightly better classification result, 72.1%.

**Table 1.** Classification results using calculated characteristic frequencies and three most discriminative frequencies by  $E$

	No load		Full load	
	Freq.	Correct	Freq.	Correct
Char. freq.		97.5%		66.8%
1st peak	206.3 Hz	97.5%	100.1 Hz	69.3%
2nd peak	23.6 Hz	87.9%	206.3 Hz	77.9%
3rd peak	383.6 Hz	91.4%	403.9 Hz	68.6%
Combined		100.0%		72.1%

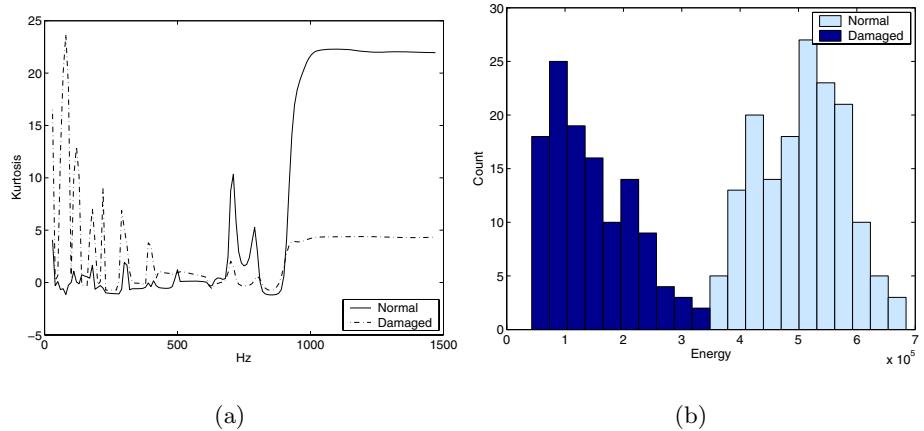
Utilizing the most discriminative frequencies by  $E$  a better accuracy was achieved than with the computed characteristic frequencies as used in literature. It seems that some of the harmonics include noise which harms the classification.

### 3.3 Verifying Gaussianity

The Fisher discriminant is based on the Gaussian probability densities. For other types densities must be explicitly estimated, for example, by using Gaussian mixture models, and the Fisher discriminant must be replaced, for example, with the Kullback-Leibler divergence. The experiments were based on the Gaussianity assumption, and thus, its validity is demonstrated.

Gaussianity was measured with the kurtosis in (7). Kurtoses of the two classes of measurements are shown in Fig. 3(a). At certain frequencies the kurtosis deviates significantly from zero and for those frequencies also the divergence measures fail and the classification cannot provide accurate results. However, by inspecting feature values at locations where discriminative energy is maximal one can ob-

serve a kurtosis close to zero and also good a separation of the two classes (Fig. 3(b)).



**Fig. 3.** (a) Kurtosis of features; (b) feature distributions at 200 Hz (approx. Gaussian)

## 4 Discussion

In this study, an automatic motor condition diagnosis was studied and methods to automatically select the most discriminative features and to classify new signals were examined. In addition, the case where the amount of failure condition measurements is not sufficient was considered.

In the experiments with the real data the provided measure  $E$  was able to find discriminative frequency bands which coincided with the bearing damage fault characteristic frequencies and their harmonics. The classification also succeeded on the same frequencies confirming their validity. The classification based on samples from only the normal class performed equally as compared to the classification with data from the both classes. This is an important result since all possible failure types can rarely be covered in training data.

There is still room for certain improvements which will be addressed in the future to move toward general condition monitoring methods. The stationary assumption can be relaxed by segmenting the signal to stationary parts, the Gaussianity limitation can be overcome by using Gaussian mixture models and the Kullback-Leibler divergence measure. Furthermore, other types of feature selectors can be used, such as the AdaBoost boosting algorithm.

## Acknowledgements

Academy of Finland (project 204708) and European Union (TEKES/EAKR project 70056/04) are acknowledged for financial support.

## References

1. M.E.H. Benbouzid, M. Vieira, and C. Theys. Induction motors' faults detection and localization using stator current advanced signal processing techniques. *IEEE Transactions on Power Electronics*, 14(1):14–22, 1999.
2. Y. Freund and R. E. Schapire. A decision theoretic generalization of on-line and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
3. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*, chapter 2.7. John Wiley & Sons, Inc., 2001.
4. J.-K. Kamarainen, V. Kyrki, T. Lindh, J. Ahola, and J. Partanen. Statistical signal discrimination for condition diagnosis. In *Proceedings of the Finnish Signal Processing Symposium*, pages 195–198, Tampere, Finland, 2003.
5. T. Lindh, J. Ahola, J.-K. Kamarainen, V. Kyrki, and J. Partanen. Bearing damage detection based on statistical discrimination of stator current. In *Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, pages 177–181, Atlanta, Georgia, USA, 2003.
6. P. Paalanen, J.-K. Kamarainen, J. Ilonen, and H. Kälviäinen. Feature representation and discrimination based on Gaussian mixture model probability densities – practices and algorithms. Research Report 95, Department of Information Technology, Lappeenranta University of Technology, 2005.
7. D.W. Peterson and R.L. Mattson. A method of finding linear discriminant functions for a class of performance criteria. *IEEE Transactions on Information Theory*, 12(3):380–387, 1966.
8. R.L. Schiltz. Forcing frequency identification of rolling element bearings. *Sound and Vibration*, pages 16–19, 1990.
9. R.R. Schoen, T.G. Habetler, F. Kamran, and R.G. Bartfield. Motor bearing damage detection using stator current monitoring. *IEEE Transactions on Industry Applications*, 31(6):1274–1279, 1995.
10. S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999. ISBN 0-12-686140-4.
11. B. Yazici and G.B. Kliman. An adaptive statistical time-frequency method for detection of broken bars and bearing faults in motors using stator current. *IEEE Transactions on Industry Applications*, 35:442–452, 1999.

# Improving K-Means by Outlier Removal

Ville Hautamäki, Svetlana Cherednichenko, Ismo Kärkkäinen,  
Tomi Kinnunen, and Pasi Fränti

Speech and Image Processing Unit,  
Department of Computer Science, University of Joensuu,  
P.O. Box 111, FI-80101, Joensuu, Finland  
`{villeh, schered, iak, tkinnu, franti}@cs.joensuu.fi`

**Abstract.** We present an Outlier Removal Clustering (ORC) algorithm that provides outlier detection and data clustering simultaneously. The method employs both clustering and outlier discovery to improve estimation of the centroids of the generative distribution. The proposed algorithm consists of two stages. The first stage consist of purely K-means process, while the second stage iteratively removes the vectors which are far from their cluster centroids. We provide experimental results on three different synthetic datasets and three map images which were corrupted by lossy compression. The results indicate that the proposed method has a lower error on datasets with overlapping clusters than the competing methods.

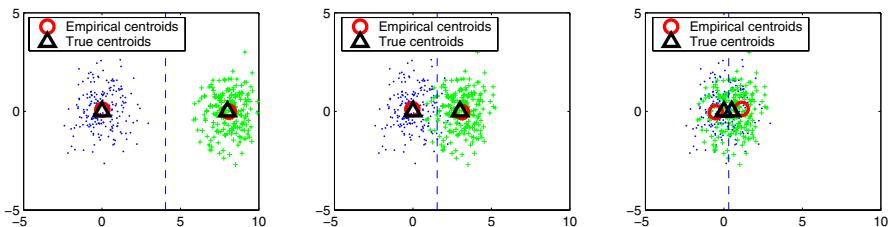
## 1 Introduction

[1] is an iterative clustering algorithm widely used in pattern recognition and data mining for finding statistical structures in data. K-means takes a training set  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  and a desired number of clusters  $M$  as its input and produces a codebook  $C = \{\mathbf{c}_1, \dots, \mathbf{c}_M\}$ . The elements  $\mathbf{c}_i$  are called code vectors, and they define partition of the input space into disjoint sets  $\{P_1, \dots, P_M\}$  so that  $P_i = \{\mathbf{x} \in \mathbb{R}^d; \|\mathbf{x} - \mathbf{c}_i\| \leq \|\mathbf{x} - \mathbf{c}_j\| \forall j \neq i\}$ . These sets are called clusters. As the clusters are disjoint, each input vector  $\mathbf{x}_i \in X$  belongs to exactly one cluster, the one whose code vector is nearest to  $\mathbf{x}_i$ . Cluster index is denoted here as  $p_i \in \{1, 2, \dots, M\}$ . The size of a cluster is defined as the number of input vectors assigned to the cluster and will be denoted here as  $n_i$ .

K-means starts with an initial codebook  $C$  and partition  $P$  it and improves them iteratively. If K-means has been initialized well, the resulting code vectors tend to be located at locally dense regions of the input space. In this way, K-means can be considered as a nonparametric density estimator which attempts to fit a model  $C$  into the input data. Text-independent speaker recognition is an example in which K-means is used in the role of a density estimator [2]. There each cluster can be thought as roughly representing a single phonetic class, and each codebook representing the distribution of the speaker's characteristic vocal space.

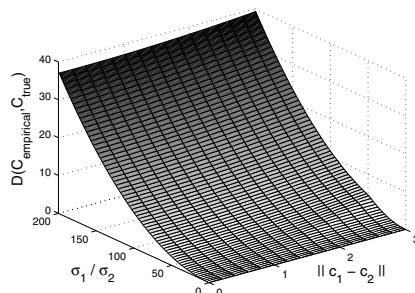
Given the density estimation property of K-means, it is desirable that the code vectors would be located at the correct locations, or in other words, at

the centres of the actual clusters that generated  $X$ . However, even if the initial code vectors would be located at the true locations, there is no guarantee that these would be the final estimated centroids. This happens more easily for overlapping clusters, see Fig. 1 for an illustration. In this figure, two Gaussian clusters having same variance have been generated by drawing  $N = 200$  independent random samples from each class, and the distance between the mean vectors is varied in the three panels. The triangles denote the true centroids of the clusters and the open circles denote the estimated centroids using K-means. The true mean vectors have been used as the initial codebook to K-means. The dotted line shows the Voronoi partitioning based on the empirical centroids.



**Fig. 1.** Problem of K-means as a density estimator

In this trivial case, we can observe that for an increased cluster overlap, the estimated cluster centroids deviate more from the true ones. The reason for this is a fundamental problem of K-means: the clusters are disjoint, and the training vectors assigned to a wrong cluster deviate the estimated centroids away from the true ones. Figure 2 shows the error between the estimated codebook and the true codebook as a function of the distance between the cluster centers  $\|\mathbf{c}_1 - \mathbf{c}_2\|$  and the ratio of their standard deviations  $\sigma_1/\sigma_2$ . Based on these observations, the usefulness of K-means as a density estimator is questionable.



**Fig. 2.** Error between the true and estimated models

Parametric and fuzzy clustering models such as (GMM) algorithm [3] and (FCM) [4] can be used to attack the problem of nonoverlapping clusters. However, K-means remains probably the most widely used clustering method, because it is simple to implement and provides reasonably good results in most cases. In this paper, we improve the K-means based density estimation by embedding a simple procedure in the algorithm. The proposed method iteratively removes vectors far from the currently estimated codevectors. By modifying the training set  $X$  in this way, we compensate for the nonoverlapping cluster assumption. However, the algorithm should remove also points that are in the overlapping regions of clusters.

## 2 Outlier Removal Followed by Clustering

is defined as a noisy observation, which does not fit to the assumed model that generated the data. In clustering, outliers are considered as observations that should be removed in order to make clustering more reliable [5].

In outlier detection methods based on clustering, outlier is defined to be an observation that does not fit to the overall clustering pattern [6]. The ability to detect outliers can be improved using a combined perspective of outlier detection and clustering. Some clustering algorithms, for example DBSCAN [7] and ROCK [8], handle outliers as special observations, but their main concern is clustering the dataset, not detecting outliers.

(ODIN) [9] is a local density-based outlier detection algorithm. Local density based scheme can be used in cluster thinning. Outlier removal algorithm can remove vectors from the overlapping regions between clusters, if the assumption holds that the regions are of relatively low density. Higher density is found near the cluster centroid. An obvious approach to use outlier rejection in the cluster thinning is as follows: (i) eliminate outliers (ii) cluster the data using any method.

---

**Algorithm 1.** ODIN+K-means( $k, T$ )

---

```

 $\{\text{ind}(\mathbf{x}_i) | i = 1, \dots, N\} \leftarrow \text{Calculate kNN graph}$ 
for  $i \leftarrow 1, \dots, N$  do
     $o_i \leftarrow 1 / (\text{ind}(\mathbf{x}_i) + 1)$ 
    if  $o_i > T$  then
         $X \leftarrow X \setminus \{\mathbf{x}_i\}$ 
    end if
end for
 $(C, P) \leftarrow \text{K-means}(X)$ 

```

---

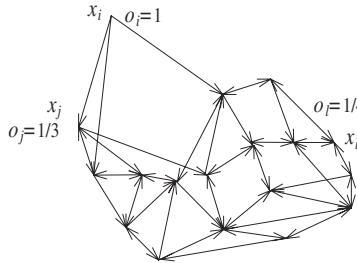
In this paper, we compare the proposed method against aforementioned scheme, in which the outlier removal method is ODIN and the clustering algorithm K-means. In ODIN, the outliers are defined using a k-nearest neighbour

(kNN) graph, in which every vertex represents a data vector, and the edges are pointers to neighbouring  $k$  vectors. The weight of an edge  $e_{ij}$  is  $\|\mathbf{x}_i - \mathbf{x}_j\|$ .

In ODIN, the outlyingness of  $\mathbf{x}_i$  is defined as:

$$o_i = \frac{1}{\text{ind}(\mathbf{x}_i) + 1}, \quad (1)$$

where  $\text{ind}(\mathbf{x}_i)$  is the degree of the vertex  $\mathbf{x}_i$ , i.e. the number of edges pointing to  $\mathbf{x}_i$ . In the first step of ODIN, a kNN graph is created for the dataset  $X$ . Then, each vertex is visited to check if its outlyingness is above threshold  $T$ . Fig. 3 shows an example of kNN graph and the outlyingness values calculated for three vectors.



**Fig. 3.** Example of outlyingness factors in ODIN

### 3 Proposed Method

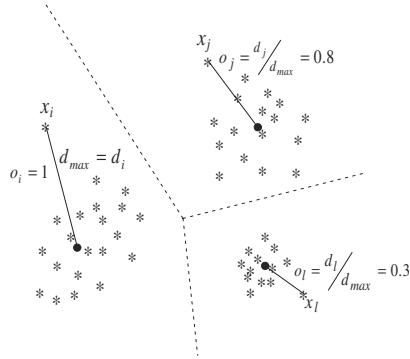
The objective of the proposed algorithm that we call Outlier Removal Clustering (ORC), is to produce a codebook as close as possible to the mean vector parameters that generated the original data. It consists of two consecutive stages, which are repeated several times. In the first stage, we perform K-means algorithm until convergence, and in the second stage, we assign an outlyingness factor for each vector. Factor depends on its distance from the cluster centroid. Then algorithm iterations start, with first finding the vector with maximum distance to the partition centroid  $d_{\max}$ :

$$d_{\max} = \max_i \{\|\mathbf{x}_i - \mathbf{c}_{p_i}\|\}, \quad i = 1, \dots, N. \quad (2)$$

Outlyingness factors for each vector are then calculated. We define the outlyingness of a vector  $\mathbf{x}_i$  as follows:

$$o_i = \frac{\|\mathbf{x}_i - \mathbf{c}_{p_i}\|}{d_{\max}}. \quad (3)$$

We see that all outlyingness factors of the dataset are normalized to the scale  $[0, 1]$ . The greater the value, the more likely the vector is an outlier. An example of dataset clustered in three clusters and calculated outlyingness factors is shown in Fig. 4.



**Fig. 4.** Example of outlyingness factors in ORC

---

**Algorithm 2.** ORC( $I, T$ )

---

```

 $C \leftarrow$  Run K-means with multiple initial solutions, pick best  $C$ 
for  $j \leftarrow 1, \dots, I$  do
     $d_{\max} \leftarrow \max_i \{ \|x_i - c_{p_i}\| \}$ 
    for  $i \leftarrow 1, \dots, N$  do
         $o_i = \|x_i - c_{p_i}\| / d_{\max}$ 
        if  $o_i > T$  then
             $X \leftarrow X \setminus \{x_i\}$ 
        end if
    end for
     $(C, P) \leftarrow$  K-means( $X, C$ )
end for

```

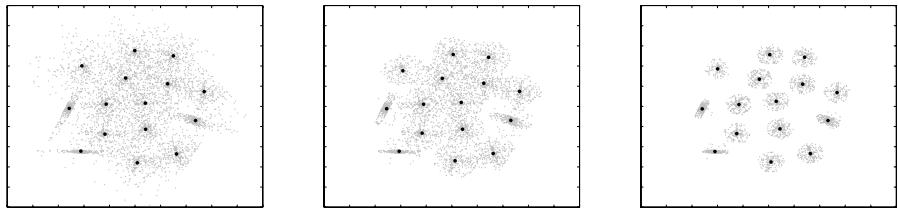
---

The vectors for which  $o_i > T$ , are defined as outliers and removed from the dataset. At the end of each iteration, K-means is run with previous the  $C$  as the initial codebook, so new solution will be a fine-tuned solution for the reduced dataset. By setting the threshold to  $T < 1$ , at least one vector is removed. Thus, increasing the number of iterations and decreasing the threshold will in effect remove more vectors from the dataset, possibly all vectors.

Fig. 5 shows an example of running the proposed method on a dataset with strongly overlapping cluster so that even the cluster boundaries are not easily observable. The black dots are the original centroids. We see that with 40 iterations clusters are little bit separated and with 70 iterations clusters are totally separated.

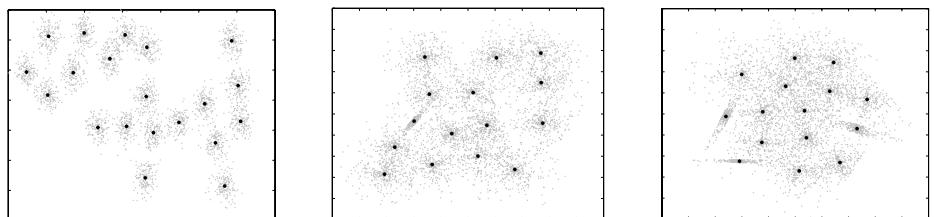
## 4 Experiments

We run experiments on three synthetic datasets denoted as A1, S3 and S4 [10], which are shown in Fig. 6 and summarized in Table 1. The original cluster



**Fig. 5.** Example of ORC. Original dataset (left), after 40 iterations (center), and after 70 iterations (right). The removal threshold is set to  $T = 0.95$  in all cases

centroids are also shown in the same figure. Vectors in datasets are drawn from multinormal distributions. In dataset A1, the clusters are fairly well separated. In dataset S3, the clusters are slightly overlapping, and in dataset S4, the clusters are highly overlapping.



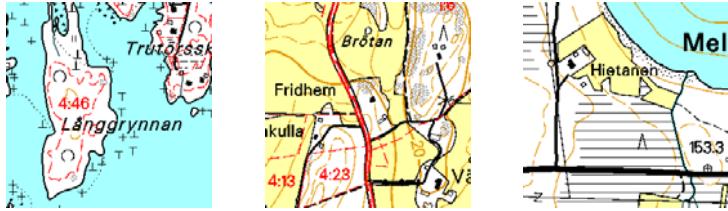
**Fig. 6.** Datasets, A1 (left), S3 (center) and S4 (right)

We run experiments also on three map image datasets (M1, M2 and M3), which are shown in Fig. 7. Map images are distorted by compressing them with a JPEG lossy compression method. The objective is to use color quantization for finding as close approximation of the original colors as possible. JPEG compression of map images creates so-called ‘‘ring’’ around the edges due to the quantization of the cosine function coefficients. In [11], color quantization methods were used to find the original colors. We apply the proposed algorithm to this problem, and we assume that the number of colors is known in advance.

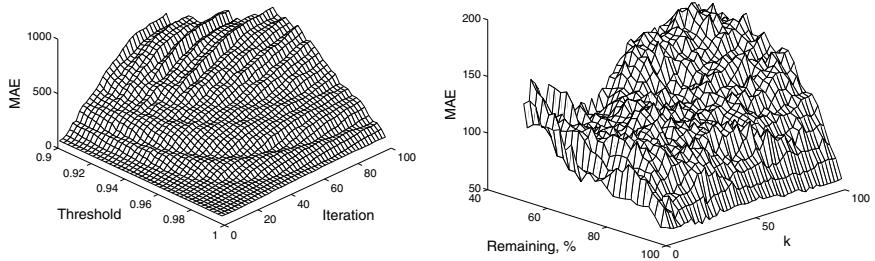
We calculate  $\frac{||\text{empirical codebook} - \text{generative codebook}||}{\text{generative codebook}}$  (MAE) to measure the difference between the empirical codebook and the generative codebook. For ODIN with K-means, we vary both the neighbourhood size  $k$  and the number of vectors removed. For ORC, we vary the number of iterations  $I$  and the threshold  $T$ .

**Table 1.** Summary of the datasets

Dataset	$N$	$M$	Dataset	$N$	$M$
A1	3000	20	M1	73714	5
S3	5000	15	M2	126246	6
S4	5000	15	M3	69115	5



**Fig. 7.** Sample 256x256 pixel fragment from the test images M1 (left), M2 (center) and M3 (right)

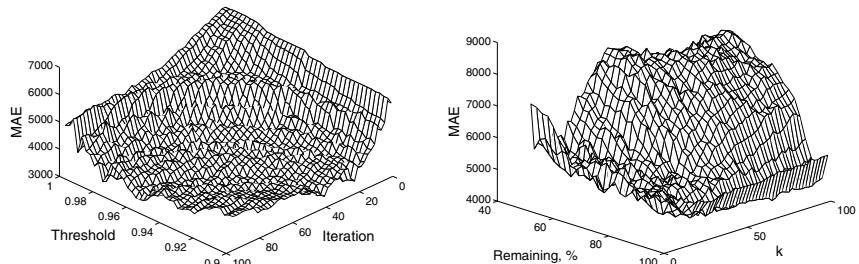


**Fig. 8.** Results for the dataset A1 for ORC (left) and for ODIN with K-means (right)

#### 4.1 Results

Fig. 8 shows the results for the dataset A1. We observe that increasing the parameters in the algorithms increases the error. Fig. 9 shows the results for the dataset S3. Situation without ORC iterations and threshold is shown in the back corner (in contrast to the previous figure, due to the shape of the error surface). ODIN has two “valleys”, where distortion values are lower, but the error is consistently decreasing as iterations proceed or the threshold is decreased.

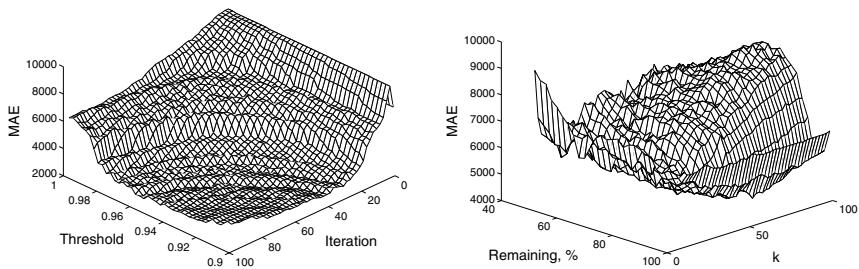
Fig. 10 shows the results for the dataset S4. Again, ODIN with K-means has two “valleys” where the error is lower. Regarding the number of remaining vectors, we see that the more vectors we remove with the ORC algorithm, the



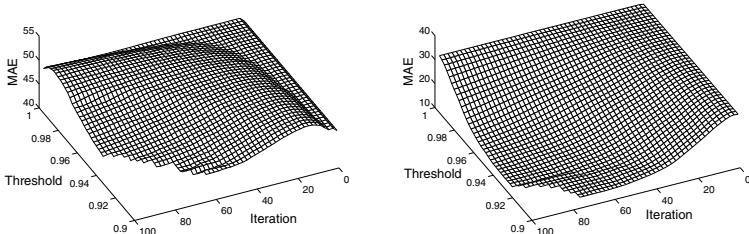
**Fig. 9.** Results for the dataset S3 for ORC (left) and for ODIN with K-means (right)

better the accuracy will be. This is because the ORC algorithm works as designed for S4 dataset by removing vectors that reside between clusters. On the other hand, when increasing parameters in ODIN algorithm first, we get lower error and then the error starts to increase.

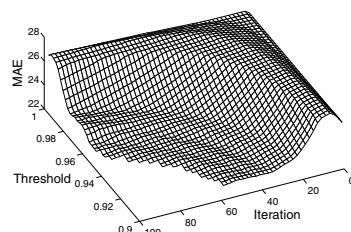
Results for the M1 - M3 datasets running the ORC algorithm are presented in Figs. 11 and 12. We note that for all the test cases, ORC reaches lower error when the number of iterations is increased enough or the threshold is decreased. Error surface of the dataset M1 has an interesting behaviour, where error first increases and then it starts to decrease. Error surfaces for ODIN are omitted as with all parameter combinations the error increases in relation to the standard K-means.



**Fig. 10.** Results for the dataset S4 for ORC (left) and for ODIN K-means (right)



**Fig. 11.** Results for the datasets M1 (left) and M2 (right) for ORC



**Fig. 12.** Results for the dataset M3 for ORC

**Table 2.** Best MAEs obtained

Algorithm	A1	S3	S4	M1	M2	M3
Plain K-means	60	5719	7100	47	32	26
ODIN + K-means	58	4439	4754	61	48	45
ORC	56	3329	2813	45	13	23

In Table 2, we show the smallest MAE between original codebook and those obtained by using K-means, ODIN with K-means and ORC. The results indicate potential of the proposed method. The ORC algorithm outperforms the traditional K-means and K-means preceded by outlier removal for all three data sets. For the non-overlapping data set (A1), the results are close to each other. However, when cluster overlap is increased, the proposed algorithm shows substantially improved performance over the baseline methods. For the most difficult data set (S4), the proposed method gives 1.5 - 2 times smaller error. Although parameter setting might be a difficult depending on the dataset. For the map image datasets, ORC performs systematically better than K-means in all cases. With datasets M1 and M3, ORC and K-means are close to each other in performance, but for M2 ORC more than halves the error in relation to K-means.

## 5 Conclusions

In this paper, we have proposed to integrate outlier removal into K-means clustering (ORC) for nonparametric model estimation. The proposed method was also compared with the standard K-means without outlier removal, and a simple approach in which outlier removal precedes the actual clustering. The proposed method was evaluated on three synthetic data sets with known parameters of the generative distribution and three map image datasets with known cluster centroids.

The test results show that the method outperforms the two baseline methods, particularly in the case of heavily overlapping clusters. A drawback is that correct parameter setting seems to depend on the dataset. Thus, the parameter setting should be automatized in future.

## References

1. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer desing. *IEEE Transactions on Communications* **28** (1980) 84–95
2. Kinnunen, T., Karpov, E., Fräntti, P.: Real-time speaker identification and verification. *IEEE Transactions on Speech and Audio Processing* (2005) Accepted for publication.
3. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* **39** (1977) 1–38

4. Dunn, J.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* **3** (1974) 32–57
5. Guha, S., Rastogi, R., Shim, K.: CURE an efficient clustering algorithm for large databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington (1998) 73–84
6. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery* **1** (1997) 141–182
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining. (1996) 226–231
8. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. In: 15th International Conference on Data Engineering. (1999) 512–521
9. Hautamäki, V., Kärkkäinen, I., Fränti, P.: Outlier detection using k-nearest neighbour graph. In: 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, United Kingdom (2004) 430–433
10. Virmajoki, O.: Pairwise Nearest Neighbor Method Revisited. PhD thesis, University of Joensuu, Joensuu, Finland (2004)
11. Kopylov, P., Fränti, P.: Color quantization of map images. In: IASTED Conference on Visualization, Imaging, and Image Processing (VIIP'04), Marbella, Spain (2004) 837–842

# Maximal Digital Straight Segments and Convergence of Discrete Geometric Estimators

François de Vieilleville<sup>1</sup>, Jacques-Olivier Lachaud<sup>1</sup>, and Fabien Feschet<sup>2</sup>

<sup>1</sup> LaBRI, Univ. Bordeaux 1

351 cours de la Libération, 33405 Talence, France.

{devieill, lachaud}@labri.fr

<sup>2</sup> LLAIC1, IUT Clermont-Ferrand 1

Campus des Cézeaux, 63172 Aubière Cedex, France

feschet@llaic3.u-clermont1.fr

**Abstract.** Discrete geometric estimators approach geometric quantities on digitized shapes without any knowledge of the continuous shape. A classical yet difficult problem is to show that an estimator asymptotically converges toward the true geometric quantity as the resolution increases. We study here the convergence of local estimators based on Digital Straight Segment (DSS) recognition. It is closely linked to the asymptotic growth of maximal DSS, for which we show bounds both about their number and sizes. These results not only give better insights about digitized curves but indicate that curvature estimators based on local DSS recognition are not likely to converge. We indeed invalidate an hypothesis which was essential in the only known convergence theorem of a discrete curvature estimator. The proof involves results from arithmetic properties of digital lines, digital convexity, combinatorics, continued fractions and random polytopes.

## 1 Introduction

Estimating geometric features of shapes or curves solely on their digitization is a classical problem in image analysis and pattern recognition. Some of the geometric features are global: area, perimeter, moments. Others are local: tangents, normals, curvature. Algorithms that performs this task on digitized objects are called . . . . . An interesting property these estimators should have is to converge towards the continuous geometric measure as the digitization resolution increases. However, few estimators have been proved to be convergent. In all works, shapes are generally supposed to have a smooth boundary and either to be convex or to have a finite number of inflexion points. The shape perimeter estimation has for instance been tackled in [12]. It proved the convergence of a perimeter estimator based on curve segmentation by maximal DSS. The speed of convergence of several length estimators has also been studied in [4]. Klette and Žunić [11] survey results about the convergence (and the speed of convergence) of several global geometric estimators. They show that discrete moments converge toward continuous moments.

As far as we know, there is only one work that deals with the convergence of local geometric estimators [3]. The symmetric tangent estimator appears to be convergent subject to an hypothesis on the growth of DSS as the resolution increases (see Conjecture 1). The same conjecture entails that a curvature estimator is convergent: it is based on DSS recognition and circumscribed circle computation (see Definition 5).

In this paper, we relate the number and the lengths of DSS to the number and lengths of edges of convex hulls of digitized shapes. Using arguments related to digital convex polygons and a theorem induced by random polytopes theory [1], we estimate the asymptotic behaviour of both quantities. We theoretically show that maximal DSS do not follow the conjecture used in [3]. Experiments confirm our result. The convergence theorem is thus not applicable to digital curves. As a consequence, the existence of convergent digital curvature estimators remains an open problem. The paper is organized as follows. First, we recall some standard notions of digital geometry and combinatoric representation of digital lines, i.e. patterns. The relations between maximal segments and edges of convex digital polygons are then studied to get bounds on maximal segments lengths and number. Finally, the asymptotic behaviour of maximal segments is deduced from the asymptotic behaviour of convex digital polygons. The growth of some DSS is thus proved to be too slow to ensure the convergence of curvature estimation. This theoretical result is further confirmed by experiments. Some proofs are omitted for limited space reason but may be found in [6].

## 2 Maximal Digital Straight Segments

We restrict our study to the geometry of 4-connected digital curves. A digital object is a set of pixels and its boundary in  $\mathbb{R}^2$  is a collection of vertices and edges. The boundary forms a 4-connected curve in the sense used in the present paper. Our work may easily be adapted to 8-connected curves. In the paper, all the reasoning are made in the first octant, but extends naturally to the whole digital plane. A set of successive points of the digital curve from index  $A$  to  $B$  by  $[C_A C_B]$  when no ambiguities are raised.

### 2.1 Standard Line, Digital Straight Segment, Maximal Segments

**Definition 1.** Let  $(x, y)$  be a point of a digital line.

$$\mu \leq ax - by < \mu + |a| + |b| \quad a \neq 0 \quad \mu \in \mathbb{Z}$$

standard line  $a/b = \mu$

The  $a/b = \mu$  are the 4-connected discrete lines. The quantity  $ax - by$  is called the  $\mu$  of the line. The points whose remainder is  $\mu$  (resp.  $\mu + |a| + |b| - 1$ ) are called upper (resp. lower) leaning points. The principal upper and lower leaning points are defined as those with extremal  $x$  values. Finite connected portions of digital lines define

Maximal segments form the longest possible DSS in the curve. They are essential when analyzing digital curves: they provide tangent estimations [7, 14],

they are used for polygonizing the curve into the minimum number of segments [8]. Any point belongs to at least one maximal segment.

## 2.2 Patterns and DSS

We here recall a few properties about, ... composing DSS and their close relations with continued fractions. They constitute a powerful tool to describe discrete lines with rational slopes [2, 9]. Since we are in the first octant, the slopes are between 0 and 1.

**Definition 2.** A pattern  $(a, b, \mu)$  is a set of points

reversed pattern

A pattern  $(a, b)$  embedded anywhere in the digital plane is obviously a DSS  $(a, b, \mu)$  for some  $\mu$ . Since a DSS contains at least either two upper or two lower leaning points, a DSS  $(a, b, \mu)$  contains at least one, ... or one, ... of characteristics  $(a, b)$ .

**Definition 3.** A simple continued fraction

$$z = a/b = [0, u_1, \dots, u_i, \dots, u_n]$$

$k$ -th convergent

$$z_k = \frac{p_k}{q_k} = [0, u_1, \dots, u_k]$$

$k+1$

There exists a recursive transformation for computing the pattern of a standard line from the, ... of its slope [2]. We call  $E$  the mapping from the set of positive rational number smaller than one onto Freeman-code's words defined as follows. First terms are stated as  $E(z_0) = 0$  and  $E(z_1) = 0^{u_1}1$  and others are expressed recursively:

$$E(z_{2i+1}) = E(z_{2i})^{u_{2i+1}} E(z_{2i-1}) \quad (1)$$

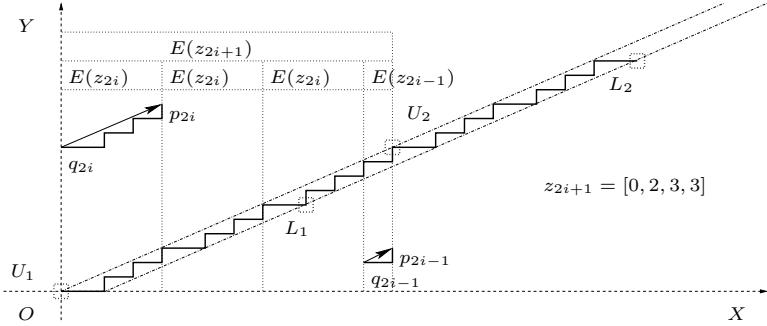
$$E(z_{2i}) = E(z_{2i-2}) E(z_{2i-1})^{u_{2i}} \quad (2)$$

In the following, the, ... of a pattern is the depth of its decomposition in simple continued fraction. We recall a few more relations:

$$p_k q_{k-1} - p_{k-1} q_k = (-1)^{k+1} \quad (3)$$

$$(p_k, q_k) = u_k (p_{k-1}, q_{k-1}) + (p_{k-2}, q_{k-2}) \quad (4)$$

We now focus on computing vector relations between leaning points (upper and lower) inside a pattern. In the following we consider a DSS  $(a, b, 0)$  in the first octant starting at the origin and ending at its second lower leaning point (whose coordinate along the  $x$ -axis is positive). We define  $a/b = z_n = [0, u_1, \dots, u_n]$  for some  $n$ . Points will be called  $U_1, L_1, U_2$  and  $L_2$  as shown in Fig. 1. We can state  $\mathbf{U}_1 \mathbf{L}_1 = \mathbf{U}_2 \mathbf{L}_2$  and  $\mathbf{U}_1 \mathbf{U}_2 = \mathbf{L}_1 \mathbf{L}_2 = (b, a)$ . We recall that the Freeman moves of  $[U_1 L_1]$  are the same as those of  $[U_2 L_2]$ . Furthermore  $[C_{U_1} C_{U_2}]$  form the, ...  $(a, b)$  and  $[C_{L_1} C_{L_2}]$  form the, ...  $(a, b)$ .



**Fig. 1.** A  $DSS(a, b, 0)$  with an odd complexity slope, taken between origin and its second lower leaning point

**Proposition 1.**

$$\begin{aligned} \mathbf{U}_1 \mathbf{L}_1 &= (u_{2i+1}-1)(q_{2i}, p_{2i}) + (q_{2i-1}, p_{2i-1}) + (1, -1) & \mathbf{L}_1 \mathbf{U}_2 &= (q_{2i}-1, p_{2i+1}) \\ &[U_1 L_1] \dots E(z_{2i})^{u_{2i+1}-1} & &[L_1 U_2] \\ &E(z_{2i-1})^{u_{2i}} \end{aligned}$$

**Proposition 2.**

$$\begin{aligned} \mathbf{U}_1 \mathbf{L}_1 &= (q_{2i-1}+1, p_{2i-1}-1) & \mathbf{L}_1 \mathbf{U}_2 &= (u_{2i}-1)(q_{2i-1}, p_{2i-1}) + (q_{2i-2}, p_{2i-2}) + \\ &(-1, 1) & &[U_1 L_1] \dots E(z_{2i-2})^{u_{2i-1}} \\ &[L_1 U_2] & &E(z_{2i-1})^{u_{2i-1}} \end{aligned}$$

### 3 Properties of Maximal Segments for Convex Curves

segments and digital edges of convex shape digitization. If  $S$  is a subset of  $\mathbb{R}^2$  its dilation by a real factor  $r$  is denoted by  $r \cdot S$ . Let  $\mathcal{D}_m$  be the digitization of step  $1/m$ :  $\mathcal{D}_m(S) = (m \cdot S) \cap \mathbb{Z}^2$ . We call  $\mathcal{L}^1$  the length estimator based on the city-block distance.

#### 3.1 Convex Digital Polygon (CDP)

**Definition 4.** convex digital polygon (CDP)  $\Gamma$  is a set of vertices  $(V_i)_{i=1..e}$  such that  $\Gamma = \mathcal{D}_1(\text{conv}(\Gamma))$ ,  $\Gamma = \mathcal{D}_1(\text{conv}(V_1, \dots, V_e))$ ,  $\Gamma = \bigcup_{i=1..e} \overline{V_i V_{i+1}}$ ,  $n_e(\Gamma) = e$ ,  $\text{Per}(\Gamma)$

A CDP is also called a lattice convex polygon [17]. An edge is the Euclidean segment joining two consecutive vertices, and a side is the discrete segment joining two consecutive vertices. It is clear that we have as many sides as edges and as vertices. From characterizations of discrete convexity [5], we clearly see that:

### Proposition 3.

We now recall one theorem concerning the asymptotic number of vertices of CDP that are digitization of continuous shapes. It comes from asymptotic properties of random polytopes.

### Theorem 1.

$$\mathcal{C}^3 \rightarrow \dots \rightarrow \mathcal{D}_m(S)$$

$$c_1(S)m^{\frac{2}{3}} \leq n_e(\mathcal{D}_m(S)) \leq c_2(S)m^{\frac{2}{3}}$$

$$\begin{array}{cc} & c_1(S) \quad c_2(S) \\ S & c_1 \quad c_2 \end{array}$$

### 3.2 Links Between Maximal Segments and Edges of CDP

Maximal segments are DSS: between any two upper (resp. lower) leaning points lays at least a lower (resp. upper) leaning point. The slope of a maximal segment is then defined by two consecutive upper and/or lower leaning points. Digital edges are patterns and their vertices are upper leaning points (from Proposition 3). Thus, vertices may be upper leaning points but never lower leaning points of maximal segments. Moreover a maximal segment cannot be strictly contained into a digital edge.

We call  $\dots$ , a digital edge whose two vertices define leftmost and rightmost upper leaning points of a maximal segment.

Following lemma gives relations between maximal DSS and digital edges:

**Lemma 1.**

the upper surface of the mandible, and the supporting edge

supporting edge

Lengths of maximal segments and digital edges are tightly intertwined, as shown by the two next propositions (Proposition 5 follows from Proposition 1 and 2).

**Proposition 4.**  $[V_k V_{k+1}]$  supporting edge,  $\frac{a}{b}$ ,  $f$ ,  $(a, b) \in MS$  maximal segment.

$$\frac{1}{3}\mathcal{L}^1(MS) \leq \mathcal{L}^1(V_k V_{k+1}) \leq \mathcal{L}^1(MS) \leq \frac{f+2}{f}\mathcal{L}^1(V_k V_{k+1}) - 2 \leq 3\mathcal{L}^1(V_k V_{k+1})$$

**Proposition 5.**  $MS_{k'}$

$$V_k$$

$$\mathcal{L}^1(MS_{k'}) \leq 4(\mathcal{L}^1(V_{k-1}V_k) + \mathcal{L}^1(V_kV_{k+1}))$$

A similar result related to linear integer programming is in [16]. It may also be obtained by viewing standard lines as intersection of two knapsack polytopes [10]. An elementary proof using pattern patterns is found in [6].

**Theorem 2.**  $E$

$$E$$

**Corollary 1.**

$$\begin{aligned} 2n+1 & z_n = [0, 2, \dots, 2] \\ m \times m & n \leq \log \frac{4m}{\sqrt{2}} / \log(1 + \sqrt{2}) - 1 \end{aligned}$$

The number  $L = [0, 2, \dots, 2, \dots]$  is a quadratic number equal to  $-1 + \sqrt{2}$ . Its recursive characterization is  $U_n = 2U_{n-1} + U_{n-2}$  with  $U_0 = 0$  and  $U_1 = 1$ . Solving it leads to  $U_n = \frac{\sqrt{2}}{4}((1 + \sqrt{2})^n - (1 - \sqrt{2})^n)$ . Hence asymptotically,  $U_n \approx \frac{\sqrt{2}}{4}(1 + \sqrt{2})^n$  and  $\lim_{n \rightarrow \infty} \frac{U_n}{U_{n+1}} = L$ .

The shortest edge of slope complexity  $n$  is clearly an  $n$ -th convergent of  $L$ . To fit into an  $m \times m$  grid, the complexity  $n$  is such that  $U_{n+1} \leq m$ . We thus obtain that  $n \leq \log \frac{4m}{\sqrt{2}} / \log(1 + \sqrt{2}) - 1$ .  $\square$

### 3.3 Asymptotic Number and Size of Maximal Segments

We assume in this section that the digital convex polygon  $\Gamma$  is enclosed in a  $m \times m$  grid. We wish to compute a lower bound for the number of edges related to at least one maximal segment. We show in Theorem 3 that this number is significant and increases at least as fast as the number of edges of the DCP divided by  $\log m$ . From this lower bound, we are able to find an upper bound for the length of the smallest maximal segment of a DCP (Theorem 4). We first label each vertex of the DCP as follows: (i) a  $\square$  is an upper leaning point of a supporting edge, (ii) a  $\square$  is an upper leaning point of some maximal segment but is not a 2-vertex, (iii)  $\square$  are all the remaining vertices. The number of  $i$ -vertices is denoted by  $n_i$ . Given an orientation on the digital contour, the number of edges going from an  $i$ -vertex to a  $j$ -vertex is denoted by  $n_{ij}$ .

**Theorem 3.**  $\Gamma$

$$\frac{n_e(\Gamma)}{\Omega(\log m)} \leq n_1 + 2n_{22}. \quad (5)$$

$$n_e(\Gamma)/\Omega(\log m).$$

From Theorem 2 and its Corollary 1, we know that a DSS hence a maximal segment cannot include more than  $\Omega(\log m)$  edges. Hence there cannot be more than  $\Omega(\log m)$  0-vertices for one 1-vertex or for one 2-vertex. We get  $n_{00} \leq (n_1 + n_2)\Omega(\log m)$ . We develop the number of edges with each possible label:  $n_e(\Gamma) = n_{22} + n_{02} + n_{12} + n_{20} + n_{21} + n_{00} + n_{01} + n_{10} + n_{11}$ . Since,  $n_{02} + n_{12} \leq n_{22}$ ,  $n_{20} + n_{21} \leq n_{22}$  and  $n_{01} + n_{10} + n_{11} \leq 3n_1$ , we get  $n_e(\Gamma) \leq 3n_{22} + n_{00} + 3n_1$ . Noting that a 2-vertex cannot be isolated by definition of supporting edges gives  $n_2 \leq 2n_{22}$ . Once inserted in  $n_{00} \leq (n_1 + n_2)\Omega(\log m)$  and compared with  $n_e(\Gamma)$ , we get the expected result.  $\square$

We now relate the DCP perimeter to the length of maximal segments.

**Theorem 4.**

$$\min_l \mathcal{L}^1(MS_l) \leq \Omega(\log m) \frac{\text{Per}(\Gamma)}{n_e(\Gamma)}. \quad (6)$$

We have  $\text{Per}(\Gamma) = \sum_{n_e} \mathcal{L}^1(E_i)$ . We now may expand the sum on supporting edges (22-edges), on edges touching a 1-vertex, and on others. Edges touching 1-vertices may be counted twice, therefore we divide by 2 their contribution to the total length.

$$\sum_{n_e} \mathcal{L}^1(E_i) \geq \sum_{n_{22}} \mathcal{L}^1(E_j^{22}) + \frac{1}{2} \sum_{n_1} \mathcal{L}^1(E_{k-1}^{?1}) + \mathcal{L}^1(E_k^{1?}) \quad (7)$$

For the first term, each supporting edge indexed by  $j$  (a 22-edge) has an associated maximal segment, say indexed by  $j'$ . From Proposition 4, we know that  $\mathcal{L}^1(E_j^{22}) \geq \frac{1}{3} \mathcal{L}^1(MS_{j'})$ . For the second term, each 1-vertex indexed by  $k$  is an upper leaning point of some maximal segment indexed by  $k'$ . Proposition 5 holds and  $\mathcal{L}^1(E_{k-1}^{?1}) + \mathcal{L}^1(E_k^{1?}) \geq \frac{1}{4} \mathcal{L}^1(MS_{k'})$ . Substitutions in Eq. (7) bring:

$$\sum_{n_e} \mathcal{L}^1(E_i) \geq \frac{1}{3} \sum_{n_{22}} \mathcal{L}^1(MS_{j'}) + \frac{1}{8} \sum_{n_1} \mathcal{L}^1(MS_{k'}) \geq \frac{1}{8}(n_1 + 2n_{22}) \min_l \mathcal{L}^1(MS_l)$$

Inserting the lower bound of Theorem 3 concludes.  $\square$

## 4 Asymptotic Properties of Shapes Digitized at Increasing Resolutions

We may now turn to the main interest of the paper: studying the asymptotic properties of discrete geometric estimators on digitized shapes. We therefore consider a plane convex body  $S$  which is contained the square  $[0, 1] \times [0, 1]$  (w.l.o.g.). Furthermore, we assume that its boundary  $\gamma = \partial S$  is  $\mathcal{C}^3$  with everywhere strictly positive curvature. This assumption is not very restrictive since people are mostly interested in regular shapes. Furthermore, the results of this section remains valid if the shape can be divided into a ... number of convex

and concave parts; each one is then treated separately. The digitization of  $S$  with step  $1/m$  defines a digital convex polygon  $\Gamma(m)$  inscribed in a  $m \times m$  grid. We first examine the asymptotic behavior of the maximal segments of  $\Gamma(m)$ , both theoretically and experimentally. We then study the  $\dots, \dots, \dots, \dots$  of a discrete curvature estimator.

#### 4.1 Asymptotic Behavior of Maximal Segments

The next theorem summarizes the asymptotic size of the smallest maximal segment wrt the grid size  $m$ .

**Theorem 5.**

$$\Gamma(m)$$

$$\min_i \mathcal{L}^1(MS_i(\Gamma(m))) \leq \Omega(m^{1/3} \log m) \quad (8)$$

Theorem 4 gives for the DCP  $\Gamma(m)$  the inequality  $\min_i \mathcal{L}^1(MS_i(\Gamma(m))) \leq \Omega(\log m) \frac{\text{Per}(\Gamma(m))}{n_e(\Gamma(m))}$ . Since  $\Gamma(m)$  is convex included in the subset  $m \times m$  of the digital plane, its perimeter  $\text{Per}(\Gamma(m))$  is upper bounded by  $4m$ . On the other hand, Theorem 1 indicates that its number of edges  $n_e(\Gamma(m))$  is lower bounded by  $c_1(S)m^{2/3}$ . Putting everything together gives  $\min_i \mathcal{L}^1(MS_i(\Gamma(m))) \leq \Omega(\log m) \frac{4m}{c_1(S)m^{2/3}}$  which is once reduced what we wanted to show.  $\square$

Although there are points on a shape boundary around which maximal segments grow as fast as  $O(m^{1/2})$  (the critical points in [13]), some of them do not grow as fast. A closer look at the proofs of Theorem 4 shows that a significant part of the maximal segments (at least  $\Omega(1/(\log m))$ ) has an average length that grows no faster than  $\Omega(m^{1/3} \log m)$ . This fact is confirmed with experiments. Fig. 2, left, plots the size of maximal segments for a disk digitized with increasing resolution. The average size is closer to  $m^{1/3}$  than to  $m^{1/2}$ .

#### 4.2 Asymptotic Convergence of Discrete Geometric Estimators

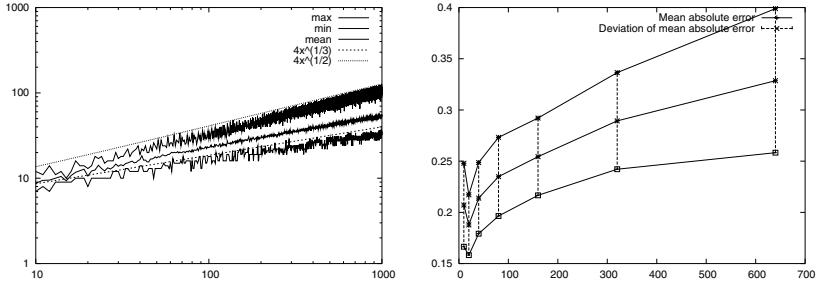
A useful property that a discrete geometric estimator may have is to converge toward the geometric quantity of the continuous shape boundary when the digitization grid gets finer [3, 4, 11].

We now recall the definition of a discrete curvature estimator based on DSS recognition [3].

**Definition 5.**

$P$   $Q$   $R$   $m$   $Q$   $R$   $P$  half-tangents curvature estimator by circumcircle  $\hat{\kappa}(P)$

Experiments show that this estimator rather correctly estimates the curvature of discrete circles ( $\approx 10\%$  error). It is indeed better than any other curvature estimators proposed in the litterature. Theorem B.4 of [3] demonstrates the  $\dots, \dots, \dots, \dots$  of this curvature estimator, subject to the conjecture:



**Fig. 2.** For both curves, the digitized shape is a disk of radius 1 and the abscissa is the digitization resolution. Left: plot in log-space of the  $\mathcal{L}^1$ -size of maximal segments. Right: plot of the mean and standard deviation of the absolute error of curvature estimation,  $|\hat{\kappa} - 1|$  (expected curvature is 1)

[3] Half-tangents on digitized boundaries grow at a rate of  $\Omega(m^{1/2})$  with the resolution  $m$ .

However, with our study of maximal segments, we can state that

Conjecture 1 is ... verified for digitizations of  $\mathcal{C}^3$ -curves with strictly positive curvature. We cannot conclude on the asymptotic convergence of the curvature estimator by circumcircle.

It is enough to note that half-tangents, being DSS, are included in maximal segments and may not be longer. Furthermore, since maximal segments cover the whole digital contour, some half-tangents will be included in the smallest maximal segments. Since the smallest maximal segments are no longer than  $\Omega(m^{1/3} \log m)$  (Theorem 5), the length of some half-tangents has the same upper bound, which is smaller than  $\Omega(m^{1/2})$ .  $\square$

The asymptotic convergence of a curvature estimator is thus still an open problem. Furthermore, precise experimental evaluation of this estimator indicates that it is most certainly not asymptotically convergent, although it is actually on average one of the most stable discrete curvature estimator (see Fig. 2, right). Former experimental evaluations of this estimator were averaging the curvature estimates on all contour points. The convergence of the average of all curvatures does not induce the convergence of the curvature at one point.

## 5 Conclusion

We show in this paper the relations between edges of convex hulls and maximal segments in terms of number and sizes. We provide an asymptotical analysis of the worst cases of both measures. A consequence of the study is the refutation of an conjecture related to the asymptotic growth of maximal segments and which was essential in proving the convergence of a curvature estimator based

on DSS and circumcircles [3]. Our work also applied to digital tangents since their convergence relies on the same conjecture. The existence of a convergent discrete estimator of curvature based on DSS is thus still a challenging problem and we are currently investigating it.

## References

1. Antal Balog and Imre Bárány. On the convex hull of the integer points in a disc. In *SCG '91: Proceedings of the seventh annual symposium on Computational geometry*, pages 162–165. ACM Press, 1991.
2. J. Berstel and A. De Luca. Sturmian words, lyndon words and trees. *Theoret. Comput. Sci.*, 178(1-2):171–203, 1997.
3. D. Coeurjolly. *Algorithmique et géométrie pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lyon 2, Décembre 2002.
4. D. Coeurjolly and R. Klette. A comparative evaluation of length estimators of digital curves. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 26(2):252–257, 2004.
5. Chul E.Kim. Digital convexity, straightness, and convex polygons. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 6(6):618–626, 1982.
6. J.-O. Lachaud F. de Vieilleville and F. Feschet. Maximal digital straight segments and convergence of discrete geometric estimators. Research Report 1350-05, LaBRI, University Bordeaux 1, Talence, France, 2005.
7. F. Feschet and L. Tougne. Optimal time computation of the tangent of a discrete curve: application to the curvature. In *Discrete Geometry and Computer Imagery (DGCI)*, volume 1568 of *LNCS*, pages 31–40. Springer Verlag, 1999.
8. F. Feschet. and L. Tougne. On the Min DSS Problem of Closed Discrete Curves. In A. Del Lungo, V. Di Gesù, and A. Kuba, editors, *IWCIA*, volume 12 of *Electronic Notes in Discrete Math.* Elsevier, 2003.
9. G.H. Hardy and E.M. Wright. *An introduction to the theory of numbers*. Oxford University Press, fourth edition, 1960.
10. A. S. Hayes and D. C. Larman. The vertices of the knapsack polytope. *Discrete Applied Mathematics*, 6:135–138, 1983.
11. R. Klette and J. Žunić. Multigrid convergence of calculated features in image analysis. *Journal of Mathematical Imaging and Vision*, 13:173–191, 2000.
12. V. Kovalevsky and S. Fuchs. Theoretical and experimental analysis of the accuracy of perimeter estimates. In Förster and Ruwiedel, editors, *Proc. Robust Computer Vision*, pages 218–242, 1992.
13. J.-O. Lachaud. On the convergence of some local geometric estimators on digitized curves. Research Report 1347-05, LaBRI, University Bordeaux 1, Talence, France, 2005.
14. J.-O. Lachaud, A. Vialard, and F. de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In E. Andrès, G. Damiand, and P. Lienhardt, editors, *Proc. Int. Conf. Discrete Geometry for Computer Imagery (DGCI'2005), Poitiers, France*, LNCS. Springer, 2005. To appear.
15. J.-P. Réveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. Thèse d'etat, Université Louis Pasteur, Strasbourg, 1991.
16. V. N. Shevchenko. On the number of extreme points in linear programming. *Kibernetika*, 2:133–134, 1981. In russian.
17. K. Voss. *Discrete Images, Objects, and Functions in  $\mathbb{Z}^n$* . Springer-Verlag, 1993.

# Improving the Maximum-Likelihood Co-occurrence Classifier: A Study on Classification of Inhomogeneous Rock Images

P. Paclík, S. Verzakov, and R.P.W. Duin

Information and Communication Theory Group,  
Delft University of Technology,

Mekelweg 4, 2628 CD Delft, The Netherlands

{P.Paclík, S.Verzakov, R.P.W.Duin}@ewi.tudelft.nl

<http://www-ict.et.tudelft.nl/~pavel>

**Abstract.** An industrial rock classification system is constructed and studied. The local texture information in many image patches is extracted and classified. The decisions made at the local level are fused to form the high-level decision on the image/rock as a whole. The main difficulties of this application lay in significant variability and inhomogeneity of local textures caused by uneven rock surfaces and intrusions. Therefore, an emphasis is paid to the derivation of informative representation of local texture and to robust classification algorithms. The study focuses on the co-occurrence representation of texture comparing the two frequently used strategies, namely the approach based on Haralick features and methods utilizing directly the co-occurrence likelihoods. Apart of maximum-likelihood (ML) classifiers also an alternative method is studied considering the likelihoods to prototypes as feature of a new space. Unlike the ML methods, a classifier built in this space may leverage all training examples. It is experimentally illustrated, that in the rock classification setup the methods directly using the co-occurrence estimates outperform the feature-based techniques.

## 1 Introduction

This study is concerned with classification of rocks in an industrial application in which a rock is placed beneath a light source and imaged by a color camera. Based on a high-resolution color image, the rock is as a whole assigned to a class of interest. In our application, the association of a rock specimen to a class depends on local texture properties of the rock surface. The nature of industrial rock classification poses several challenges on the design of a pattern recognition system. Because the rock surfaces are uneven the local texture information may exhibit considerable variability over a single image. Moreover, the rocks contain intrusions, which further increase the multi-modal nature of the class description.

In this study, we partition the design of the rock classification system into three steps: the derivation of an informative representation of local texture, the training of a local patch classifier and, eventually, the combination of per-patch decisions into a single decision on the entire image/rock. We focus on an

investigation of the first two steps and fix the combination strategy by using the majority vote over the classification results in a set of local image patches. We do not consider the use of color information in this study and work solely with gray-level textures.

There exists abundant literature on the representation of texture for the sake of classification [1, 2, 3]. In our study, we have focused on one of the most commonly used strategies based on co-occurrence matrices [4]. A co-occurrence matrix (CM) estimates the gray-level dependencies in a local neighborhood for a given displacement step and angle. Two major approaches for building classifiers using the gray-level co-occurrences are used. The majority of studies assume that the rich texture description present in the co-occurrence matrix must be first reduced to a set of features because the original co-occurrence distribution is too large to be used directly for classification [4, 5]. Typically only a subset of the original features, proposed by Haralick in [4], is used [1, 6]. Principally different approach was adopted by Vickers and Modestino [7] who consider the co-occurrence entries directly as features. The decision is derived by a maximum-likelihood classifier (ML), operating on co-occurrence matrices estimated per-class. Instead of using the class prototypes<sup>1</sup>, Ojala et.a. [8] derived separate prototype from each image patch in the training set. Therefore, the resulting ML classifier mimics rather the nearest neighbor approach.

It is an open question which of the two major strategies is beneficial in the rock classification problem. Focusing more on the likelihood-based methods, we observe that the ML-based classifiers effectively derive their decisions only from the stored prototypes of the class or local co-occurrences. The existing abundant collections of training image patches are used only for the estimation of the class co-occurrences, or even entirely discarded apart of local prototypes in the nearest-neighbor sense. In order to fully leverage the existing training sets, we also adopt a recently developed strategy for building classifiers on (dis)similarity representations (Duin et.al. [9, 10, 11]). Here the likelihoods to prototypes are considered as dimensions of a new space, which is populated by all available training examples. A general-purpose classifier, such as the Fisher linear discriminant (FLD), built in this space may thereby exploit the correlations between the likelihoods to prototypes. Additional advantage of this approach is that apart of likelihoods also other dissimilarity measures for probability distributions may be used, such as the Kullback-Leibler divergence. We illustrate, that this classification strategy is beneficial for multi-modal rock classification problems as it facilitates derivation of non-linear classifiers.

In the next section, we introduce the rock classification system. Section 3 explains how the discussed data representation and classifiers may be built. In Section 4 we describe a set of experiments on a dataset of rock images. Finally, in the last section, we conclude our findings.

---

<sup>1</sup> In order to emphasize that the co-occurrence models are estimated from the training data, we adopt the pattern recognition terminology in this paper and refer to prototypes.

## 2 The Rock Classification System

The rock classification system is trained using a set of labeled images. Because the classification is performed on the basis of local textures, the set of local image patches must be first extracted from the training images. Each image patch is accompanied with the information on the class of rock it represents. The set of all labeled image patches is processed so that the local texture information captured in each patch is properly represented. In this paper, we represent the local textures by the co-occurrence estimates. For the sake of classification, this intermediate texture representation must be transformed accordingly. It is either further reduced to a set of feature values or represented by a set of likelihoods to prototype co-occurrences. In this representation, the classifier is finally built using the training set of labeled local patches.

When processing of a new image by the trained rock classification system, a set of image regions is first extracted. The texture within each patch is represented by the co-occurrence matrix and the classifier-aware representation is derived similarly to the training stage. The trained patch classifier is invoked on each of the image patches and its decisions are fused by the majority voting combiner to a decision on the level of the complete image/rock.

## 3 Likelihood-Based Classification of Local Textures

The co-occurrence matrix  $\mathbf{P}$ , estimated from the image patch  $r$  given the displacements  $\Delta x$  and  $\Delta y$  is defined as:

$$P(g_1, g_2) = \{\text{pairs}(g_1, g_2) | r(x, y) = g_1 \text{ and } r(x + \Delta x, y + \Delta y) = g_2\}, \quad (1)$$

where  $g_1$  and  $g_2$  denote the gray-levels,  $r(x, y)$  represents the gray-level at coordinates  $x$  and  $y$  in the image patch  $r$  and the functional *pairs* returns the number of situations.

The likelihood of a co-occurrence estimate  $\mathbf{P}$  with respect to the prototype co-occurrence  $\mathbf{Q}$  may be expressed as:

$$L(\mathbf{P}, \mathbf{Q}) = \sum_{g_1, g_2} P(g_1, g_2) \ln \frac{Q(g_1, g_2)}{\sum_{g_1, g_2} Q(g_1, g_2)} \quad (2)$$

The  $C$ -class maximum-likelihood classifier assigns a new observation  $\mathbf{P}$  to the class of the closest prototype:

$$\omega(\mathbf{P}) = \arg \max_c \max_k \{L(\mathbf{P}, \mathbf{Q}_k^c)\}, \quad (3)$$

where  $\omega(\mathbf{P})$  denotes the class of the observation  $\mathbf{P}$  and  $\mathbf{Q}_k^c$  represents the  $k$ -th prototype of the class  $\omega_c$ ,  $c = 1, \dots, C$ .

Traditionally, the ML classifier represents each class by a single prototype. This is estimated by averaging the training population of the local co-occurrences for each class [7]. This approach is analogous to the nearest mean classification.

The ML classifier may also utilize more prototypes per class each of them directly selected from the training set [8]. This strategy resembles the nearest-neighbor classifier.

### 3.1 Training the Classifier on the Likelihood Representation

The alternative classification scheme considers the likelihoods computed with respect to a set of  $M$  prototypes as dimensions of a new  $M$ -dimensional feature space where a co-occurrence estimate  $\mathbf{P}$  may be represented by the vector:

$$\mathbf{x}^L = \{L(\mathbf{P}, \mathbf{Q}_m)\}_{m=1}^M, \quad \mathbf{x}^L \in R^M. \quad (4)$$

In order to train a classifier in this representation, the space is populated using a set of likelihoods corresponding to  $N_p$  training image patches. The training set will, therefore, contain  $N_p$  labeled feature vectors  $\mathbf{x}^L$  in the  $M$ -dimensional space. We propose to train the Fisher linear discriminant (FLD) in this new space. For the two-class situation it becomes:

$$\omega(\mathbf{P}) = \text{sign} \left( \sum_m^M w_k L(\mathbf{P}, \mathbf{Q}_m) + b \right). \quad (5)$$

In the multi-class situation the minimum least square solution may be used, based on regression [12]. Note that the class membership of prototypes is not directly used by this classification rule, on contrary to the ML classifier (3).

The Equations (2) and (5) may be rearranged to show that the linear discriminant built in the space spanned by likelihoods delivers linear solution with respect to the original space defined by the bins of the co-occurrence distribution. The proposed linear discriminant yields, in fact, as opposed to the maximum-likelihood classifier (3). A non-linear classifier may be constructed by leveraging a different distance measure for probability distributions, such as the Kullback-Leibler divergence (KL) [13]:

$$d_{KL}(\mathbf{P}, \mathbf{Q}) = \sum_{g_1, g_2} P(g_1, g_2) \ln \frac{P(g_1, g_2)}{Q(g_1, g_2)} + \sum_{g_1, g_2} Q(g_1, g_2) \ln \frac{Q(g_1, g_2)}{P(g_1, g_2)}. \quad (6)$$

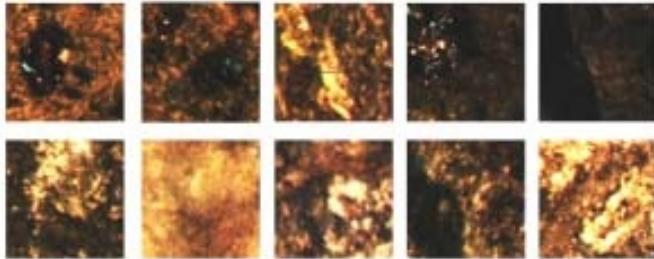
By comparing the Equations (6) and (2) we can observe the close relation between both measures. Each of the sums in the Kullback-Leibler divergence (6) may be decomposed into entropy and likelihood terms.

## 4 Experiments

### 4.1 Dataset Description

The dataset used in our experiments contains 72 color images from two classes of rocks, each represented by 36 images with resolution of 1392 times 1040 pixels. Images were acquired under controlled lighting conditions. Apart from cleaning, the rocks remain untreated with no cutting or polishing applied. Each image, therefore, views a multi-faceted rock surface. The Figure 1 shows several patches

extracted from the images. Patches in each row were extracted from images of the same class. Note that the patches exhibit significant inhomogeneity and intensity variation caused by uneven rock surfaces and frequent intrusions.



**Fig. 1.** Examples of image patches for each of the two classes. Each row corresponds to examples of one class

#### 4.2 Texture Descriptions

We consider the following texture descriptions based on co-occurrence:

- Co-occurrence bins are directly considered as features.
- 14 Haralick features computed from the co-occurrence matrix.
- Likelihoods with respect to co-occurrence prototypes considering either mean class prototypes or local prototypes as discussed in Section 3.
- Kullback-Leibler (KL) divergences to local co-occurrence prototypes.

For the sake of comparison, we also include two base methods widely used for texture classification:

Local Binary Patterns (LBP) describe a distribution of binary patterns constructed by thresholding a micro-region and accumulating this information over an entire image patch [14]. The LBP features were found to be considerably more robust to varying illumination than the techniques using the gray-levels directly. Three types of LBP features are computed with 50 features in total.

Gabor Features A bank of 24 Gabor filters is designed according to [15]. A total set of 48 features is formed by means and standard deviations of the filter responses computed over an image patch.

#### 4.3 Data Representation and Classification

For the sake of simplicity, we employ the random selection of prototypes. This allows us to estimate the classifier performance for a growing number of prototypes.

While the class co-occurrence prototypes are derived from thousands of observations, each of the local co-occurrences is estimated from a single image patch. Due to the limited amount of data, some bins of the local co-occurrence histogram may remain empty. As it can be seen from Equation (2), the likelihood of any observation to the prototype with an

empty bin becomes infinitely small. In order to avoid situations where a test object cannot be classified because the likelihoods to all prototypes are infinitely small, we introduce the regularized likelihood measure using prototypes:

$$\mathbf{Q}_{\text{reg}} = \mathbf{Q}(1 - \delta) + \delta G^{-1} \sum_{g_1, g_2} \mathbf{Q}(g_1, g_2), \quad (7)$$

where  $G$  denotes the total number of co-occurrence bins and  $\delta$  stands for the regularization parameter. For the Kullback-Leibler divergence (6), both the observed and prototype co-occurrences  $\mathbf{P}$  and  $\mathbf{Q}$  are regularized. The regularization parameter was fixed to  $\delta = 10^{-6}$  in all the experiments.

In several experiments, the supervised version of PCA was employed which estimates the pooled covariance matrix using the per-class covariances [12]. In all cases 0.99 of variance was preserved.

In the experiments, standard classifiers were used such as the nearest neighbor classifier (1-NN), Fisher linear discriminant (FLD) or quadratic classifier assuming normal densities (QDC) [12]. We refer to the maximum-likelihood classifier with class prototypes as NMC (nearest mean classifier).

#### 4.4 Experimental Setup

In order to estimate the performance of the rock classification system utilizing different texture characterization and classifiers we have adopted a 10-fold cross-validation procedure over images. Thereby, all the local patches originating from one image, appear always together either in training set or in the test set. All the steps required for training of the image classifier i.e. the estimation of the co-occurrences, computation of likelihoods or dissimilarities, extraction of texture features and the training of the patch classifiers is carried only on the images in the training set.

Vickers [7] and Valkealahti [16] propose to equalize the histograms of local image patches. This procedure assumes that the first-order statistics of the local histogram are not informative and thus may be removed. Experiments on our dataset show that the mean and standard deviation of the gray-level computed over local neighborhoods carry important discriminatory information. We have, therefore, decided not to perform any histogram equalization and rather maintain as constant illumination as possible during the acquisition process.

The original color images were converted to gray-level. All the methods used image patches of  $64 \times 64$  pixels. The patches are extracted without mutual overlap so that each image is represented by cca 330 image regions. In each fold of the cross-validation procedure, about 20 000 local patches are used for training and about 2 600 image patches during testing. Prior to the estimation of co-occurrence matrices, the histograms of the local regions are reduced from 256 to 8 levels. All co-occurrence matrices are estimated for one pixel displacement in the vertical direction [4].

## 4.5 Results and Discussion

The Table 1 summarizes our rock classification experiments. For each method, three error measure are given, namely the classification error over image patches, the error over images obtained by majority voting and the total number of test images ( $e$ ), misclassified during the cross-validation.

**Table 1.** Results of the rock classification experiment with 72 images. Estimated mean errors over image patches and over images are given with the respective standard deviations of the mean estimators. The last column  $e$  denotes the total number of erroneously classified images in the cross-validation experiment

<i>method</i>	<i>texture description</i>	<i>classifier</i>	<i>error (patches)</i> $\hat{\mu} (\hat{\sigma}_\mu)$	<i>error (images)</i> $\hat{\mu} (\hat{\sigma}_\mu)$	<i>e</i>
1	CM, ML, class proto.	NMC	0.327 (0.023)	0.258 (0.040)	19
2	CM, ML, regul., 10 proto.	1-NN	0.416 (0.035)	0.371 (0.044)	27
3	CM, ML, regul., 50 proto.	1-NN	0.380 (0.020)	0.238 (0.052)	17
4	CM, ML, regul., 200 proto.	1-NN	0.362 (0.012)	0.188 (0.045)	14
5	CM, likelihood, class proto.	FLD	0.328 (0.022)	0.258 (0.032)	19
6	CM, likelihood, regul., 10 proto.	FLD	0.314 (0.021)	0.217 (0.043)	16
7	CM, likelihood, regul., 50 proto.	FLD	0.306 (0.020)	0.229 (0.049)	17
8	CM, likelihood, regul., 200 proto.	FLD	0.305 (0.020)	0.217 (0.043)	16
9	CM, KL, regul., 200 proto.	1-NN	0.354 (0.012)	0.167 (0.029)	12
10	CM, KL, regul., 10 proto.	FLD	0.304 (0.022)	0.167 (0.039)	12
11	CM, KL, regul., 50 proto.	FLD	0.269 (0.018)	0.125 (0.039)	9
12	CM, KL, regul., 200 proto.	FLD	0.271 (0.017)	0.125 (0.039)	9
13	CM, directly used as features	FLD	0.305 (0.021)	0.229 (0.049)	17
14	CM, 14 Haralick features	FLD	0.316 (0.024)	0.221 (0.045)	16
15	CM, 14 Haralick features, PCA	QDC	0.378 (0.018)	0.354 (0.032)	26
16	LBP, 50 features	FLD	0.371 (0.021)	0.196 (0.047)	14
17	LBP, 50 features, PCA 0.99	QDC	0.363 (0.018)	0.225 (0.047)	16
18	Gabor filers, 48 features	FLD	0.345 (0.029)	0.283 (0.053)	21
19	Gabor filers, 48 features, PCA 0.99	QDC	0.355 (0.024)	0.271 (0.052)	20

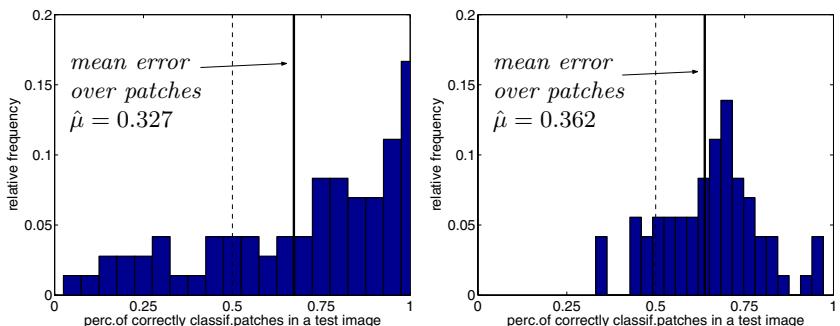
First part of the table corresponds to the likelihood-based methods. The per-image performance of the traditionally-used ML classifier with class prototypes (row 1) can be improved by utilizing the local co-occurrence prototypes in the nearest neighbor fashion (rows 2-4). Note, that while the method 4 reaches lower per-image error than the classifier 1, its error over patches remains higher.

In order to understand this behaviour, one needs to focus on the voting combiner based on the ratio of correctly classified patches within an test image. The distribution of this ratio computed for all the test images is depicted in Figure 2 for the ML classifier 1 (on the left) and the nearest neighbor rule 4 (on the right). Although exhibiting lower error over patches, the model-based ML classifier 1 suffers (on the image level) from the problem heterogeneity i.e. from the existence of images that are very different from the model but still belong

to the same class. On the contrary, the nearest neighbor rule 4 is able to cope with the problem multi-modality.

This point is illustrated again on the performance of the FLD classifiers trained on the likelihood representation (methods 6-8). Compared to the nearest neighbor rules based on the same prototypes (rows 2-4) the classifiers 6-8 exhibit significantly better per-patch performances. This is understandable as the methods 6-8 utilize all available 20 000 training examples in the 200D spaces while the methods 2-4 use only the 200 training examples. The per-image error of the nearest-neighbor classifiers 2-4, however, decreases significantly faster and, eventually, the method 4 even outperforms the FLD classifier 8. As mentioned in Section 3.1, the FLD classifier based on likelihoods is still a linear discriminant, similarly to the ML classifier 1. Therefore, we attribute the better performance of the weaker nearest neighbor classifier 4 to its non-linear nature.

The rows 9-12 in Table 1 refer to classifiers based on the Kullback-Leibler divergence. Based on the identical sets of prototypes as the likelihood-based classifier 4, the nearest neighbor rule 9 yields a better performance. Further improvements are possible by training the FLD classifiers on the Kullback-Leibler divergences as all the available training examples are now exploited. The algorithms in rows 11 and 12 yield the best overall results in our study (12.5% of error over images). We conclude that this improvement over the likelihood-based methods 6-8 is caused by the non-linearity introduced by the Kullback-Leibler divergence.



**Fig. 2.** The effect of the voting combiner on two likelihood-based classifiers: the linear ML classifier 1 (left) and the nearest neighbor rule 4 (right). For each of the test images in cross-validation, the ratio of correctly classified patches is computed. The figures depict the histogram of this ratio over all 72 images. Although the mean error over all patches is lower for the classifier on the left, the fraction of images misclassified by voting (left of the dashed line) is higher

The rows 13-15 refer to the feature-based methods utilizing directly the co-occurrence matrices (13) or Haralick features (14,15). It is interesting to note that the co-occurrence bins directly used as features (13) yield better result than the traditionally used ML classifier 1. Employing the Haralick features results in

an additional minor performance improvement. From the results, we conclude that the use of quadratic classifier in a PCA-reduced representation (15) is not beneficial.

The rows 16-19 represent methods based on different principles than co-occurrence, namely the Local Binary Pattern features (Ojala et.al. [14]) and the features derived from the bank of Gabor filters. While the Gabor features deliver only mediocre performance, the LBP features yield better or comparable results than the best likelihood-based approaches. The LBP features are designed to be more resilient to variable illumination than the co-occurrence matrices. Based on the similar performance of both approaches we conclude that the illumination in our dataset is rather constant.

## 5 Conclusions

The rock classification employs a local texture description in order to classify the high-resolution images into a set of pre-defined classes. Unlike the existing studies which deal with images of inhomogeneous but polished and cut rocks [17, 18], the images used in this study depict rocks with uneven surfaces and intrusions. The local textures extracted from a single image therefore exhibit significant variability. The aim of this paper was to understand which texture representations and what types of classifiers are robust and well-performing for this type of problem. We have focused on the family of texture representations based on the co-occurrence matrix investigating two distinct approaches to texture characterization. The first derives the Haralick features from the co-occurrence matrices and employs a conventional classifier. The second approach leverages the co-occurrence estimates directly and is traditionally bundled with the maximum-likelihood classifier.

The maximum-likelihood classifier operating in the nearest neighbor fashion on the local co-occurrence estimates appears to be beneficial to the traditionally employed ML classifier utilizing the class-specific co-occurrences. Although weaker in classifying individual patches, the nearest neighbor classifier yields a non-linear class-separation boundary. The traditional maximum-likelihood classifier operates on the nearest mean principle and, therefore, cannot cope well with the multi-modal rock-classification problem.

The maximum-likelihood classifiers use the available training set only for estimation of class prototypes (the nearest mean scenario) or even entirely discard all the training examples apart of prototypes (the nearest neighbor approach). The proposed alternative classifier is derived by training the Fisher linear discriminant on likelihoods to prototypes. It uses all the available training examples and leverages correlations between the likelihoods (or dissimilarities) to prototypes. While this method still yields a linear classifier when applied to likelihoods, the use of other distance measures such as the Kullback-Leibler divergence results in a non-linear decision rule. The dissimilarity-based classifiers built using the Kullback-Leibler divergence also deliver the highest performance of all studied approaches.

Surprisingly, the linear classifier built on Haralick features outperforms the state-of-the art maximum-likelihood classifier operating directly on co-occurrences. However, the more sophisticated treatment of dissimilarity representations yields systems, performing significantly better than any of the feature-based classifiers. We conclude that while the Haralick features are simple and provide good accuracy, the dissimilarity-based classifiers offer higher flexibility and eventually better performance.

## Acknowledgments

The authors would like to thank to David Tax, Carmen Lai and Thomas Landgrebe for support and stimulating discussions on the manuscript. This research is/was supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Ministry of Economic Affairs.

## References

1. Ohanian, P.P., Dubes, R.C.: Performance Evaluation for Four Classes of Textural Features. *Pattern Recognition* **25** (1992) 819–833
2. Haralick, R.M.: Statistical and Structural Approaches to Texture. *Proceedings of the IEEE* **67** (1979) 786–804
3. Randen, T., Husøy, J.H.: Filtering for texture classification: A comparative study. *IEEE Trans. Patt.Anal. and Mach.Int.* **21** (1999) 291–310
4. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. *IEEE Trans. Syst.Man and Cybernetics* **3** (1973) 610–621
5. Cohen, F.S., Fan, Z., Patel, M.A.: Classification of rotated and scaled textured images using gaussian markov random field models. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 192–202
6. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Classification of non-homogenous textures by combining classifiers. In: Proc.of IEEE Int.Conf. on Image Processing, Barcelona, Spain, September 14.-17. Volume 1. (2003) 981–984
7. Vickers, A.L., Modestino, J.W.: A maximum likelihood approach to texture classification. *IEEE Trans. Pattern Analysis and Machine Intelligence* **4** (1982)
8. Ojala, T., Pietikäinen, M., Kyllönen, J.: Gray level cooccurrence histograms via learning vector quantization. In: Proc. 11th SCIA, Kangerlussuaq, Greenland. (1999) 103–108
9. Duin, R.P.W., de Ridder, D., Tax, D.M.J.: Experiments with object based discriminant functions; a featureless approach to pattern recognition. *Pattern Recognition Letters* **18** (1997) 1159–1166
10. Pekalska, E., Paclík, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity based classification. *Journal of Machine Learning Research* **1** (2001) 175–211 Special Issue "New Perspectives on Kernel Based Learning Methods".
11. Paclík, P., Duin, R.P.W.: Dissimilarity-based classification of spectra: computational issues. *Real Time Imaging* **9** (2003) 237–244
12. Duin, R.P.W., Juszczak, P., de Ridder, D., Paclík, P., Pekalska, E., Tax, D.M.J.: PR-Tools 4.0, a Matlab toolbox for pattern recognition. Technical report, ICT Group, TU Delft, The Netherlands (2004) <http://www.prtools.org>.

13. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press (1998) ISBN 0-12-686140-4.
14. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence* **24** (2002) 971–987
15. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.* **18** (1996) 837–842
16. Valkealahti, K., Oja, E.: Reduced multidimensional co-occurrence histograms in texture classification. *IEEE Trans. Patt.Anal. and Mach.Int.* **20** (1998) 90–94
17. Partio, M., Cramariuc, B., Gabbouj, M., Visa, A.: Rock texture retrieval using gray-level co-occurrence matrix. In: ORSIG-2002, 5th Nordic Signal Processing Symposium, On Board Hurtigruten M/S Trollfjord, Norway, October 4–7. (2002)
18. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Rock image classification using non-homogenous textures and spectral imaging. In: WSCG proc., WSCG'2003, Plzen, Czech Republic, Feb. 3.–7. (2003)

# The Tangent Kernel Approach to Illumination-Robust Texture Classification

S. Verzakov, P. Paclík, and R.P.W. Duin

Information and Communication Theory Group

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Mekelweg 4, 2628 CD Delft, The Netherlands

s.verzakov, p.paclik, r.p.w.duin@ewi.tudelft.nl

**Abstract.** Co-occurrence matrices are proved to be useful tool for the purpose of texture recognition. However, they are sensitive to the change of the illumination conditions. There are standard preprocessing approaches to this problem. However, they are lacking certain qualities. We studied the tangent kernel SVM approach as an alternative way of building illumination-robust texture classifier. Testing on the standard texture data has shown promising results.

## 1 Introduction

Often, it is impractical to keep experimental conditions strictly constant or redo full sensor recalibration. Imprecisions in calibration causes poorer generalization of the recognition system. Such effects can be compensated by increasing of learning sets sizes or by special data treatment: preprocessing or (which is somewhat similar to that) building robust recognition systems.

In this work we focus on building a texture classification system robust to the variability in illumination conditions. The common procedure to deal with this problem is to perform the full equalization of image histogram or mere the contrast stretching. Applying these techniques, one must decide what is the standard histogram form. It is not easy to figure out if the chosen one suits well for the purpose of the discrimination between different types of textures. Also, these methods may still leave some amount of illumination fluctuations because of the variability in data.

We propose an alternative approach which consists in a modification of a similarity measure between textures which is robust to the changes in the illumination. As a texture description we use co-occurrence matrices [1, 2] and employ tangent kernel SVM [3, 4] in order to built a robust classifier.

Our approach can be applied to any histogram-like type of data: biomedical data (histograms of DNA content), and normalized spectra. Basically, it may be used with any data represented by non-negative features with fixed sum of elements and suffering from the imprecise (linear) calibration.

The rest of the paper is structured as follows. The next section contains short review of tangent kernel SVM. Then, in section 3 it is shown how this approach

can be applied to the illumination-robust texture recognition. Section 4 contains the description of data set and discussion of the results of numerical experiments. In section 5 we conclude our work.

## 2 Tangent Kernel SVM Technique

For reader convenience we provide a short account of ideas from [3, 4]. Suppose that two-class classification problem has to be solved: having an object  $\mathbf{x} \in \mathbb{R}^d$ , a label  $y \in \{-1, +1\}$  should be assigned, defining, to which one of the two classes it belongs. In other words, the task consists of building a classification function

$$y = f(\mathbf{x}) : \mathbb{R}^d \longrightarrow \{-1, +1\}$$

Here we assume that  $f(\mathbf{x}) = \text{sign}(g(\mathbf{x}))$ , and  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  is the smooth discriminant function.

Suppose also that we have a prior knowledge that a one-parametric transformation  $\mathcal{L}_t : \mathbb{R}^d \longrightarrow \mathbb{R}^d$ ,  $t \in \mathbb{R}$  does not change the class membership of the object. To simplify the task one can demand more stronger condition: an invariance of discriminant function  $g$

$$g(\mathcal{L}_t \mathbf{x}) - g(\mathbf{x}) = 0$$

Assuming that  $\mathcal{L}_t$  satisfies

$$\begin{aligned} \mathcal{L}_0 \mathbf{x} &= \mathbf{x} \\ \mathcal{L}_{t_1} \mathcal{L}_{t_2} &= \mathcal{L}_{t_1+t_2} \end{aligned} \tag{1}$$

invariance property can be reformulated in a differential form

$$\frac{\partial g(\mathcal{L}_t \mathbf{x})}{\partial t} \Big|_{t=0} = 0$$

which transforms after some algebra into

$$\begin{aligned} \partial_{\mathbf{x}}^T g(\mathbf{x}) \mathcal{M} \mathbf{x} &= 0 \\ \mathcal{M} &\equiv \frac{\partial \mathcal{L}_t}{\partial t} \Big|_{t=0} \\ \partial_{\mathbf{x}} &\equiv \left( \frac{\partial}{\partial x^{(1)}}, \dots, \frac{\partial}{\partial x^{(d)}} \right)^T \end{aligned} \tag{2}$$

The condition (2) is supposed to be valid for all  $\mathbf{x}$  from the data domain. Thus, it puts constraint on the possible choice of  $g$ .

Another approximate approach of taking into account Eq. (2) consists in the adding a penalty term

$$\begin{aligned} r(g; \mathbf{X}) &\equiv \frac{1}{2} \sum_i [\partial_{\mathbf{x}_i}^T g(\mathbf{x}_i) \mathcal{M} \mathbf{x}_i]^2 = \sum_i \partial_{\mathbf{x}_i}^T g(\mathbf{x}_i) \mathbf{C}_i \partial_{\mathbf{x}_i} g(\mathbf{x}_i) \\ \mathbf{C}_i &\equiv (\mathcal{M} \mathbf{x}_i)(\mathcal{M} \mathbf{x}_i)^T \end{aligned}$$

to the original learning criterion by which minimization  $g$  meant to be found (e.g. inverse margin, noise to signal ratio, etc.). This transforms original minimization task into

$$\begin{aligned} g^* &= \arg \min_g R_\gamma(g; \mathbf{X}, \mathbf{y}) \\ R_\gamma(g; \mathbf{X}, \mathbf{y}) &= (1 - \gamma)R(g; \mathbf{X}, \mathbf{y}) + \gamma r(g; \mathbf{X}) \\ \gamma &\in [0, 1] \end{aligned} \quad (3)$$

where  $R$  is the original learning criterion,  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$  is the training data set,  $\mathbf{y} = (y_1, \dots, y_N)^T$  contains labels of training objects and parameter  $\gamma$  defines how is the penalty term important with regard to the  $R$ .

If we decide that  $g$  has linear  $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$  and choose as a learning algorithm the SVM [5] with  $R = \frac{1}{2} \|\mathbf{w}\|^2$ , then the modified minimization criterion takes a form

$$R_\gamma = \frac{1 - \gamma}{2} \|\mathbf{w}\|^2 + \frac{\gamma}{2} \mathbf{w}^T \mathbf{C} \mathbf{w}$$

$$\mathbf{C} = \sum_i \mathbf{C}_i$$

By substitution

$$\begin{aligned} \tilde{\mathbf{w}} &= [(1 - \gamma)\mathbf{I} + \gamma\mathbf{C}]^{1/2} \mathbf{w} \\ \tilde{\mathbf{x}} &= [(1 - \gamma)\mathbf{I} + \gamma\mathbf{C}]^{-1/2} \mathbf{x} \end{aligned}$$

we recover the original form of the SVM criterion:  $R_\gamma = \frac{1}{2} \|\tilde{\mathbf{w}}\|^2$ . So, to implement this technique we need only to redefine the matrix of inner products (kernel matrix)

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \tilde{\mathbf{x}}^T \tilde{\mathbf{y}} = \mathbf{x}^T [(1 - \gamma)\mathbf{I} + \gamma\mathbf{C}]^{-1} \mathbf{y} \quad (4)$$

If there is overlap between classes, then a soft-margin version of SVM algorithm should be used. The modifications are straightforward and do not change Eq. (4). For the sake of brevity we will use LTK-SVM (Linear Tangent Kernel SVM) abbreviation to name the soft margin SVM algorithm with kernel defined by Eq. (4). The extension of this approach to the non-linear discriminant functions  $g$  is possible [3] but it is beyond of the scope of this paper.

### 3 Derivation of the Tangent Kernel for n-Dimensional Histograms

Suppose that the input of the recognition system is a continues distribution density function of an  $n$ -dimensional random vector  $\boldsymbol{\eta}$  such that

$$\boldsymbol{\eta} = e^t \boldsymbol{\xi}(\boldsymbol{\theta})$$

where  $\xi$  is a measurement made on a perfectly calibrated device,  $t$  is a parameter responsible for the imprecisions in sensor calibration and  $\theta$  is a set of parameters which defines intra- and inter-class variability.

The distribution of the "observed"  $\eta$  can be expressed in terms of the distribution of "ideal"  $\xi$  as

$$\rho_\eta(z_1, \dots, z_n) = e^{-nt} \rho_\xi(e^{-t} z_1, \dots, e^{-t} z_n)$$

To achieve better classification rates we need to built classifier robust to the data transformations caused by the operator

$$\mathcal{L}_t^{sc} \rho(z_1, \dots, z_n) = e^{-nt} \rho(e^{-t} z_1, \dots, e^{-t} z_n)$$

This operator satisfies conditions Eq. (1) and thus, we can employ tangent kernel SVM. By taking the first derivative of  $\mathcal{L}_t^{sc} \rho$  at  $t = 0$ , we find that

$$\mathcal{M}^{sc} \rho(z_1, \dots, z_n) = -n \rho(z_1, \dots, z_n) - \sum_{j=1}^n z_j \partial_{z_j} \rho(z_1, \dots, z_n)$$

In practice one deals with histograms (e.g. co-occurrence matrices) not with distribution densities. Assuming that  $\mathbf{P} = (P_{k_1, \dots, k_n})$  such a multi-dimensional histogram (properly normalized to be an estimation of a density function) we redefine the  $\mathcal{M}^{sc}$  operator as

$$(\mathcal{M}^{sc} \mathbf{P})_{k_1, \dots, k_n} = -n P_{k_1, \dots, k_n} - \sum_{j=1}^n z_j^{(k_j)} (\Delta_j \mathbf{P})_{k_1, \dots, k_n}$$

Here,  $z_j^{(k_j)}$  are the centers of histogram bins in the  $j$ -th direction and  $\Delta_j$  is the operator taking the (smoothed) finite differences of the histogram  $\mathbf{P}$  in the the same direction  $j$ . Assuming that  $\mathbf{u}(\mathbf{P})$  is unfolding of an  $n$ -dimensional array into a column vector, it is possible to write down the penalty term

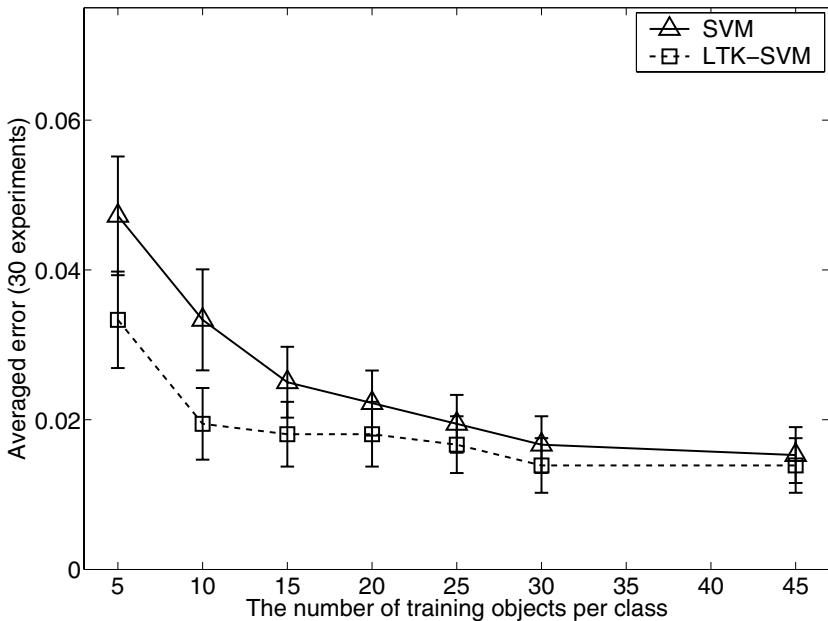
$$\begin{aligned} r &= \mathbf{w}^T \mathbf{C}^{sc} \mathbf{w} \\ \mathbf{C}^{sc} &= \sum_i \mathbf{u}(\mathcal{M}^{sc} \mathbf{P}_i) \mathbf{u}(\mathcal{M}^{sc} \mathbf{P}_i)^T \end{aligned}$$

Thus, the new similarity measure reads as

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \{(1 - \gamma) \mathbf{I} + \gamma \mathbf{C}^{sc}\}^{-1} \mathbf{y}$$

## 4 Numerical Experiments

As data for our experiments we used Brodatz textures 1.3.04 and 1.3.05 from the [6]. Each image of 1024-by-1024 size and 8-bit depth was split into 64 128-by-128 non-overlapping patches. To imitate the change in the illumination conditions each patch as a whole was multiplied by the randomly generated number uniformly distributed in the region  $[1 - \alpha; 1]$ . The  $\alpha$  values in the range from 0 to 0.5

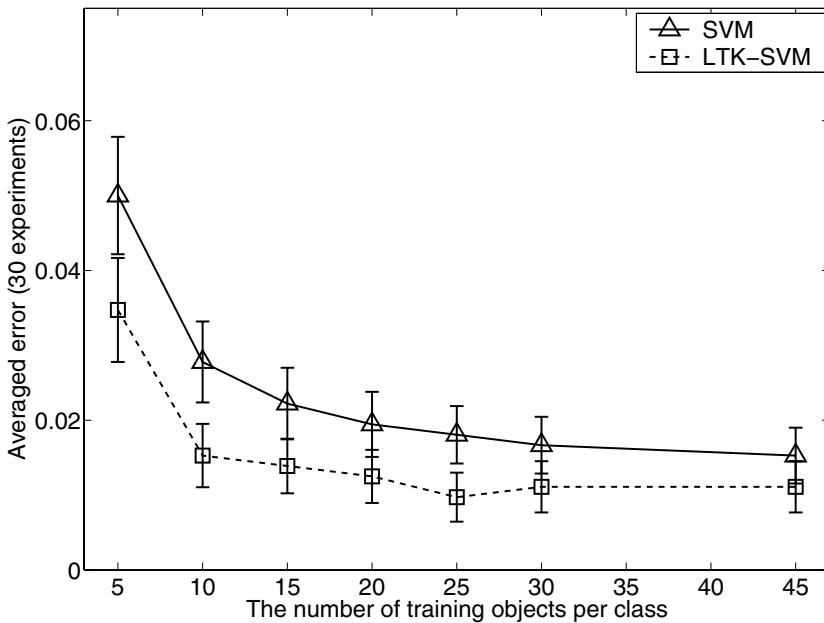


**Fig. 1.** Learning curves for 16-by-16 co-occurrence matrices data set.  $\alpha = 0$

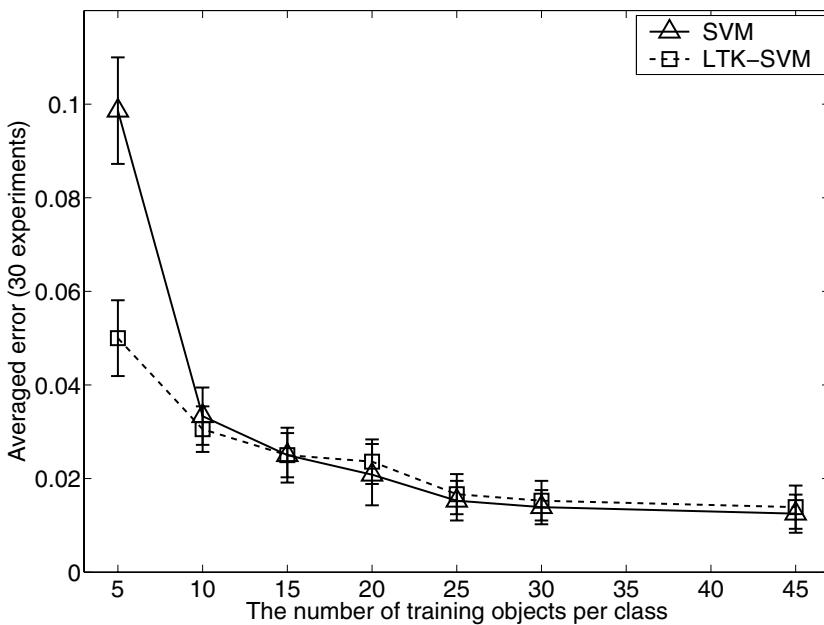
were used to create a number data sets. Afterwards, for each such data set we computed 128 (by the number of patches) 2-dimensional co-occurrence matrices of 16-by-16 and 32-by-32 sizes which served as an input of classifiers. Neither contrast stretching nor histogram equalization was applied to data in any way in all experiments presented in this paper.

We studied the difference in the performance of the conventional linear SVM classifier and its tangent kernel version. To see how useful the usage of prior knowledge can be for different sizes of training set the learning curves were computed. The results were obtained by averaging over 30 hold-out experiments: for each experiment we randomly took out 20% of all objects, the rest 80% objects were used as the training pool. The learning curves were obtained for each hold-out experiment by training classifiers on the sequence of the nested training sets generated from the training pool.

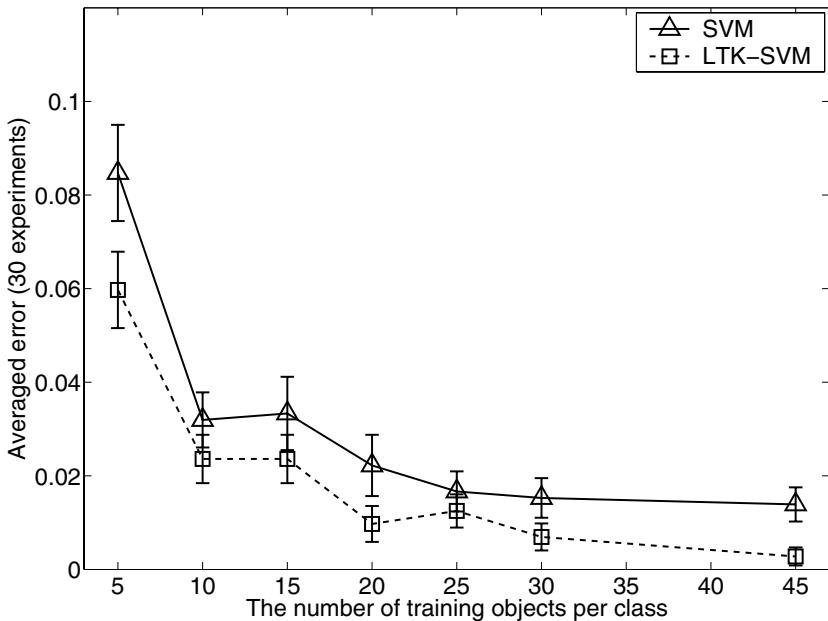
In all experiments the  $\nu$  regularization parameter [7] of SVM/LTK-SVM classifier was optimized by grid search over the set of predefined values. For each candidate value the preliminary classifier was trained on 75% data randomly selected for the training procedure. The other 25% were used to measure the performance and select the actual  $\nu$  value. Using this value, the final classifier was trained on the whole currently available training data set. Linear tangent kernel SVM was trained at a number of  $\gamma$  values (no internal optimization). For the smoothing of finite differences we used Savitsky-Golay filter of the first order. The size of the filter window  $w$  was always set to 3.



**Fig. 2.** Learning curves for 16-by-16 co-occurrence matrices data set.  $\alpha = 0.1$



**Fig. 3.** Learning curves for 16-by-16 co-occurrence matrices data set.  $\alpha = 0.5$



**Fig. 4.** Learning curves for 32-by-32 co-occurrence matrices data set.  $\alpha = 0.5$

Figures 1, 2 and 3 show the learning curves of the conventional linear SVM and LTK-SVM ( $\gamma = 0.95$ ) on the 16-by-16 co-occurrence matrices. It can be seen that even at  $\alpha = 0$  (original illumination of images) the applying of modified similarity leads to the better classification rates. The effect is more recognizable at  $\alpha = 0.1$ . However, larger variability in the illumination ( $\alpha = 0.5$ ) cannot be compensated by the use of LTK-SVM.

On the other hand, LTK-SVM can cope with the such an amount of illumination variability being applied to the 32-by-32 co-occurrence matrices (Fig. 4). Probably, this effect can be explained by the fact that the similarity measure being derived for continuous distributions gives better results for histograms with finer bins.

## 5 Conclusions

We studied the possibility of application of the tangent kernel approach to the stabilization of illumination conditions for better texture classification. The tangent kernel approach shows significant advantages at smaller sample sizes comparing to the conventional SVM. Unlike the preprocessing methods, proposed technique can be applied directly to the co-occurrence matrices even when raw images are unavailable. The use of the tangent kernel approach does not require

to select invariant features or select standard form of image histogram. All this makes it to be a promising tool in many practical situations. However, there are open questions: e.g. what is the optimal strategy to select the tradeoff parameter  $\gamma$  or how necessary and convenient may be the exploitation of nonlinear kernels. Definitely, more study is needed including testing on the real-world data.

## Acknowledgments

This research was supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Ministry of Economic Affairs.

## References

1. R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *3(6):610–621*, November 1973.
2. R.M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, *67:786–804*, 1979.
3. B. Schölkopf. *Support Vector Learning*. PhD thesis, Munich, 1997.
4. B. Schölkopf, P. Y. Simard, A. J. Smola, and V. N. Vapnik. Prior knowledge in support vector kernels. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 640–646, Cambridge, MA, 1998. MIT Press.
5. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, USA, 2000.
6. USC-SIPI Image Database.
7. B. Schölkopf, A. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, *12:1207–1245*, 2000.

# Tissue Models and Speckle Reduction in Medical Ultrasound Images

Radim Kolář and Jiří Jan

Department of Biomedical Engineering, FEEC,  
Brno University of Technology, Czech Republic  
[kolarr@fieec.vutbr.cz](mailto:kolarr@fieec.vutbr.cz)

**Abstract.** This paper presents a new method for speckle noise reduction in medical ultrasound images. It is based on the statistical description of the envelope of ultrasound signal by the virtue of the Nakagami-m distribution. Parameter of this distribution is used to adjust an adaptive filter.

## 1 Introduction

Ultrasound tissue modeling can provide an important information that can be used for diagnosis, image segmentation, interpretation or visualization. There are many tissue models based on various kind of probability distribution describing tissue from more or less complex way - Rayleigh or Rice distribution [4], K-distribution [3] or Nakagami distribution. We will focus on the Nakagami distribution [7, 8] that is able to distinguish between various kind of tissue conditions.

The aim of this paper is to describe a statistical based approach to ultrasound image processing (Section 2) and to utilize Nakagami distribution (Section 3) for speckle suppression (Section 4). Therefore, the speckles are considered as a noise makes the manual/automatic image segmentation difficult. Section 5 presents some results and discussion.

## 2 Echo Model

The echo signal can be considered as a sum of backscattered and backreflected single echoes from a number of scattering points and strong reflectors in the tissue [3]. We can express this echo signal (in one point/time) with the help of phasor notation. Each scatter reflects  $x_k$  amount of signal with the phase shift  $\theta_k$  (due to the random location)

$$X = \sum_{i=0}^{N-1} x_i \cdot e^{j\theta_i}. \quad (1)$$

In most cases the amplitude  $x_i$  can be considered deterministic and we can rewrite (1) as

$$X = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \alpha_i e^{j\theta_i}, \quad (2)$$

where  $\alpha_i = \frac{x_i}{\sqrt{N}}$  are normalized to  $\sqrt{N}$ . This normalization means that the scattered amplitudes from each echo is equal. Mean values,  $E\{\cdot\}$ , of the real and imaginary parts can be considered as a zero [3]:  $E\{X_r\} = E\{X_i\} = 0$ . This holds for independent and identical distributions for  $\alpha_i$  and  $\theta_i$ . For this condition these mean values are zero, because  $\theta_i$  is assumed to be equally distributed within the interval  $[-\pi, \pi]$ . For the second moments, it can be shown that:

$$E\{X_i\} = E\{X_r\} = \sigma^2. \quad (3)$$

Now, we express the radiofrequency signal as a time signal by involving  $\omega_0$ :

$$s(t) = X_r(t).cos(\omega_0 t) + j.X_i(t).sin(\omega_0 t). \quad (4)$$

With the clinical scanner we can obtain only the  $\dots$  (the first) component. For envelope detection we assume the analytical nature of this signal and the  $\dots$  (the second) component is obtain by the virtue of the Hilbert transform ( $HT$ ). The envelope is simply obtained by

$$S(t) = \sqrt{X_r^2(t) + X_i^2(t)} = \sqrt{X_r^2(t) + HT\{X_r(t)\}^2}. \quad (5)$$

### 3 Nakagami Distribution

We would like to describe various tissues conditions using only one universal mathematical model. This may be done using the Nakagami distribution. The probability density function of the envelope,  $f(S)$ , under the Nakagami model is given by [7]

$$f(S) = \frac{2m^m S^{2m-1}}{\Gamma(m)\Omega^m} \cdot e^{-\frac{m}{\Omega}S^2} \quad (6)$$

where  $m$  is the Nakagami parameter and  $\Omega$  is the scaling parameter. The SNR value for Nakagami distributed envelope is given by:

$$SNR_N = \frac{mean}{standard deviation} = \frac{E[S]}{\sqrt{E[S^2] - E[S]^2}} = \frac{1}{\sqrt{m[\frac{\Gamma(m)}{\Gamma(m+0.5)}]^2 - 1}} \quad (7)$$

Now, we will explore this distribution for various values of  $m$  in the connection with physical conditions [7].

Consider the first the case, where large number of scatters is present within the resolution cell and there is no correlation between them. For this case  $m = 1$

and Nakagami distribution changes to Rayleigh distribution with one parameter  $\Omega$ :

$$f(S) = \frac{2S}{\Omega} \cdot e^{-\frac{S^2}{\Omega}} \quad (8)$$

where  $E(S^2) = \Omega$  and represents reflected power. For this case the  $SNR_N = 1.91$ .

Consider now a case with presence of subresolvable periodic structure (we cannot distinguish between particular scatters) with spacing  $\lambda/2$ , where  $\lambda$  corresponds to the wavelength and a collection of randomly located scatters. Under these condition the envelope will be Rician distributed with a density function

$$f(S) = \frac{S}{\sigma^2} \cdot e^{-\frac{S^2+S_0^2}{2\sigma^2}} I_0\left(\frac{S \cdot S_0}{\sigma}\right) \quad (9)$$

where  $S_0$  represents a coherent component (the mean value of the inphase component arising from  $\lambda/2$  spacing) and  $I_0(.)$  is the zero-order modified Bessel function.  $\sigma$  is a quantity describing random (diffuse) component [6]. The relation between the Nakagami parameter, and Rician parameters is:

$$\frac{1}{m} = 1 + \frac{S_0^4}{(2\sigma^2 + S_0^2)^4} \quad (10)$$

The ratio  $SNR_R = \frac{S_0}{\sigma}$  indicates the SNR. Thus the equation (10) can be rearranged to

$$m = \frac{1 + SNR_R^2 + \frac{1}{4}SNR_R^4}{1 + SNR_R^2}. \quad (11)$$

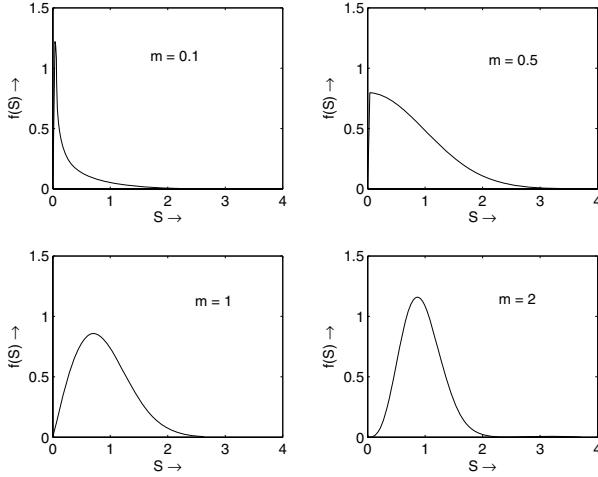
As the  $SNR_R$  increases the  $m$  increases too (the subresolvable structure is more strong than diffuse scatters) because of the four power in the numerator. The  $SNR_N$  will achieve values higher than 1.91.

Consider another case of scattering structure - the spacing of periodic subresolvable structure is less than  $\lambda/4$ . The pdf for this case is

$$f(S) = \frac{S}{\sigma_1 \sigma_2} \cdot e^{-\left(\frac{S^2}{4\sigma_1^2} + \frac{S^2}{4\sigma_2^2}\right)} I_0\left[\frac{S^2}{4} \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)\right] \quad (12)$$

Signal-to-noise  $SNR_N$  ratio for signal coming from this distribution is less than 1.91 (Rayleigh value). As the number of periodic located scatters becomes smaller the SNR approach to 1.91 and  $m$  approach to 1. The connection between the Nakagami and generalized Rician distribution is for values  $0.5 < m < 1$  [7].

In some applications the value of  $m$  is constrained such that  $m \geq 0.5$ . This corresponds to the half-Gaussian shape of the pdf (see Fig.1). However, the  $m$ -value can be less than 0.5 in ultrasound application [7]. This situation corresponds to the very low scatter density. The shape of pdf is more sharp with heavy tail. Four pdf plots of the Nakagami distribution are shown on Fig.1.



**Fig. 1.** Plots of the Nakagami pdf for different values of  $m$  and  $\Omega = 1$

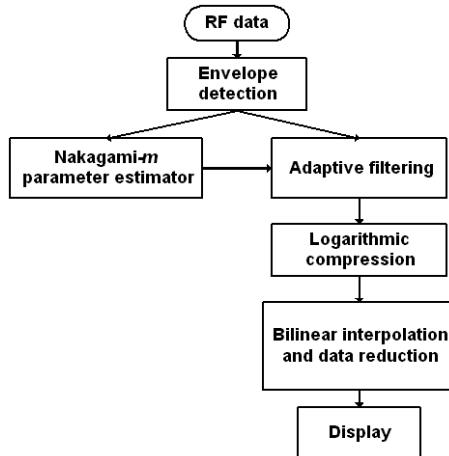
## 4 Model Based Filtering

The idea for model-based filtering comes from the basic model for adaptive filter [1]:

$$y(i, j) = \bar{x} - k(i, j) \cdot (\bar{x} - x(i, j)), \quad (13)$$

where  $y(i, j)$  is a pixel of the output image,  $\bar{x}$  is the mean of the window sample and  $k(i, j)$  stands for adaptive factor that controls the filtering operation. Speckles are in fact an impulsive noise and it can be eliminated more effective by median filter rather than mean filter. Therefore the value  $\bar{x}$  is replaced by median value. The key parameter of this filtering concept is to find  $k(i, j)$ , which is usually based on signal-to-noise ratio value. We try to design this parameter based on the  $m$  parameter from the Nakagami pdf.

The filtering operation is performed on the non-interpolated envelope image. This means that we don't use an interpolated samples, but only original data (before scan conversion, data reduction and logarithmic compression) (see Fig.2). Let's go over the above mentioned cases for tissue. First, assume the low number of scatters such that  $m < 0.5$ . This occurs in medium with very low density (e.g. blood) and also at the boundary with different acoustic impedances (specular reflections). In this case there should be no filtering, because of edge blurring. Therefore the  $k$  should be adjusted to 1. The second case is for  $0.5 < m < 1$ . The tissue has periodic unresolvable structure with spacing less than  $\frac{\lambda}{2}$ . The speckles are present and we should perform some kind of filtering. Therefore  $k$  should be 1 for  $m = 0$  (no filtering) and for  $m$  close to 1,  $k$  should be close to 0. Situation for  $m = 1$  corresponds to Rayleigh (random) scattering and  $k = 0$  (maximum filtering).



**Fig. 2.** Algorithm for adaptive filtering using Nakagami parameter

**Table 1.** Relation between m and k parameter

Nakagami-m	Tissue	Parameter k
$m < 0.5$	<i>low scatters density or specular reflection</i>	$k=1$
$0.5 \leq m < 1$	<i>unresolvable structure with spacing <math>&lt; \frac{\lambda}{2}</math></i>	$k = 1$ for $m = 0.5$ and $k$ is close to 0 for $m$ close to 1
$m=1$	<i>random scattering</i>	$k=1$
$m > 1$	<i>unresolvable structure with spacing <math>\frac{\lambda}{2}</math></i>	$k$ increase with increasing $m$

The last situation is for tissues with subresolvable periodic structure with spacing  $\lambda/2$ . As the ratio of this structure to random scatters rises, the  $m$  rises too. Therefore the filter must operate such that  $k$  is close to 0 for  $m$  close to 1 and with increasing value of  $m$  (and increasing SNR) the  $k$  should fall to 0 (minimum filtering). These conditions are summarized in Tab.1.

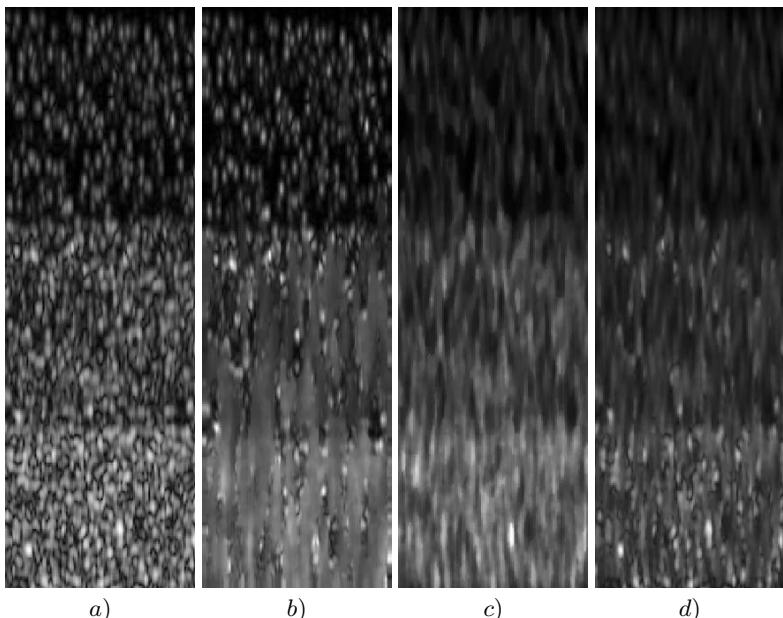
The quality of estimator for the  $m$ -parameter is an important part for Nakagami based filter. More details can be found in [5], where the comparison of several estimator is made. We've used an approximating estimator as proposed by Greenwood and Durand in 1960.

Next problem arises with the size of the window. There are in fact two windows: one for  $m$  parameter estimation and the second window for filtering. We use the same window size for both cases. For too small window we obtain only few samples to estimate the parameter  $m$ . For the large window we decrease the spatial resolution. The choice is trade-off between this two negative properties. It seems reasonable to not to decrease the lateral resolution, because in nowadays ultrasound scanners this resolution is poor in comparison with the

axial resolution. Therefore the minimum window size in lateral direction is 3 (an odd number for simple computing). The resolution in axial direction is better and corresponds to the wavelength and central frequency of the ultrasound wave  $\lambda = \frac{c}{f}$ . Assuming  $c = 1540m.s^{-1}$  and  $f_0 = 2MHz$  the wavelength is  $0.77mm$  and the resolution can be considered as a  $\lambda/2$ . Axial resolution also depends on the pulse length (number of cycles), which is usually about 4. The corresponding number of samples depends on the sampling frequency ( $f_s = 20MHz$  in our case) and is given by  $n = \frac{4}{2} \frac{f_0}{f_s} = 20$ . The window size should be higher than this value because of obtaining representative sample to determine the  $m$  parameter. Several window sizes were tested and as an appropriate size is between 19 and 35 samples. This match with the length from  $1.33mm$  to  $2.67mm$ .

## 5 Filtering Results and Discussion

The filtering performance was tested on the simulated images with different scatterers densities. A one-dimensional discrete scattering model was used in our simulation [5]. Fig. 3a) shows simulated ultrasound image with random scatter density 2, 5, 20 within resolution cell. Fig. 3b) shows filtering result after applying the proposed method( Nakagami filter ). One can see that the smoothing depends on the scatter densities. For low scatter density (upper part of the image) there is almost no smoothing. Middle and lower part of the image

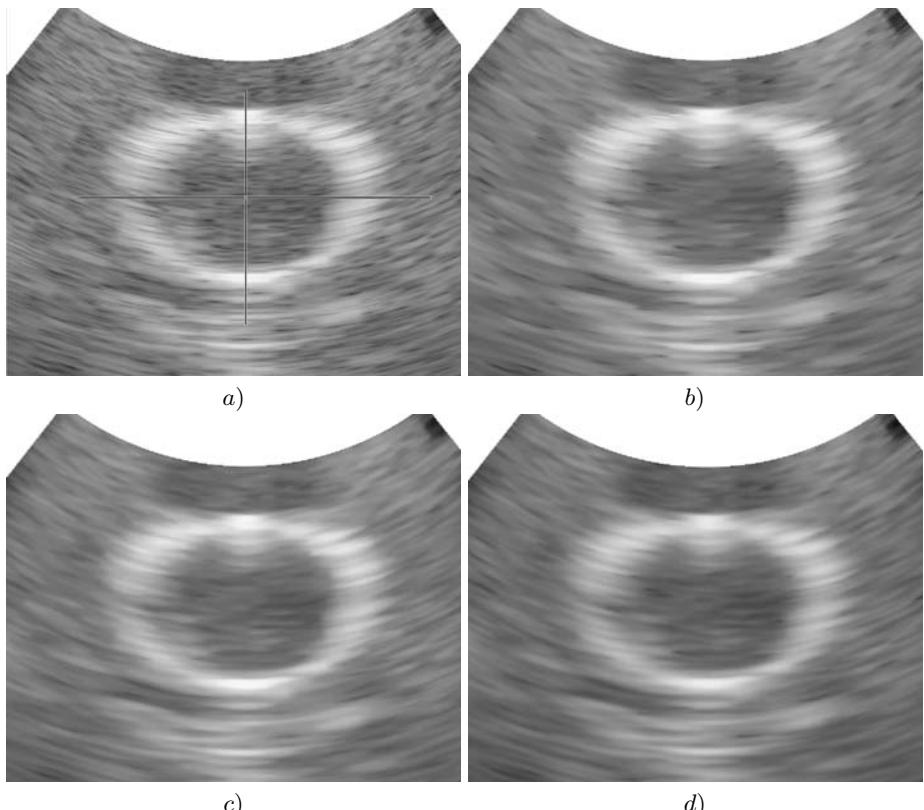


**Fig. 3.** Simulated image with different scatter densities: a) Original image , Filtered image by b) the proposed method, c) median filter, d) Wiener filter

**Table 2.** The SI Values for Simulated Image

<i>region</i>	<b>1</b>	<b>2</b>	<b>3</b>
<i>original</i>	0.91	1.61	1.84
<i>Nakagami filter</i>	0.96	2.94	3.45
<i>Median filter</i>	1.30	4.1	4.2
<i>Wiener filter</i>	1.9	3.80	3.10

are more smoothed. If some stronger reflection is presence it can be seen that the smoothing performance is decreased. The proposed method was compared with the simple median filter and adaptive Wiener filter as implemented in Matlab. This Wiener filter estimates the local mean and variance around each pixel within the sliding window and compute the estimate of new sample [9]. The speckles can be quantified by the means of speckle index (SI) evaluated for homogenous region [2]:

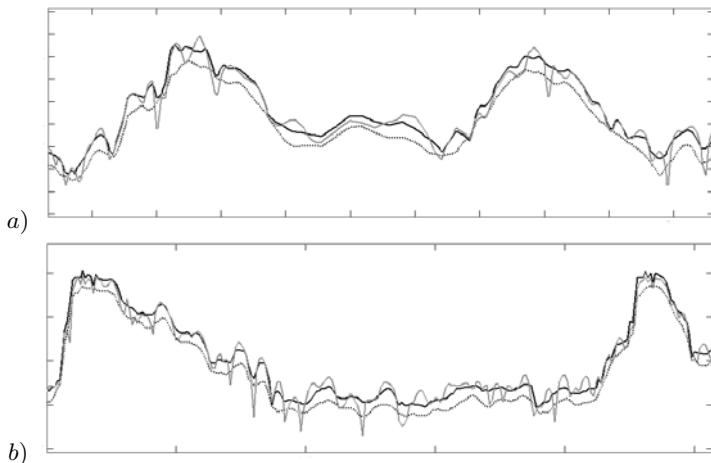


**Fig. 4.** Real images of circular shaped object: a) Original image , b) After Nakagami filtering, c) median filtering and d) Wiener filtering

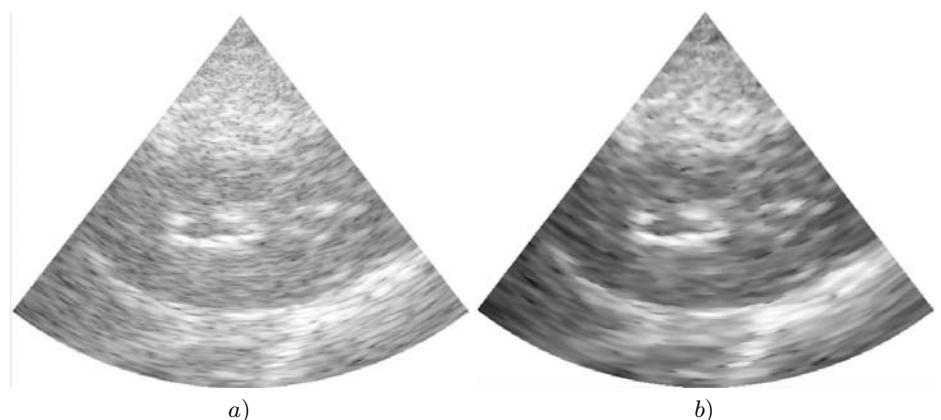
$$SI = \frac{\mu}{\sigma} \quad (14)$$

where  $\mu$  and  $\sigma$  are region mean value and standard deviation. From the view of SI evaluation it seems that the median filter is the best choice for speckle suppression. This is can be true for homogeneous regions, because the speckles are consider as an impulsive noise. On the other hand, real tissues are highly nonhomogeneous and some advanced method should be employed for speckle filtering.

Sample of circular object with strong reflecting borders is shown on Fig.4a) and the image after filtering by Nakagami filter (window size  $31 \times 3$ ) is shown on Fig.4 b). Fig.4 c,d) shows the images after median and Wiener filtering with the



**Fig. 5.** 1D profile of the circular object. Original profile (gray line), Nakagami filtered (black line), median filtered (dotted line)



**Fig. 6.** Real images of the kidney: a) Original image , b) Nakagami filtered image

same size of window. The filtering was made on the envelope of the RF matrix before scan conversion. The blurring artefact is more obvious in images c) and d), particularly in lateral direction. After Nakagami filter the edges remains more sharp and the speckles are reduced. This is shown on Fig.5, where the vertical and horizontal cross-sections through this circular object are depicted. Only Wiener filtered image is shown on Fig.5. The 1D profile for median filter is very similar and therefore is not shown. Real tissue Nakagami filtering was also performed. The kidney and heart was tested and visually evaluated. Samples of kidney filtering is shown on Fig.6. The probe frequency was 3.3MHz, depth of scanning 10cm.

## 6 Conclusion

The new method for speckle reduction in ultrasound images has been presented. It uses the parameter  $m$  from the Nakagami distribution and it therefore depends on the quality of estimators. This problem hasn't been discussed here in details. We've focused only on the design of the adaptive filter. The design is in fact simple and empirical for the intention of speckle elimination. The apparent speckle reduction was achieved both, in simulated and in ultrasound images from phantom and real tissues, while the edges remain sharp.

## Acknowledgment

This work has been supported by the grant of Ministry of Education (Czech republic) CEZ MS 0021 630513 and the grant no. 102/03/P153 of Czech Science Foundation.

## References

1. Bamber J.C., Daft C., Adaptive filtering for reduction of speckle in ultrasonic puls-echo images, Ultrasonics, January, 1986, pp. 41-44.
2. Crimmins,T.R., Geometric filter for speckle reduction, Applied Optics vol.24, no.10, pp.1438-1443, 1985
3. Dutt, V. Statistical Analysis of Ultrasound Echo Envelope, PhD Thesis, The Mayo Graduate Scholl, 1995
4. Insana,M.F. et al. On the Information Content of Diagnostic Ultrasound, Proceedings of the Tenth International Conference on Information Processing in Medical Imaging, 1987, pp.437-455
5. Kolář R., Jiřík R., Jan J. Estimator Comparison of the Nakagami-m Parameter and its Application in Echocardiography, Radioengineering, April, 2004, vol. 13, no. 1, pp.8-12
6. Kolář, R., Kozumplík, J.: Noise Suppresion in Ultrasound Images Using Wavelets. Proceedings of the International Conference Applied Electronics 2001, Sept. 2001, Pilsen, pp. 130-133.

7. Shankar, P.M. Ultrasonic Tissue Characterization Using a Generalized Nakagami Model, *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 2001, vol. 48, no. 6, pp. 1716-1720.
8. Shankar, P.M., A General Statistical Model for Ultrasonic Backscattering from Tissues, *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 2000, vol. 47, no. 3, p. 727-736.
9. User's Guide, Image Processing Toolbox, Mathworks, <http://www.mathworks.com>
10. Zhang, Q.T., A Note on the Estimation of Nakagami-m Fading Parameter, *IEEE Communications Letters*, 2002, vol. 6, no. 6, p. 237-238.

# A Comparison Among Distances Based on Neighborhood Sequences in Regular Grids

Benedek Nagy\*

Department of Computer Science,  
Faculty of Informatics, University of Debrecen,  
PO Box 12. H-4010 Debrecen, Hungary,  
Research Group on Mathematical Linguistics,  
Rovira i Virgili University, Tarragona, Spain  
[nbenedek@inf.unideb.hu](mailto:nbenedek@inf.unideb.hu)

**Abstract.** The theory of neighborhood sequences is applicable in many image-processing algorithms. The theory is well developed for the square grid. Recently there are some results for the hexagonal grid as well. In this paper, we are considering all the three regular grids in the plane. We show that there are some very essential differences occurring. On the triangular plane the distance has metric properties. The distances on the square and the hexagonal case may not meet the triangular inequality. There are non-symmetric distances on the hexagonal case. In addition, contrary to the other two grids, the distance can depend on the order of the initial elements of the neighborhood sequence. Moreover in the hexagonal grid it is possible that circles with different radii are the same (using different neighborhood sequences). On the square grid the circles with the same radius are in a well ordered set, but in the hexagonal case there can be non-comparable circles.

**Keywords:** Digital geometry, Neighborhood sequences, Square grid, Hexagonal grid, Triangular grid, Distance.

## 1 Introduction

The regular grids are very useful in many applications. They used in computer graphics and in image processing as well as in some applications of grid theory. In discrete mathematics they are called regular lattices. From both approaches there are well developed theories about this topic.

The theory of neighborhood sequences comes from the digital geometry. In [13] the two neighborhood relations are introduced on the square grid, and distance defined by each of them or by both of them (alternating used). In [2, 14] the periodic neighborhood sequences were investigated. The aim is to get more flexible distance functions in the digital plane. In [4] the authors introduced the

---

\* This research was partly supported by the grant OTKA F043090.

general, not necessary periodic neighborhood sequences. In [1] the neighborhood sequences in the cubic grid were analyzed.

The neighborhood relations are also well known, for the triangular and the hexagonal grid (they are used, for example in [3]). The neighborhood sequences based on these relations are defined for these grids in [9, 12]. Note, that in this paper we refer the nodes of the grids as points, mostly for graphical reasons. In computer graphics and in description of networks this notion is usually used.

In this paper, after recalling the basic definitions and concepts we present some interesting differences among the distances on different grids. The simplest is the triangular grid, on which only 1 neighborhood relation and therefore only 1 distance can be defined based on the neighborhood relation. The square grid has huge literature, using two kinds of neighborhood relations. There are some recent papers about neighborhood sequences on the hexagonal grid (e.g. [11, 12]). There are three kinds of neighbors, and two kinds of points (they have symmetric roles). Some surprising properties of the hexagonal grid is shown, in which there are significant difference between the hexagonal and the square (or triangular) grid.

## 2 Definitions

In this part we recall some definitions concerning neighborhood sequences. After the general approach the definitions are applied to the regular grids.

We can define the relation  $, k \leq 1, \dots, \infty$ , among the nodes in (planar) graphs. Two nodes are neighbors if they are on the border of the same region. They are  $k$ -neighbors if they are neighbors and the shortest path between them includes at most  $k$  edges.

It is obvious that these  $k$ -neighborhood relations are reflexive and symmetric relations. Moreover they have the following inclusion properties. All  $(k - 1)$ -neighbors of a point are its  $k$ -neighbors as well.

According to the possible types of neighbors of a grid, we can define the so-called neighborhood sequences on this specific grid.

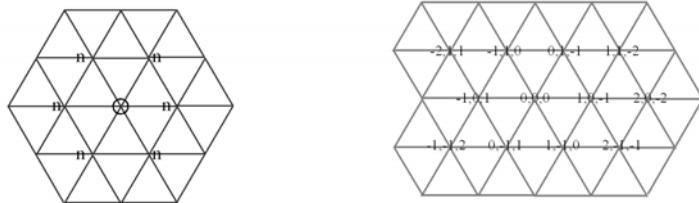
The infinite sequence  $B = (b(i))_{i=1}^{\infty}$ , – in which the values  $b(i) \in \mathbb{N}$  are possible types of neighborhood criteria in the digital space that is used – is called a neighborhood sequence of the used grid.

Let  $p$  and  $q$  be two points (nodes) and  $B = (b(i))_{i=1}^{\infty}$  a neighborhood sequence. A finite point sequence  $\Pi(p, q; B)$  of the form  $p = p_0, p_1, \dots, p_m = q$ , where  $p_{i-1}, p_i$  are  $b(i)$ -neighbor points (we call this step a  $b(i)$ -step) for  $1 \leq i \leq m$ , is called a  $B$ -path from  $p$  to  $q$ . We write  $m = |\Pi(p, q; B)|$  for the length of the path. Denote by  $\Pi^*(p, q; B)$  a shortest path from  $p$  to  $q$ , and set  $d(p, q; B) = |\Pi^*(p, q; B)|$ . We call  $d(p, q; B)$  the  $B$ -distance from  $p$  to  $q$ .

Now, let us see, how these definitions work for the regular grids.

### 2.1 The Triangular Grid

On the triangular grid there is only one neighborhood relation, it can be seen on the left hand side of Fig 1. The used coordinate system is a symmetric with three dependent (namely zero-sum) values (see right hand side of Fig. 1).



**Fig. 1.** The neighbors and the coordinate values of the triangular grid

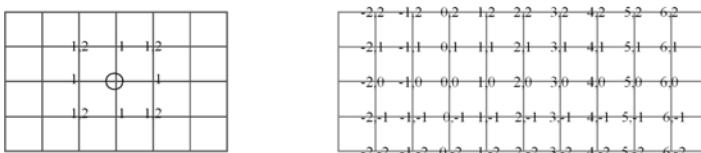
By the help of the coordinate system we have a mathematical definition of the neighborhood: Let  $p(p(1), p(2), p(3))$  and  $q(q(1), q(2), q(3))$  be two points of the triangular grid. The points  $p$  and  $q$  are neighbors if the following condition holds:

1.  $|p(i) - q(i)| \leq 1$  for  $1 \leq i \leq 3$ .

Since there is only one type of neighborhood, there is exactly one distance based on the neighborhood relation. Therefore the theory of neighborhood sequences in the triangular grid is almost trivial.

## 2.2 The Square Grid

The two neighborhood relations are from [13]. A point has four 1-neighbors (excluding itself) and four more 2-neighbors. Fig. 2 shows both kinds of them (left). The used coordinate frame is the Cartesian (Fig. 2, right), which fits very well to this grid.



**Fig. 2.** Types of neighbors on the square grid and the Cartesian coordinate frame

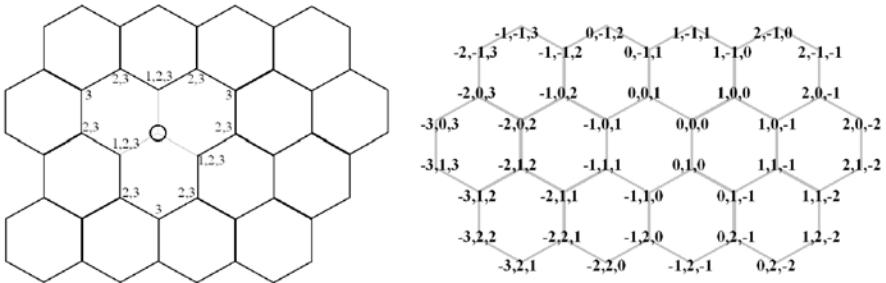
So, mathematically one can redefine the neighborhood criteria in the following way: Let  $p(p(1), p(2))$  and  $q(q(1), q(2))$  be two points in  $\mathbb{Z}^2$ . Let  $k \in \{1, 2\}$ . The points  $p$  and  $q$  are  $k$ -neighbors if the following two conditions hold:

1.  $|p(i) - q(i)| \leq 1$  for  $1 \leq i \leq 2$ ,
2.  $\sum_{i=1}^2 |p(i) - q(i)| \leq k$ .

Because of the two kinds of neighbors, one can define various distances based on the neighborhood sequences (containing the values 1 and 2).

### 2.3 The Hexagonal Grid

In this part we recall some important concepts about the hexagonal grid. Usually, we define three types of neighbors on this grid, as Fig. 3 shows on the left. Each point has three 1-neighbors, nine 2-neighbors (the 1-neighbors, and six more 2-neighbors), and twelve 3-neighbors (nine 2-neighbors, and three more 3-neighbors). We use three coordinate values to represent the points of the grid (based on [11]), see Fig. 3, right-side.



**Fig. 3.** Types of neighbors on the hexagonal grid and the coordinate values

We would like to give some other definitions and notation. We recall some of them from [9, 11, 12]. We have two types of points according to the sum values of the coordinates. If the sum is 0, then we call the point  $\dots$ , if the sum is 1, then the point has  $\dots$ .

We redefine the neighborhood relations mathematically using the presented coordinate-frame. Let  $p(p(1), p(2), p(3))$  and  $q(q(1), q(2), q(3))$  be two points in the hexagonal grid. Let  $k \in \{1, 2, 3\}$ . The points  $p$  and  $q$  are  $k$ -neighbors if the following two conditions hold:

1.  $|p(i) - q(i)| \leq 1$  for  $1 \leq i \leq 3$ ,
2.  $\sum_{i=1}^3 |p(i) - q(i)| \leq k$ .

One can see that if two points are 1-neighbors, then their parities are different. Two non 1-neighbor, but 2-neighbor points have the same parities. Two 3-neighbor points which are not 2-neighbors have different parities.

Since there are three different neighborhood relations on the hexagonal grid, there are more flexibility to define distances by neighborhood sequences.

## 3 Properties of Distances

It is obvious, that for all these distances the value is a natural number (number of steps). Moreover it is 0 if and only if the two points are the same. Now, we

present some properties in which there are differences among the grids and the distances defined on them.

First let us show some properties of the distance on the triangular grid and a basic difference between the triangular grid and the others.

It is easy to show, the proofs can be found in [9, 10] that the distance of two points  $p(p(1), p(2), p(3))$  and  $q(q(1), q(2), q(3))$  of the triangular grid is given by:  $d(p, q; tri) = \max_{i=1,2,3} \{|p(i) - q(i)|\}$ .

Note, that this distance has very nice properties. (It is very easy to compute, moreover:) It is close to the Euclidean distance. If the Euclidean length of a side of the triangle is 1, and the digital distance of two points is  $h$ , then the Euclidean distance of them is between  $\frac{\sqrt{3}}{2}h \approx 0.866h$  and  $h$ .

The digital distance based on the neighborhood criterion of the triangular grid is a metric.

The property above does not hold for all distances based on neighborhood sequences on the square and on the hexagonal grid. To show this fact we present examples for both grids. Let  $B = (2, 1, 1, 1, \dots)$  a neighborhood sequence. (It can be for both of the square and the hexagonal grid, because the 1-neighbors and the 2-neighbors are defined on both of them.) On square grid let  $r(-1, 0)$ ,  $s(0, 1)$  and  $t(1, 2)$  be three points. One can easily check that:  $d(r, s; B) = 1$ ,  $d(s, t; B) = 1$ , but  $d(r, t; B) = 3$ . Therefore  $d(r, s; B) + d(s, t; B) < d(r, t; B)$ .

Now, we are showing an example, when this  $B$ -distance fails on the triangle inequality on the hexagonal grid. Let  $r(0, 0, 0)$ ,  $t(-2, 1, 1)$  and  $s(-1, 0, 1)$  be three points on the hexagonal grid. One can check, that  $d(r, s; B) = 1$ ,  $d(s, t; B) = 1$  and  $d(r, t; B) = 3$ . Therefore we have  $d(r, t; B) + d(t, s; B) < d(r, s; B)$ , again. A necessary and sufficient condition for the neighborhood sequence to generate a metrical distance is proved in [8] for the square grid and in [12] for the hexagonal grid.

Now we are dealing to another metric property, the symmetry.

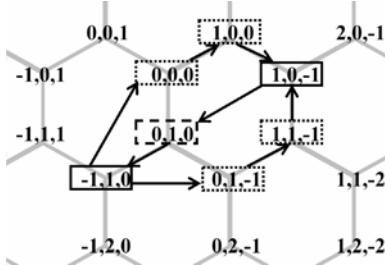
On the square grid, all  $B$ -distances are symmetric. It is a simple consequence of the symmetry of the square-grid: all the points are of the same type, and with a central mirroring of the grid any two points can interchange each-other. In this way any shortest path has a mirror image path, which is also a shortest connecting the points in reverse order.

Now, we are presenting example for the interesting fact, that a  $B$ -distance can be non-symmetric on the hexagonal grid.

Let  $p(1, 0, -1)$  and  $q(-1, 1, 0)$  be two points and  $B = (3, 1, 1, \dots)$  be a neighborhood sequence on the hexagonal grid. Then, by easy calculation one get:  $d(p, q; B) = 2$  (using  $(0, 1, 0)$  at the first step). But,  $d(q, p; B) = 3$ . So  $d(p, q; B) \neq d(q, p; B)$ . See Fig. 4, where the shortest paths are shown for both directions.

This surprising fact occurs because there are two kinds of points, and we cannot use the symmetric pair of a path.

On square grid a  $B$ -distance  $d(p, q; B) > k$  never depends on the order of the first  $k$  element of the used neighborhood sequence  $B$ . It is simple consequence of the fact that one can modify any of the coordinate values in any directions in each step (only the sum of the modified values are limited by the corresponding value



**Fig. 4.** The shortest paths between the points  $(1, 0, -1)$  and  $(-1, 1, 0)$  using neighborhood sequence  $(3, 1, 1, \dots)$

of the neighborhood sequence). So one can use many mirroring transformations of the grid among the points of a path to reorder the elements of the neighborhood sequence getting a new path.

Opposite to this, in hexagonal grid we have:

**Proposition 1.**  $d(p, q; B) > k$

For proving this fact we need to show an example. Let our example be the following: let  $p(1, 0, -1)$ ,  $q(-2, 2, 1)$  and  $B = (3, 1, 3, 1, 1, \dots)$ . Then  $d(p, q; B) = 3$ . Let interchange the first two elements of  $B$  obtaining:  $B' = (1, 3, 3, 1, 1, \dots)$ . Then we have  $d(p, q; B') = 5$ . (Big difference!)  $\square$

The fact above also related to the symmetric properties of the hexagonal grid and the two kinds of points.

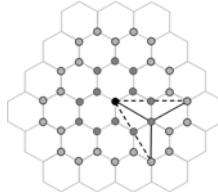
Further in this section, we are detailing several other properties of the neighborhood sequences present on the hexagonal grid, but does not on the square (and on the triangular) grid.

In hexagonal grid, there are different neighborhood sequences which generate the same distance function, formally:

**Proposition 2.**  $d(p, q; B_1) = d(p, q; B_2)$

We show an example. Let  $B_1 = (3, 3, 3, \dots)$  and  $B_2 = (3, 2, 2, \dots)$  with repeating 3's and 2's, respectively. Then one can see in Fig. 5, that (s)he cannot go further by 3-steps than 2-steps after the first 3-step. Fig. 5 shows a point (black) with its 3-neighbors and with their 3-(or 2-)neighbors, since these sets are the same.  $\square$

On square grid for each neighborhood sequence the generated distance function is different. We show, how one can find a point  $p$  for each pair of  $B_1$ - and  $B_2$ -distances (for  $B_1 \neq B_2$ ) such that  $d((0, 0), p; B_1) \neq d((0, 0), p; B_2)$ . Let  $i$  be the smallest index with  $b_1(i) \neq b_2(i)$ . Let the first coordinate of  $p$  be:  $p(1) = i$  and the other one the larger number of the values 2 among the first  $i$  element of



**Fig. 5.** The 3-neighbors (and non 2-neighbors) of a 3-neighbors (and non 2-neighbors) of a (black) point are 2-neighbors of one of its 2-neighbors

the sequences  $B_1$  and  $B_2$ . Then the  $B_1$ -distance of the points cannot be the same as their  $B_2$ -distance, because the number of values 2 among the first  $i$  elements differ for these neighborhood sequences. Using the neighborhood sequence containing more values 2, the distance is exactly  $i$ , while with the other sequence the distance is  $i + 1$ .

In the next proposition, we state that there are 'digital circles' on the hexagonal grid which are the same, but their radii is not equal (using different neighborhood sequences).

**Proposition 3.**  $\{r|d(p, r; B_1) \leq n\} = \{r|d(p, r; B_2) \leq m\}$

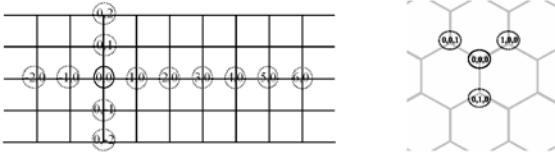
We are presenting an example. Let  $B_1 = (1, 1, 1, \dots)$ ,  $B_2 = (2, 2, \dots)$ ,  $n = 2$  and  $m = 1$ . It is easy to check, that  $\{r|d(p, r; B_1) \leq n\} = \{r|d(p, r; B_2) \leq m\}$  independently of the chosen point  $p$ .  $\square$

It is trivial for the triangular grid that such a digital distance does not exist. On the square grid, it is also trivial. It is a simple consequence of the fact, that for instance the point  $(n, 0)$  has exactly distance  $n$  from the point  $(0, 0)$ , independently of the used neighborhood sequence. (One can step 1 to left at each step, independently of the given value of the sequence.)

On the hexagonal grid, there is no way to go in only a single direction, the three coordinate values depend on each other. Only the distance of the 1-neighbors (including the point itself) of a point is independent of the used neighborhood sequence (in each other cases the neighborhood sequences  $B_1 = (1)_{i=1}^{\infty}$  and  $B_2 = (2)_{i=1}^{\infty}$  define different values of distances). On the square grid, it is a simple fact, that the distances are independent of the neighborhood sequence, when the points has difference at most in 1 coordinate value. On the triangular grid there is no question, because only 1 distance is defined. (See Fig. 6.)

On the square grid the sets of the points which have distance at most  $h$  from a fixed point  $p$  are in a well ordered set. So, for each pair of neighborhood sequence  $B_1$  and  $B_2$  at least one of the following relations hold:

$$\{r|d(p, r; B_1) \leq h\} \subseteq \{r|d(p, r; B_2) \leq h\}$$



**Fig. 6.** The points for which their distance from the Origin are independent of the used neighborhood sequence

or

$$\{r|d(p, r; B_1) \leq h\} \supseteq \{r|d(p, r; B_2) \leq h\}.$$

This ordering is exactly depends on the number 2's among the first  $h$  elements of the neighborhood sequences. When these numbers are the same for two neighborhood sequences, then these sets are equal. If there are more values 2 in the initial part (up to index  $h$ ) of  $B_1$  than  $B_2$  then the second set is a proper subset of the first one.

Now, we are presenting how this fact appears on the hexagonal grid. Let  $r(0, 0, 0)$ ,  $s(-2, 1, 1)$ ,  $p(-1, 2, -1)$  and  $u(-1, 2, 0)$  be four points, and  $B_3 = (3, 1, 2, 2\dots)$ ,  $B_4 = (1, 3, 2, 2\dots)$  be two neighborhood sequences. Then one can compute, that:

$$\begin{aligned} d(r, s; B_3) &= 2 < d(r, s; B_4) = 3, \\ d(r, p; B_3) &= 3 > d(r, p; B_4) = 2, \text{ and} \\ d(r, u; B_3) &= 2 = d(r, u; B_4). \end{aligned}$$

So, one cannot say that the  $B_3$ -distances or the  $B_4$ -distances are greater than the other, it depends on the points as well. Moreover, the set of points which have 2 as  $B_3$ -distances from the point  $(0, 0, 0)$  and the set of points which have 2 as  $B_4$ -distances from the point  $(0, 0, 0)$  are non-comparable sets. So, the 'digital circles' with radius 2 using the origin  $(0, 0, 0)$  are incomparable sets. It is also an important difference between the square and the hexagonal grids. We showed points which are in both of these sets (for instance  $u$ ) and which are in the difference of them. (The point  $s$  is in the first set, but not in the second, while  $p$  is in the second set but it is not in the first one, see Fig. 7.)

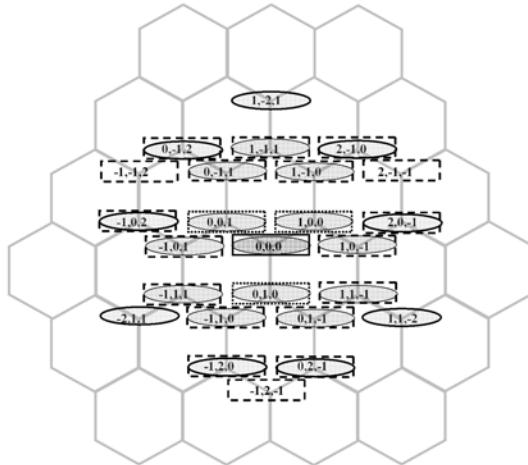
So, on the hexagonal grid there are  $B$ -distances and points such that:

- there are  $B$ -distances, value  $h$  and points, such that from a point to another one the distances are the same (i.e. it is  $h$ ) with the given neighborhood sequences,
- there are points such that for any chosen neighborhood sequence (among the used ones) the  $B$ -distance is  $h$ , but the other distance is greater than  $h$ .

This is also a surprising result, because on the square grid we have not this.

This difference occurs, because the symmetry of these grids, and because of the fact presented in Proposition 1.

The facts described above are interesting properties of the hexagonal grid with comparison to the other regular grids. Now, we are showing a more strict property.



**Fig. 7.** The points of circles with radius 2 from the Origin by neighborhood sequences  $(3, 1, \dots)$  (signed by ellipses) and  $(1, 3, \dots)$  (signed by rectangles)

**Proposition 4.**

$$\text{Let } B_1, B_2 \in \mathbb{N}^{\mathbb{N}}, \text{ then } S_h = \{q \in \mathbb{N}^{\mathbb{N}} \mid d(p, q; B_2) + h > d(p, q; B_1) \text{ for all } p \in S_h\}$$

Before we prove this by an example, we can state it in a more general way:

**Proposition 5.**

$$\text{Let } B_1, B_2 \in \mathbb{N}^{\mathbb{N}}, \text{ then } S_{h,p,r} = \{q \in \mathbb{N}^{\mathbb{N}} \mid d(p, q; B_2) + h > d(r, q; B_1) \text{ for all } q \in S_{h,p,r}\}$$

Let  $B_1 = (1, 1, 1, 1, \dots)$  containing only 1's and  $B_2$  contain  $b_2(i) = 2$  for all  $i$ . One can compute the set  $S_{h,p,r}$  depending only on  $h$  and on the points  $p$  and  $r$ . It is a finite set, because the 'digital circle' using the neighborhood sequence  $B_2$  is blowing faster than the circle using  $B_1$ .  $\square$

It is trivial, that these facts above do not hold on the square grid.

As we have presented, the approximation of the Euclidean distance is good in the triangular grid. In [5] it is showed, that with neighborhood sequences one can obtain better approximation (but, with a lot more computing) on the square grid. The best approximations are given by neighborhood sequences on the hexagonal grid.

## 4 Conclusions

The square grid is well known and used in several applications. One of the advantages is that the Cartesian coordinate frame fits well to it. The triangular

grid is simpler than the square grid, based on the widely used neighborhood criterion, only 1 digital distance can be introduced. When one wants a very fast and short computation with a quite good result, we recommend to use this grid instead of various neighborhood sequences on the square or on the hexagonal grid. The theory of neighborhood sequences is well developed for the square grid, but (with a little bit harder mathematic) on the hexagonal grid there are nicer results, according to the more flexibility (three kinds of neighbors). Some interesting properties of these distances are presented. The distance is an important concept in image processing. The distances presented here can have some non-usual applications as well. Non-metrical distances based on neighborhood sequences are used in practical applications in [6, 7].

## References

1. P.E. Danielsson, 3D Octagonal Metrics, In: *Proceedings of Eighth Scandinavian Conference on Image Processing* 1993, pp. 727-736.
2. P.P. Das, P.P. Chakrabarti, and B.N. Chatterji, Distance functions in digital geometry, *Information Science* vol. 42, 1987, pp. 113-136.
3. E.S. Deutsch, Thinning algorithms on rectangular, hexagonal and triangular arrays, *Communications of the ACM*, vol. 15, no.3, 1972, pp. 827-837.
4. A. Fazekas, A. Hajdu and L. Hajdu, Lattice of generalized neighborhood sequences in  $n$ D and  $\infty$ D, *Publicationes Mathematicae Debrecen* vol. 60, 2002, pp. 405-427.
5. A. Hajdu and B. Nagy, Approximating the Euclidean circle using neighbourhood sequences, In: *Proceedings of the third KEPAF conference* Domaszék, Hungary, Jan 2002, pp. 260-271.
6. A. Hajdu, B. Nagy and Z. Zörgő, Indexing and segmenting colour images using neighborhood sequences, In: *Proceedings of IEEE International Conference on Image Processing*, Barcelona, Spain, Sept 2003, vol 1, pp. 957-960.
7. A. Hajdu, J. Kormos, B. Nagy and Z. Zörgő, Choosing appropriate distance measurement in digital image segmentation, *Ann. Univ. Sci. Budapest. Sect. Comput.* vol. 24, 2004, 193-208.
8. B. Nagy, Distance functions based on neighbourhood sequences, *Publicationes Mathematicae Debrecen* vol. 63, 2003, pp. 483-493.
9. B. Nagy, Shortest Path in Triangular Grids with Neighborhood Sequences, *Journal of Computing and Information Technology* vol. 11, 2003, pp. 111-122.
10. B. Nagy, Metrics based on neighborhood sequences in triangular grids, *Pure Mathematics and Applications* vol. 13, 2002, pp. 259-274.
11. B. Nagy, A symmetric coordinate system for the hexagonal networks, In: *Proceedings of Information Society 2004 – Theoretical Computer Science (ACM Slovenija conference)*, Ljubljana, Slovenia, Oct 2004, vol. D, pp. 181-184.
12. B. Nagy, Non-metrical distances on the hexagonal plane, In: *Proceedings of the 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies*, St. Petersburg, Russian Federation, Oct 2004, pp. 335-338, accepted for publication in *Pattern Recognition and Image Analysis*
13. A. Rosenfeld and J.L. Pfaltz, Distance functions on digital pictures, *Pattern Recognition* vol. 1, 1968, pp. 33-61.
14. M. Yamashita and T. Ibaraki, Distances defined by neighborhood sequences, *Pattern Recognition* vol. 19, 1986, pp. 237-246.

# Restoration of Multitemporal Short-Exposure Astronomical Images

Michal Haindl<sup>1</sup> and Stanislava Šimberová<sup>2</sup>

<sup>1</sup> Institute of Information Theory and Automation,  
Academy of Sciences CR,  
Prague, CZ182 08, Czech Republic

haindl@utia.cas.cz  
<http://www.utia.cas.cz/R0/>

<sup>2</sup> Astronomical Institute  
Academy of Sciences CR,  
Ondřejov, CZ251 65, Czech Republic  
ssimber@sunkl.asu.cas.cz  
<http://www.asu.cas.cz/>

**Abstract.** A multitemporal fast adaptive recursive restoration method based on the underlying spatial probabilistic image model is presented. The method assumes linear degradation model with the unknown possibly non-homogeneous point-spread function and additive noise. Pixels in the vicinity of image steep discontinuities are left unrestored to minimize restoration blurring effect. The method is applied for astronomical sunspot image restoration, where for every ideal undegraded unobservable image several degraded observed images are available.

## 1 Introduction

The major degradation of a ground-based telescope is caused by random fluctuations of the optical way between the object space and the image formation device. There are limitations originating mostly in the Earth's atmosphere and also in the instrumentation itself (telescope & imaging system). Very serious limitation of solar observations is ..., which has its origin in the Earth's atmosphere. The image quality in the image space decreases by variations of the index of refraction, dominated by the thermal turbulence. Influence of the turbulence in image formation were analyzed since the 1950s, see e.g.[1],[2],[3].

Image degradation by seeing is a very complicated process. The three different aspects can be identified:

1. ... - image loses its sharpness; it represents the de focusing effect,
2. ... - if the image remains sharp but it is rapidly shifted back and forth,
3. ... - substantial parts of the image remain sharp but are shifted relative to each other.

Exposure of  $10^{-2}$  s and faster can therefore substantially reduce these aspects of solar seeing. Image degradation is described by the changing complex

point spread function (PSF). PSF of the telescope embodies all the important behaviour of the optical image formation system.

The optical transfer function (OTF)  $H = \mathcal{F}(PSF)$  is generally a complex quantity, and in most cases only its modulus  $MTF(u, v) = \|OTF(u, v)\|$  (modulation transfer function) can be measured. The system transfer function is possible to separate into components originating from diffraction  $OTF_{diff}$ , system aberrations  $OTF_{ab}$ , and from seeing  $OTF_{see}$ , then  $OTF_{total} = OTF_{diff} \times OTF_{ab} \times OTF_{see}$ .

From the restoration point of view we look on the degradation in general, i.e. there is the only one degradation complex unknown function involving all aspects of degradation. The degradation function model has to differentiate between different types of astronomical observations. There are obviously distinct assumptions in modeling of degradation by the long-exposure stellar objects (exposure time about minutes) and the short-exposure solar images (exposure time about milliseconds).

In our case we focus on modeling of degradation of the short-exposure solar images resulting in recovering an undegraded image. In ground-based solar observations it is often possible to obtain several images of the object under investigation that differ just by the component of PSF originating from seeing. For the reconstruction we suppose a short time sequence of images. The exposure time of each image is  $< 15$  ms, so we are allowed to involve them into the "short-exposure" ones and suppose a still scene in the image. Our approach to the image restoration is different to direct and indirect techniques like various types of filtering, power spectral equalization, constrained least-squares restoration, maximum entropy restoration, etc. The image restoration task is to recover an unobservable image given the whole sequence differently corrupted images with respect to some statistical criterion.

Many different image restoration methods have been published and several sophisticated algorithms in the last 10 years have been applied. Most of these methods are general purpose image restoration algorithms which cannot benefit from the specific multitemporal solar observation measurement setup. The simplest restoration method is to smooth the data with an isotropic linear or non-linear shift-invariant low-pass filter. Usual filtering techniques (e.g. median filter, Gaussian low pass filter, band pass filters, etc.) tend to blur the location of boundaries. Several methods [4] try to avoid this problem by using a large number of low-pass filters and combining their outputs. Similarly anisotropic diffusion [5],[6] addresses this problem but it is computationally extremely demanding. Image intensity in this method is allowed to diffuse over time, with the amount of diffusion at a point being inversely proportional to the magnitude of local intensity gradient. A nonlinear filtering method developed by Nitzberg and Shiota [7] uses an offset term to displace kernel centers away from presumed edges and thus to preserve them, however it is not easy to propose all filter parameters to perform satisfactory on variety of different images and the algorithm is very slow. In the exceptional case, when the degradation point-spread function is known, the Wiener filter [8] or deconvolution methods [9] can be used. Model-

based methods use most often Markov random field type of models either in the form of wide sense Markov (regressive models) or strong Markov models. The noncausal regressive model used in [10] has the main problem in time consuming iterative solution based on the conjugate gradient method. Similarly Markov random field based restoration methods [11], [12], [13] require time consuming application of Markov chain Monte Carlo methods. Besides this both approaches have solve the problem when to stop these iterative processes. A similar combination of causal and non-causal regressive models as in this paper was used in [14]. However they assume the homogeneous point-spread function and they identify all parameters simultaneously using extremely time consuming iterations of the EM (expectation maximization) algorithm which is not guaranteed to reach the global optimum. This work generalizes our monospectral restoration method [15],[16] for the multiversion images. It is seldom possible to obtain a degradation model analytically from the physics of the problem. More often a limited prior knowledge supports only some elementary assumptions about this process. Usual assumption, accepted also in this work, is that the corruption process can be modeled using a linear degradation model.

The next section introduces the image degradation model, the core part of the restoration algorithm and contains the model selection criterion. The following sections present results (3) and conclusions (4).

## 2 Image Model

Suppose  $Y$  represents a true but unobservable monospectral image defined on the finite rectangular  $N \times M$  underlying lattice  $I$ . Suppose further that we have a set of  $d$  observable images  $\mathcal{X}$  where each  $X_{\bullet,i} \in \mathcal{X}$  is the  $i$ -th version of  $Y$  distorted by the unknown PSF and noise independent of the signal. The notation  $\bullet$  has the meaning of all possible values of the corresponding multiindex (e.g. the multiindex  $r = \{r_1, r_2\}$  which has the row and columns indices, respectively). We assume knowledge of all pixels elements from the reconstructed scene. For the treatment of the more difficult problem when some data are missing see [17], [18]. The image degradation is supposed to be approximated by the linear discrete spatial domain degradation model

$$X_{r,\bullet} = \sum_{s \in I_r} H_s Y_{r-s} + \epsilon_{r,\bullet} \quad (1)$$

where  $H$  is a discrete representation of the unknown point-spread function,  $X_{r,\bullet}$  is the  $d \times 1$  vector of the  $r$ -th pixel in different distortions and  $Y_{r-s}$  are ideal (unobservable) image pixels. The point-spread function is assumed to be either homogeneous or it can be non-homogeneous but in this case we assume its slow changes relative to the size of an image.  $I_r$  is some contextual support set, and a noise vector  $\epsilon$  is uncorrelated with the true image, i.e.,  $E\{Y \epsilon_{\bullet,i}\} = 0$ . The point-spread function is unknown but such that we can assume the unobservable image  $Y$  to be reasonably well approximated by the expectation of the corrupted image

$$\hat{Y} = E\{X_{\bullet,i}\} \quad (2)$$

in regions with gradual pixel value changes and  $i$ -th degraded image  $X_{\bullet,i} \in \mathcal{X}$  is the least degraded image from the set  $\mathcal{X}$ . The index  $i$  of the least degraded image is excluded from the next equations (3)-(7), (10)-(19) to simplify corresponding notation. The above method (2) changes all pixels in the restored image and thus blurs discontinuities present in the scene although to much less extent than the classical restoration methods due to our restoration model (8) adaptivity. This excessive blurring can be avoided if pixels with steep step discontinuities are left unrestored, i.e.,

$$\hat{Y}_r = \begin{cases} E\{X_r\} & \text{if (4) holds} \\ X_r & \text{otherwise} \end{cases}, \quad (3)$$

where the condition (4) is

$$p(X_r | X^{(r-1)}) > \kappa, \quad (4)$$

and where  $\kappa$  is a probabilistic threshold based on the prediction density. The expectation (2) can be expressed as follows:

$$E\{X\} = \int \begin{pmatrix} x_1 & x_2 & \dots & x_M \\ x_{M+1} & x_{M+2} & \dots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{NM-M+1} & x_{NM-M+2} & \dots & x_{NM} \end{pmatrix}_{r=1}^{NM} \prod_{r=1}^{NM} p(x_r | X^{(r-1)}) dx_1 \dots dx_{NM} \quad (5)$$

where  $X^{(r-1)} = \{X_{r-1}, \dots, X_1\}$  is a set of noisy pixels in some chosen but fixed ordering. For single matrix elements in (5) it holds

$$E\{X_j\} = \int x_j \prod_{r=1}^{NM} p(x_r | x^{(r-1)}) dx_1 \dots dx_{NM} = E_{X^{(j-1)}}\{E_{X_j}\{X_j | X^{(j-1)}\}\} \quad (6)$$

Let us approximate after having observed  $x^{(j-1)}$  the  $\hat{Y}_j = E\{X_j\}$  by the conditional expectation  $E\{X_j | X^{(j-1)} = x^{(j-1)}\}$  where  $x^{(j-1)}$  are known past realization for  $j$ . Thus we suppose that all other possible realization  $x^{(j-1)}$  than the true past pixel values have negligible probabilities. This assumption implies conditional expectations approximately equal to unconditional ones, i.e., then the expectation (6) is  $E\{X_j\} \approx E\{X_j | X^{(j-1)}\}$ , and

$$\hat{Y} = E\{X\} \approx \begin{pmatrix} E\{X_1 | x^{(0)}\} & \dots & E\{X_M | x^{(M-1)}\} \\ E\{X_{M+1} | x^{(M)}\} & \dots & E\{X_{2M} | x^{(2M-1)}\} \\ \vdots & \ddots & \vdots \\ E\{X_{NM-M+1} | x^{(NM-M)}\} & \dots & E\{X_{NM} | x^{(NM-1)}\} \end{pmatrix} \quad (7)$$

Suppose further that a noisy image can be represented by an adaptive 2.5D causal simultaneous autoregressive model

$$X_{r,i} = \sum_{s \in I_r^c} A_s X_{r-s,\bullet} + \epsilon_r , \quad (8)$$

where  $\epsilon_r$  is a white Gaussian noise vector with zero mean, and a constant but unknown covariance matrix  $\Sigma$ . The noise vector is uncorrelated with data from a causal neighbourhood  $I_r^c$ .  $A_s = [a_{s,1}, \dots, a_{s,d}] \forall s$  are parameter vectors. The model adaptivity is introduced using the standard exponential forgetting factor technique in parameter learning part of the algorithm. The model can be written in the matrix form

$$X_{r,i} = \gamma Z_r + \epsilon_r , \quad (9)$$

where

$$\gamma = [A_1, \dots, A_\eta] , \quad (10)$$

$$\eta = \text{card}(I_r^c) \quad (11)$$

is a  $1 \times d\eta$  parameter matrix and  $Z_r$  is a corresponding vector of  $X_{r-s}$ . To evaluate conditional mean values in (7) the one-step-ahead prediction posterior density  $p(X_r | X^{(r-1)})$  is needed. If we assume the normal-gamma parameter prior for parameters in (8) (alternatively we can assume the Jeffrey's parameter prior) this posterior density has the form of Student's probability density

$$p(X_r | X^{(r-1)}) = \frac{\Gamma(\frac{\beta(r)-d\eta+3}{2}) \pi^{-\frac{1}{2}} \lambda_{(r-1)}^{-\frac{1}{2}}}{\Gamma(\frac{\beta(r)-d\eta+2}{2}) (1 + Z_r^T V_{zz(r-1)}^{-1} Z_r)^{\frac{1}{2}}} \left( 1 + \frac{(X_r - \hat{\gamma}_{r-1} Z_r)^T \lambda_{(r-1)}^{-1} (X_r - \hat{\gamma}_{r-1} Z_r)}{1 + Z_r^T V_{zz(r-1)}^{-1} Z_r} \right)^{-\frac{\beta(r)-d\eta+3}{2}} , \quad (12)$$

with  $\beta(r) - d\eta + 2$  degrees of freedom, where the following notation is used:

$$\beta(r) = \beta(0) + r - 1 , \quad (13)$$

$$\hat{\gamma}_{r-1}^T = V_{zz(r-1)}^{-1} V_{zx(r-1)} , \quad (14)$$

$$\begin{aligned} V_{r-1} &= \tilde{V}_{r-1} + I , \\ \tilde{V}_{r-1} &= \begin{pmatrix} \tilde{V}_{xx(r-1)} & \tilde{V}_{zx(r-1)}^T \\ \tilde{V}_{zx(r-1)} & \tilde{V}_{zz(r-1)} \end{pmatrix} , \end{aligned} \quad (15)$$

$$\tilde{V}_{uw(r-1)} = \sum_{k=1}^{r-1} U_k W_k^T , \quad (16)$$

$$\lambda_{(r)} = V_{x(r)} - V_{zx(r)}^T V_{z(r)}^{-1} V_{zx(r)} . \quad (17)$$

where  $\beta(0) > 1$  and  $U, W$  denote either  $X$  or  $Z$  vector, respectively. If  $\beta(r-1) > \eta$  then the conditional mean value is

$$E\{X_r | X^{(r-1)}\} = \hat{\gamma}_{r-1} Z_r \quad (18)$$

and it can be efficiently computed using the following recursion

$$\hat{\gamma}_r^T = \hat{\gamma}_{r-1}^T + \frac{V_{z(r-1)}^{-1} Z_r (X_r - \hat{\gamma}_{r-1} Z_r)^T}{1 + Z_r^T V_{z(r-1)}^{-1} Z_r} .$$

The selection of an appropriate model support ( $I_r^c$ ) is important to obtain good restoration results. If the contextual neighbourhood is too small it can not capture all details of the random field. Inclusion of the unnecessary neighbours on the other hand adds to the computational burden and can potentially degrade the performance of the model as an additional source of noise. The optimal Bayesian decision rule for minimizing the average probability of decision error chooses the maximum posterior probability model, i.e., a model  $M_i$  corresponding to  $\max_j\{p(M_j|X^{(r-1)})\}$ . If we assume uniform prior for all tested support sets (models) the solution can be found analytically. The most probable model given past data is the model  $M_i$  ( $I_{r,i}^c$ ) for which  $i = \arg \max_j\{D_j\}$ .

$$D_j = -\frac{1}{2} \ln |V_{z(r-1)}| - \frac{\alpha(r)}{2} \ln |\lambda_{(r-1)}| + \frac{d\eta}{2} \ln \pi \left[ \ln \Gamma\left(\frac{\alpha(r)}{2}\right) - \ln \Gamma\left(\frac{\beta(0) - d\eta + 2}{2}\right) \right], \quad (19)$$

where  $\alpha(r) = \beta(r) - d\eta + 2$ .

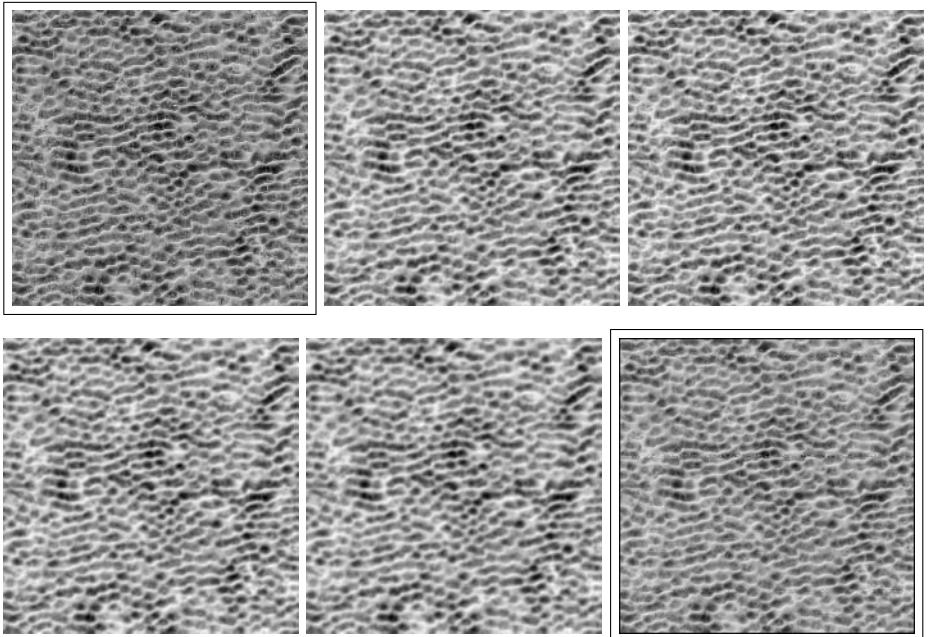
### 3 Results

For the experiments on real data a set of the short-exposure ( $t_{exp} = 10$  ms) solar images has been used. The set consist of four images (Fig.2) observed in the blue part of visible spectra ( $\lambda = 450.7$  nm) with resolution  $0.041''/\text{pixel}$ . Time sequence of the whole set is about 70s. During the standard preprocessing chain all images were corrected by MTF of the telescope, no aberration and seeing corrections have been applied. Because there is no ideal short-exposure solar image available to verify restoration results, we have chosen the most similar skin texture (Fig.1-upper left, both sets of images are presented here in different measure) from our Prague Texture Database (about 1000 prime textures) and artificially tried to simulate solar image degradations (Fig.1). The "ideal" prime skin texture was used to verify the integral restoration criterion (20) as well as the restoration quality itself.

As an objective measure of the restoration performance an integral of a sum of image partial derivatives [19] has been used:

$$D(\hat{Y}) = \int \int \left( \left| \frac{\partial \hat{Y}}{\partial r_1} \right| + \left| \frac{\partial \hat{Y}}{\partial r_2} \right| \right) dr_1 dr_2 \quad (20)$$

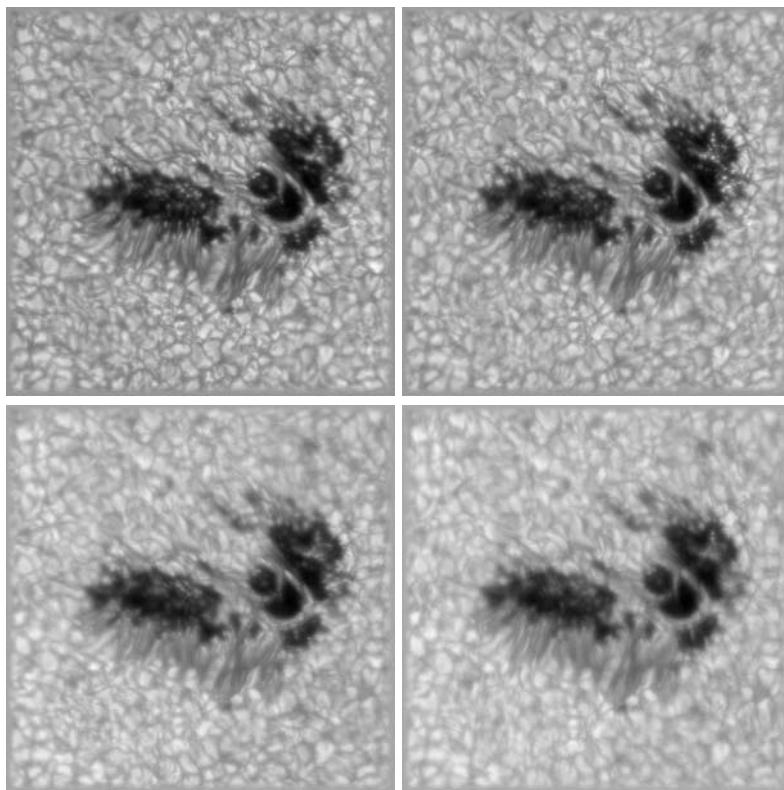
If the unknown point-spread function  $H$  is non-negative, i.e.,  $H_s \geq 0 \ \forall s$  and it preserves the image energy ( $\int \int H_r dr_1 dr_2 = 1$ ) then  $D(Y * H) \leq D(Y)$ . This means that for the less blurred images the  $D$  is growing up. Numerical (absolute mean differences between ideal and degraded images) as well as



**Fig. 1.** Skin texture (framed upper row left) artificially degraded with Gaussian convolution filter ( $5 \times 5, \sigma^2 = 1; 7 \times 7, \sigma^2 = 1$  upper row,  $7 \times 7, \sigma^2 = 2; 11 \times 11, \sigma^2 = 1$  bottom row) and the reconstructed image (framed bottom row right), respectively

the visual evaluation suggest that this criterion can serve as the crude restoration quality estimator. Tab. 1 contains its value for all presented images in the paper.

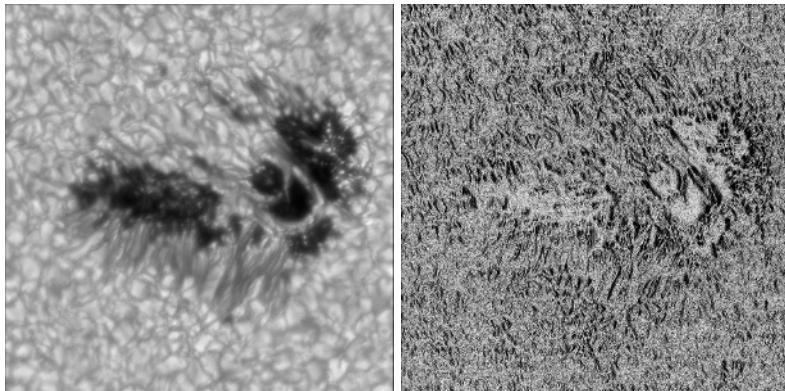
Solar images Fig.2 were reconstructed using the presented algorithm with probabilistic threshold  $\kappa = 0.05$  which left 54 % observation pixels unchanged. The best one from the sunspot set (Fig.2-upper left) served as the reference image for the algorithm. Reconstruction result is in Fig.3 together with the corresponding criterion value in Tab.1. Fig.3-right shows gray level coded predictor probabilities which served to control the restoration switching. The lighter shades represent higher predictor probabilities while dark areas were not changed during the reconstruction. Visual comparison of the reconstructed images with the input degraded images as well as the criterion value demonstrate clear deblurring effect of the presented algorithm and restoration improvement over all used solar measurements. The proposed method was superior over the classical methods (e.g. several low pass filters, pixelwise averaging; blind deconvolution not presented here) using both criteria (20) and the visual one. The Tab. 1 demonstrates the improvement of the deblurring and noise removal effect of the presented algorithm over the real observed image data. The proposed method is clearly superior for the degraded images.



**Fig. 2.** The measured degraded sunspot multitemporal images (courtesy of M. Sobotka)

**Table 1.** The measure of the restoration quality  $D$  evaluated for blurred and restored images

Image	Sunspot and Skin-texture image restoration		$D$
	Restored / Unrestored		
Fig.1 upper left (ideal)		U	88.66
Fig.1 upper middle		U	62.82
Fig.1 upper right		U	70.06
Fig.1 lower left		U	58.56
Fig.1 lower middle		U	55.84
Fig.1 lower right	R		<b>77.17</b>
Fig.2 upper left		U	24.24
Fig.2 upper right		U	23.25
Fig.2 lower left		U	21.39
Fig.2 lower right		U	20.65
Fig.3 left	R		<b>25.61</b>



**Fig. 3.** The reconstructed sunspot image using our method and its corresponding prediction probability image

## 4 Conclusions

The proposed recursive multitemporal blur minimizing reconstruction method is very fast (approximately five times faster than the median filter) robust and its reconstruction results surpasses some standard reconstruction methods, which we were able to implement for verification. Causal models such as [17] have obvious advantage to have the analytical solution for parameter estimation, prediction, or model identification tasks. However, this type of models may introduce some artifacts in restored images. These undesirable effects are diminished by introducing **adaptivity** into the model. This novel formulation allow us to obtain extremely fast adaptive multichannel / multitemporal restoration and it can be easily parallelized. The method can be also easily and naturally generalized for multispectral (e.g. colour, multispectral satellite images) or registered images which is seldom the case for alternative methods. Finally, this method enables to estimate homogeneous or slowly changing non-homogeneous degradation point-spread function (not presented here). Although our preliminary results are very promising, comparison with sophisticated recent alternatives and testing on larger observation sequences is still needed to confirm our conclusions.

## Acknowledgments

This research was supported by the EC projects no. IST-2001-34744 RealReflect, FP6-507752 MUSCLE and partially by the grants no. A2075302 of the Grant Agency of the Academy of Sciences CR, GACR no. 102/04/0155, MŠMT 1M6798555601, and AV CR project 1QS300120506.

## References

1. Fried, D. J. Opt. Soc. Am. **56** (1966)
2. Roddier, F. In Wolf, E., ed.: Progress in Optics. Volume XIX., Nort-Holland (1981)
3. Schulz, T.: Multiframe blind deconvolution of astronomical images. J. Opt. Soc. Am. A **10** (1993) 1064–1073
4. Perona, P.: Deformable kernels for early vision. IEEE Trans. Pattern Anal. Mach. Int. **17** (1995) 488–489
5. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Int. **12** (1990) 629–639
6. Fischl, B., Schwartz, E.: Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. IEEE Trans. Pattern Anal. Mach. Int. **19** (1997) 342–352
7. Nitzberg, M., Shiota, T.: Nonlinear image filtering with edge and corner enhancement. IEEE Trans. Pattern Anal. Mach. Int. **16** (1992) 826–833
8. Andrews, H.C., Hunt, B.: Digital Image Restoration. Prentice-Hall, Englewood Cliffs (1977)
9. Hunt, B.: The application of constraint least square estimation to image restoration by digital computer. IEEE Trans. Computers **22** (1973) 805–812
10. Chellappa, R., Kashyap, R.: Digital image restoration using spatial interaction models. IEEE Trans. Acoustics, Speech and Sig. Proc. **30** (1982) 284–295
11. Geman, S., Geman, D.: Stochastic relaxation , gibbs distributions and bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Int. **6** (1984) 721–741
12. Geman, D.: Random fields and inverse problems in imaging. Springer, Berlin (1990)
13. Jeffs, B., Pun, W.: Simple shape parameter estimation from blurred observations for a generalized gaussian mrf image prior used in map restoration. In: Proc. IEEE CVPR Conf., San Francisco, IEEE (1996) 465–468
14. Lagendijk, R., Biemond, J., Boekee, D.: Identification and restoration of noisy blurred images using the expectation-maximization algorithm. IEEE Trans. on Acoust., Speech, Signal Processing **38** (1990) 1180–1191
15. Haindl, M.: Recursive model-based image restoration. In Sanfeliu, A., Villanueva, J., Vanrell, M., Alquezar, R., Huang, T., Serra, J., eds.: Proceedings of the 15th IAPR Int. Conf. on Pattern Recognition. Volume III., Los Alamitos, IEEE Press (2000) 346–349
16. Haindl, M.: Recursive model-based colour image restoration. In Caelli, T., Amin, A., Duin, R.P.W., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Proceedings, Berlin, Springer (2002) 617–626
17. Haindl, M., Šimberová, S.: A high - resolution radiospectrograph image reconstruction method. Astronomy and Astrophysics, Suppl.Ser. **115** (1996) 189–193
18. Haindl, M., Šimberová, S.: A scratch removal method. Kybernetika **34** (1998) 423–428
19. Subbarao, M., Choi, T., Nikzad, A.: Focusing techniques. J. Optical Eng. **32** (1993) 2824–2836

# A Comparative Study of Angular Extrapolation in Sinogram and Stackgram Domains for Limited Angle Tomography

A.P. Happonen and U. Ruotsalainen

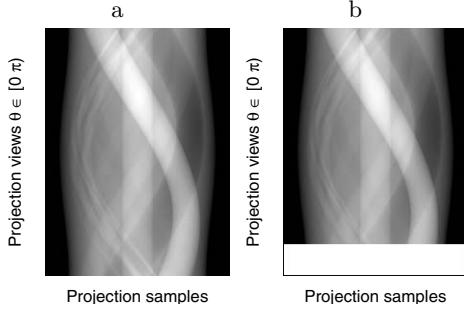
Institute of Signal Processing  
Tampere University of Technology  
P.O.Box 553, FI-33101 Tampere, Finland  
`{antti.happonen, ulla.ruotsalainen}@tut.fi`

**Abstract.** In limited angle tomography, the projection views over a complete angular range of  $180^\circ$  are not available for image reconstruction. The missing part of the projection or sinogram data need to be extrapolated numerically, if standard image reconstruction methods are applied. A novel stackgram domain can be regarded as an intermediate form of the sinogram and image domains, in which the signals along the sinusoidal trajectories of a sinogram can be processed independently. In this paper, we compare extrapolation of incomplete sinogram data in the sinogram and stackgram domains along the angular directions. The extrapolated signals are assumed to be band-limited, other *a priori* assumptions about the data are not made. In this study, we employed simulated numerical data with different ranges of the limited projection views. According to our experiments, extrapolation of the incomplete data in the stackgram domain provides quantitatively better results as compared to extrapolation in the sinogram domain. In addition, tangential degradation in the reconstructed images can not be observed in the case of stackgram extrapolation, in contrast to angular sinogram extrapolation.

## 1 Introduction

In tomography, measurement of a two-dimensional (2-D) cross-section of an object is represented as a sinogram with one-dimensional (1-D) projections. The sinogram is a 2-D matrix representation, where the horizontal row refers to radial samples and the vertical column refers to evenly spaced angular views. A 2-D image of the projected cross-section is recovered from the sinogram data using image reconstruction. A reconstruction method such as filtered back-projection (FBP) [1] can be employed for this ill-posed inverse problem.

In this paper, we consider the problem of limited angle tomography, i.e. sinograms with incomplete ranges of projection views (Fig. 1). In practice, this kind of problem can arise, for example, in positron emission tomography (PET) imaging with BPET [2] or PENN-PET [3] tomograph. Without any restoration technique, the “naive” reconstruction (i.e. the missing projections are replaced with



**Fig. 1.** Two sinograms illustrating limited angle problem in tomography: a) full range of projections; b) limited view of projections

zeros) of incomplete sinogram data does not provide quantitative images and introduces clear artifacts.

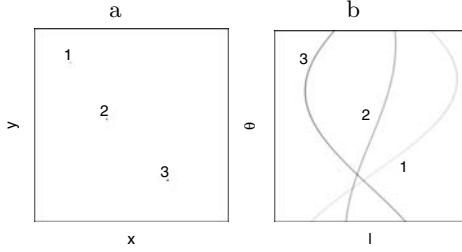
Some algorithms using  $\cdot$ ,  $\cdot$ ,  $\cdot$  information about the sinogram data for extrapolation of the missing data have been introduced (e.g. [4] or [5]). Furthermore, comparisons of different algorithms for limited angle tomography problem have been published [6], [7]. Some of these algorithms do not extrapolate the missing projections of the sinogram directly in the sinogram domain, but incorporate  $\cdot$ ,  $\cdot$ ,  $\cdot$  information e.g. in iterative image reconstruction [8]. In this paper, we compare extrapolation of the missing projections using an extrapolation technique based on the Gerchberg–Papoulis algorithm [9] in the sinogram and stackgram [10] [11] domains along the angular directions. After extrapolation of the missing data, the sinograms can be reconstructed with common reconstruction algorithms.

Extrapolation of discrete signals can be seen as a signal filtering application. In the sinogram domain, filtering along the angular direction introduces tangential or non-uniform blurring to the reconstructed images [12]. In contrast, the angular direction of the stackgrams can be exploited without introducing spatially varying blurring in the reconstructed images, according to our experimental investigation [13]. This suggests that extrapolation in the stackgram domain could be performed in a more appropriate way, and gives a motivation for this comparative study. In our experiments, we assumed that the signals to be extrapolated are band-limited. Additional  $\cdot$ ,  $\cdot$ ,  $\cdot$  knowledge about the data was not exploited, in order to provide a fair comparison between the two domains.

## 2 Background and Problem Formulation

### 2.1 The Stackgram Domain

In the stackgram domain, the signals along the sinusoidal trajectory signals of the sinogram (see Fig. 2) can be processed without interaction with the crossing signals, in contrast to the sinogram domain. The mapping  $S$  from the sinogram



**Fig. 2.** An image and the corresponding sinogram: a) three points in the image domain  $(x, y)$ , and b) their sinusoidal trajectories in the sinogram domain  $(l, \theta)$

$g(l, \theta)$  into the stackgram domain  $(x, y, \theta)$  can be defined in the continuous case as [10]

$$h(x, y, \theta) = Sg(l, \theta) = g(x \cos \theta + y \sin \theta, \theta), \quad (1)$$

where  $\theta \in [0, \pi]$  and  $(x, y) \in \mathbb{R}^2$ . This mapping seems to be similar as the back-projection operator [1]. In the equation (1), however, the back-projection integration from zero to  $\pi$  along the  $\theta$ -direction is replaced by the third  $\theta$ -dimension resulting in the function  $h(x, y, \theta)$ , which forms the stackgram (Fig. 3).

In the stackgram, the values along the sinusoidal trajectory signals of the sinogram, or the locus-signals, are

$$h_{locus}(\theta) = h(x, y, \theta), \text{ for each point } (x, y) \in \mathbb{R}^2, \quad (2)$$

where  $\theta \in [0, \pi]$ .

An inverse of the stack operator  $S_\rho^{-1}$  is a mapping from the stackgram domain  $(x, y, \theta)$  into the sinogram domain  $(l, \theta)$ . This can be written with the weighted Radon transform as

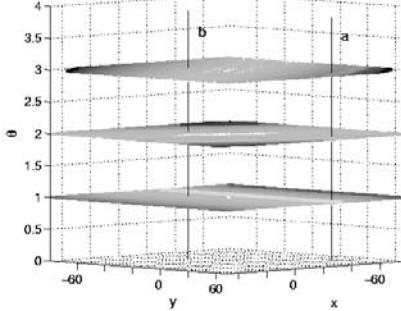
$$g(l, \theta) = S_\rho^{-1}h = \int \int_{-\infty}^{\infty} \rho(x, y, l, \theta)h(x, y, \theta)\delta(x \cos \theta + y \sin \theta - l)dxdy, \quad (3)$$

where  $l \in (-\infty, \infty)$ ,  $\theta \in [0, \pi]$ , and  $\rho$  is a weight function. The operator  $S_\rho^{-1}$ , denoted as the generalized inverse stack operator, maps each back-projected projection at the angle  $\theta$  with the weight function  $\rho$  into a 1-D projection of the sinogram  $g(l, \theta)$ .

Discrete implementations of the both operators (Eq. 1 and 3) are described in the reference [13]. The discrete stack operator is reversible, when it is implemented with the three-pass-rotation algorithm and sinc-interpolation [14].

## 2.2 The Extrapolation Procedure

Consider a discrete sinogram matrix  $\mathbf{g}(l, \theta)$ ,  $l = 0, \dots, M-1$  and  $\theta = 0, \dots, N-1$ , where  $l$  denotes the projection samples and  $\theta$  the number of equally spaced projection views between  $[0 \pi]$  radians. Similarly, let an array  $\mathbf{h}(x, y, \theta)$ ,  $x, y = 0, \dots, M-1$  and  $\theta = 0, \dots, N-1$ , be the corresponding discrete stackgram



**Fig. 3.** An illustration of the stackgram  $(x, y, \theta)$ . The shown layers are back-projected from three projections of the sinogram. In the stackgram, signals parallel to the angular  $\theta$ -axis correspond to the signals along the sinusoidal trajectories of the sinogram (see Fig. 2). The lines **a** and **b** depict the locations of two different trajectory signals in the stackgram

(see Eq. 1). In limited angle tomography, the sinogram  $\mathbf{g}(l, \theta)$ , and therefore the stackgram  $\mathbf{h}(x, y, \theta)$ , are available only for the limited views  $\theta = 0, \dots, L - 1$ , where  $L < N$ . In this study, our objective is to extrapolate the missing projections  $L, \dots, N - 1$  of an incomplete sinogram to a sinogram with a full range of projections  $0, \dots, N - 1$  using the sinogram and stackgram domains.

We employ an extrapolation technique based on the well-known Gerchberg–Papoulis algorithm [9]. The applied extrapolation technique, however, is not recursive, and reduced to single matrix multiplication with the extrapolation matrix [15]. The  $N \times N$  extrapolation matrix  $E$  can be constructed using an ideal low-pass filter matrix  $B_{fc}$  with a cut-off frequency  $f_c$  expressed in number of sample points in the discrete frequency domain. In addition, ones in a switching matrix  $X = \text{diag}\{0, 0, \dots, 0, 1, \dots, 1, 1\}$  represent the missing part of the data  $L, \dots, N - 1$ . The extrapolation matrix  $E$  can be written as [15]

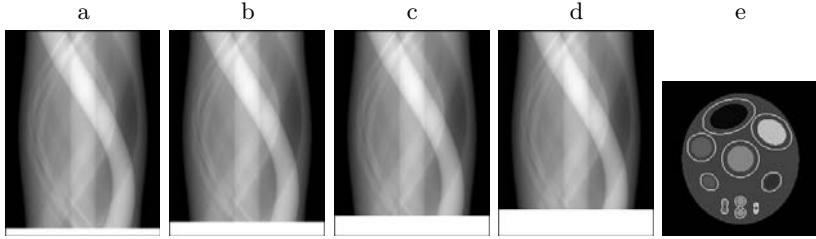
$$E_n = (I - (XB_{fc})^{n+1})(I - XB_{fc})^{-1}, \quad (4)$$

where  $I$  is the identity matrix.

### 3 Methods

We compared extrapolation in the sinogram and stackgram domains along the angular  $\theta$ -directions. The stackgrams  $\mathbf{h}(x, y, \theta)$  were transformed from the sinograms  $\mathbf{g}(l, \theta)$  using the discrete stack-operator [13]. Prior to the inverse transformation, the locus-signals of the stackgrams (Eq. 2) were extrapolated along the  $\theta$ -direction for each  $(x, y)$  position. For comparison, the angular signals of the sinograms were extrapolated along the  $\theta$ -direction for each index  $l$ .

In this study, numerical sinogram data with four different ranges of missing projections (Fig. 4(a-d)) were generated for the comparison. The size of the sinograms were 192 in the radial samples, and 257 in the angular views. The



**Fig. 4.** The employed sinogram data in the study: a) 9 ; b) 17; c) 25; and d) 33 missing projections. In e), the shown ROI:s were used to evaluate with the MSE the performance of the extrapolation methods. For the error evaluation, all of the different ROI:s were regarded together as one ROI

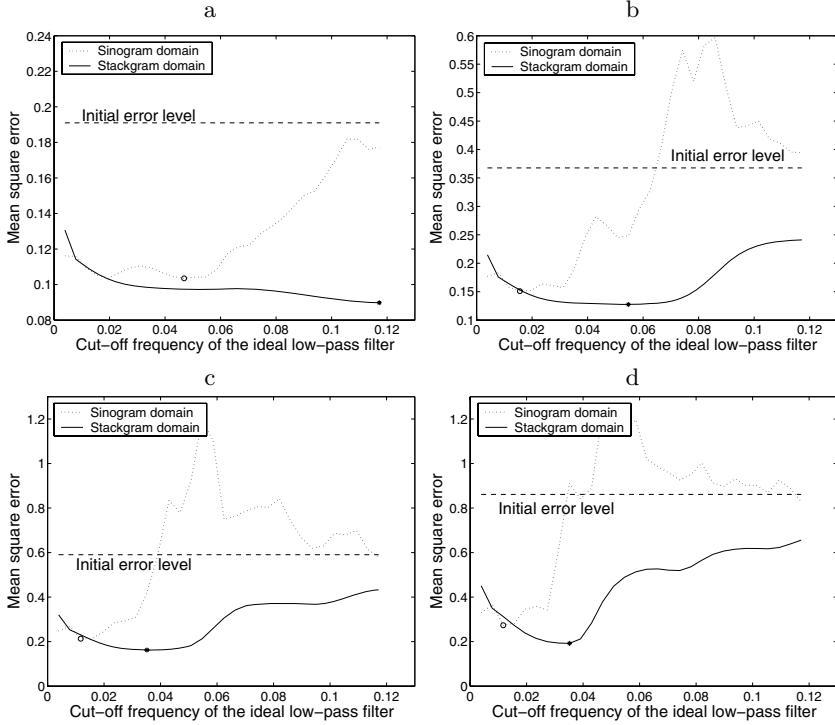
numbers of the missing projections  $L, \dots, N - 1$  were 9, 17, 25, and 33. The corresponding ranges of the projection views were  $174^\circ$ ,  $168^\circ$ ,  $163^\circ$ , and  $157^\circ$ , respectively, when the full range is 180 degrees (as e.g. in PET).

We applied the extrapolation matrices  $E$  with 30 different cut-off frequencies  $f_c$  (Eq. 4). The cut-off frequencies of the ideal low-pass filters were equally spaced from 0.004 to 0.117 (Nyquist frequency 0.5). Extrapolation using these filters results in signals composed of sinusoidal curves of 1-30 different frequencies, since the extrapolation method (Eq. 4) can be seen as a sinusoidal fitting of discrete signals. It can be noticed that the condition of the extrapolation matrix  $E$  is getting worse when the cut-off frequency (or the number of sample points in the discrete frequency domain) increases, because this introduces a more sparse diagonal for the matrix  $E$ . Similar effect happens if the signal to be extrapolated contains too many missing values. In our study, the matrix power  $n$  was 500 (see Eq. 4), which corresponds to the number of iterations in the Gerchberg–Papoulis algorithm [9].

Evaluated images were reconstructed with the FBP-algorithm from the extrapolated sinograms. The reconstructed images were then evaluated with the mean square error (MSE) using a region of interest (ROI) shown in Fig. 4(e). Since extrapolation (as well as filtering) has different meaning in the angular directions of the sinogram and stackgram domains, 30 different extrapolation matrices, as described above, were applied. The cut-off frequencies versus the MSE values, or trade-off curves, are shown for the two extrapolation methods using the four incomplete sinograms. According to the evaluated curves, the resulting best sinograms and FBP-images are shown as well.

## 4 Results

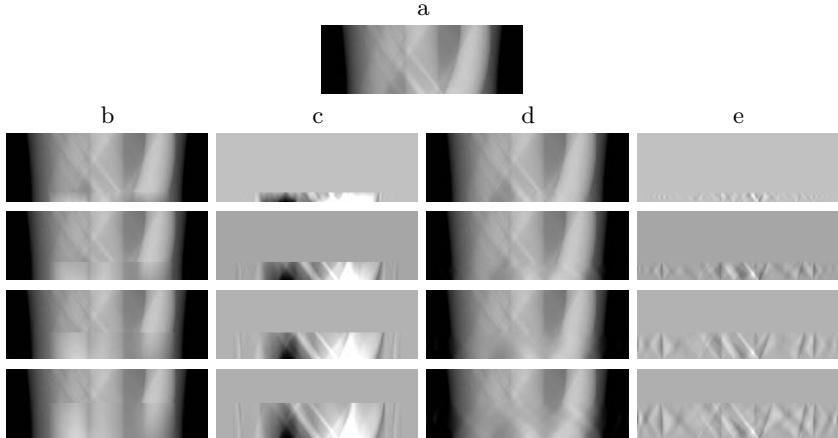
The evaluated trade-off curves with the four sinograms (Fig. 4(a-d)) for the compared methods are shown in Fig. 5. In Fig. 5, the constant curves in dashed line express the initial error levels, which were evaluated in such a manner that the missing projections were simply replaced by the zeros prior to image reconstruc-



**Fig. 5.** The evaluated MSE:s of the reconstructed FBP–images versus the cut-off frequencies (Nyqvist frequency 0.5) of the applied ideal low–pass filters in extrapolation: a) 9; b) 17; c) 25; and d) 33 extrapolated projections. The constant dashed lines show the initial error for the images reconstructed from the incomplete sinogram data (see Fig. 4). The evaluated minimum values are shown with the circles and stars in the sinogram and stackgram cases, respectively. As regards to these curves, extrapolation in the stackgram domain provides better results. The corresponding best sinograms and reconstructed images are shown in Fig. 6 and 7

tion. As it can be noticed, stackgram extrapolation provides the best MSE values and smoother or more predictable trade–off curves for the two methods (Fig. 5), regardless of the range of missing projections. Besides, in the MSE sense, extrapolation in the stackgram domain seems to enable higher cut–off frequencies, and thus, resulting in more complex signals compared to sinogram extrapolation. As regards to the marked minimum values in Fig. 5, the corresponding best sinograms and FBP–images are shown in Fig. 6 and 7.

The sinogram of the full projection range is shown in Fig. 6(a). Fig. 6(b and d) shows the best extrapolated sinograms for the methods, according the minimum values of the curves in Fig. 5. The corresponding error sinogram images are shown in Fig. 6(c and e), respectively. The shown sinograms are congruent with the trade–off curves (Fig. 5). Sinogram extrapolation (Fig. 6(b)) introduces clearly less details (or frequencies) in the extrapolated projections, compared to stackgram extrapolation (Fig. 6(d)).

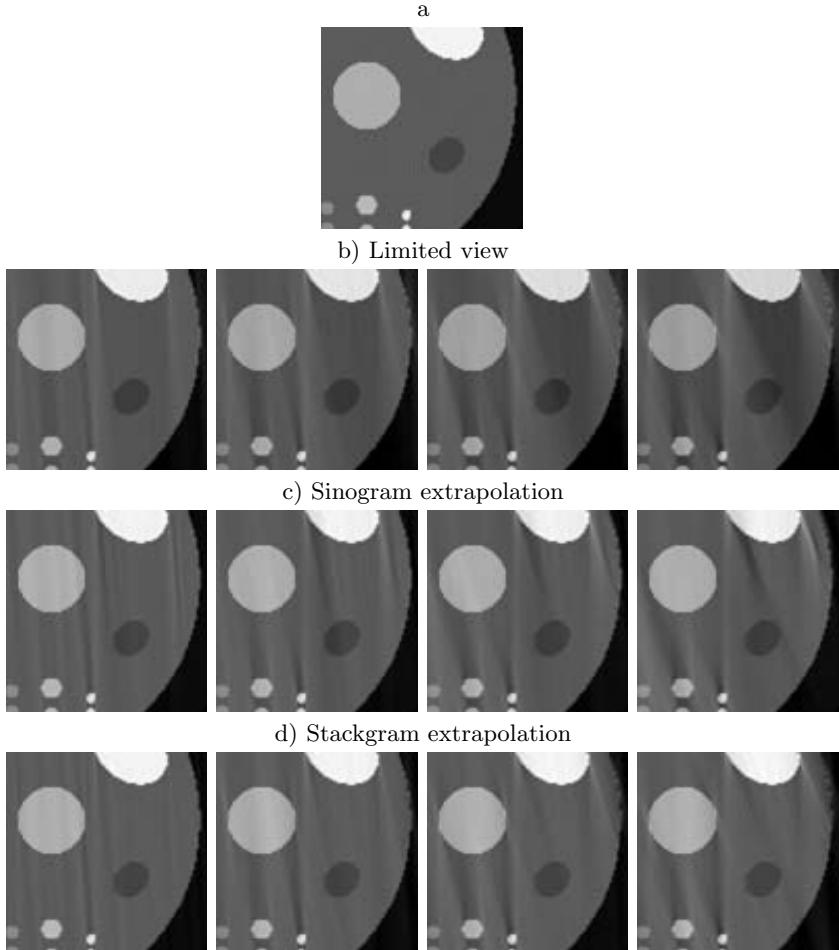


**Fig. 6.** Lower part of the sinograms: a) full range of projections; b) sinogram domain extrapolation; c) error images of b compared to a; d) stackgram domain extrapolation; e) error images of d compared to a. In b to e, the number of extrapolated projections were 9, 17, 25, and 33 from top to bottom, respectively. The figure shows the best sinograms for the both methods, according to the Fig. 5 (see its marked minimum values). The reconstructed images are shown in Fig. 7. In stackgram domain extrapolation (d), the sinusoidal structure of the sinograms is preserved better, compared to sinogram domain extrapolation (b)

The reconstructed images from the sinograms (Fig. 6(a, b, and d)) are shown in Fig. 7. Fig. 7(a) shows the image of the complete sinogram, while Fig. 7(b) shows the images reconstructed with the “naive” FBP-algorithm from the incomplete sinogram data (Fig. 4(a-d)). At first, the resulting FBP-images of sinogram and stackgram extrapolation look similar (Fig. 7(c-d)). However, it can be noticed that the stackgram extrapolation introduces the lowest variations in gray level values, in comparison with both “naive” reconstruction and sinogram extrapolation (Fig. 7(b-c)). Besides, sinogram extrapolation causes tangential blurring in the reconstructed images, as expected, in contrast to stackgram extrapolation. This tangential distortion is more visible when the number of missing projections increases (Fig. 7(c-d)).

## 5 Discussion

In this paper, we compare extrapolation of missing projection data for limited angle tomography in the sinogram and stackgram domains along their angular directions. Stackgram extrapolation performs quantitatively and visually better than sinogram extrapolation, according to the experiments. Extrapolation in the sinogram domain along the angular direction introduces observable tangential blurring in proportion to the incomplete range of missing projections (Fig. 7(c)). Similar blurring effect is well-known in the case of angular sinogram filtering [12], therefore it is commonly avoided in noise reduction. According to



**Fig. 7.** A part of reconstructed FBP–images: a) full range of projections; b) limited projections; c) sinogram domain extrapolation; d) stackgram domain extrapolation. In b, the “naive” reconstruction from the incomplete sinogram data (see Fig. 4(a-d)), in which the number of missing projections were 9, 17, 25, and 33 from left to right, respectively. In c and d, the corresponding FBP–images from the extrapolated sinograms (see Fig. 6(b and d)). The best FBP–images are shown, according to the minimum values of the curves in Fig. 5. In c and d, the angular sinogram extrapolation introduces tangential distortions in the reconstructed images, unlike the angular stackgram extrapolation. It can be noticed that stackgram extrapolation provides quantitatively best results (the least variations in gray level values), as the curves (Fig. 5) also indicate. The images share a common gray scale

our experiments, the stackgram domain offers a more convenient extrapolation environment, since the tangential blurring cannot be observed in the reconstructed images (Fig. 7(d)). Our earlier filtering studies [13] support this finding as well.

The sinogram  $g(l, \theta)$  is symmetric, i.e.  $g(l, \theta) = g(-l, \theta \pm \pi)$ , and periodic in  $\theta$  with period  $2\pi$ , but not with  $\pi$  [1]. Thus, in the discrete case ( $l = 0, \dots, M - 1$  and  $\theta = 0, \dots, N - 1$ ), the values of the sinogram along the  $\theta$ -direction at 0 and  $N - 1$  can differ distinctly (one can verify this from Fig. 2). In the stackgram domain, on the other hand, the values of the locus-signals (Eq. 2) are periodic in  $\theta$  with period  $\pi$ . The discrete Fourier transform, which is applied in the extrapolation algorithm (Eq. 4), considers the signals as periodic. This may explain the reason why the locus-signals, in contrast to the angular sinogram signals, can be extrapolated using the higher cut-off frequencies and resulting in better extrapolated signals (Fig. 5). However, the extrapolated parts of the signals contain low-frequencies or simple shapes, since the applied cut-off frequencies are relatively low (Fig. 5). The higher cut-off frequencies would introduce ill-posed matrix inverses (Eq. 4).

In this study, we did not use , , knowledge about the data to be extrapolated. We employed noiseless numerical data in the experiments. In practice, e.g. in the case of PET data, the sinograms can be corrupted by the noise significantly. Therefore, more sophisticated extrapolation techniques for stackgram extrapolation need to be further studied.

## 6 Conclusion

We compared extrapolation of missing projections for limited angle tomography in the sinogram and stackgram domains along the angular directions. The extrapolated signals were assumed to be band-limited, additional , , knowledge about the data were not exploited. Our experiments show, although further studies are still needed, that stackgram domain extrapolation can provide quantitatively and visually better results than sinogram domain extrapolation. In the stackgram extrapolation, the sinusoidal structure of the extrapolated projections can be restored better and the results are more predictable, compared to the sinogram extrapolation, according to our experiments. In limited angle tomography, the stackgram domain can offer a new potential approach for extrapolation of the missing projections from the incomplete sinogram data.

## Acknowledgments

This work was supported by the Academy of Finland (project no. 104834).

## References

1. Jain, A. In: *Fundamentals of Digital Image Processing*. Prentice-Hall International, Englewood Cliffs, NJ (1989) 434–448
2. Freifelder, R., Cardi, C., Grigoras, I., Saffer, J.R., Karp, J.S.: First results of a dedicated breast PET imager, BPET, using NaI(Tl) curve plate detectors. In: *IEEE Nuclear Science Symposium Conference Record*, 2001, San Diego, CA (2001) 1241–1245

3. Karp, J., Muehllehner, G., Mankoff, D., Ordóñez, C., Ollinger, J., Daube-Witherspoon, M., Haigh, A., Beerbohm, D.: Continuous-slice PENN-PET: a positron tomograph with volume imaging capability. *Journal of Nuclear Medicine* **31** (1990) 617–627
4. Yau, S.F., Wong, S.H.: A linear sinogram extrapolator for limited angle tomography. In: 3rd International Conference on Signal Processing, Beijing, China (1996) 386–389
5. Kazantsev, I.G., Matej, S., Lewitt, R.M.: Limited angle tomography and ridge functions. In: IEEE Nuclear Science Symposium Conference Record. (2002) 1706–1710
6. Ollinger, J.M., Karp, J.S.: An evaluation of three algorithms for reconstructing images from data with missing projections. *IEEE Trans. Nucl. Sci.* **35** (1988) 629–634
7. Oskoui, P., Stark, H.: A comparative study of three reconstruction methods for a limited-view computer tomography problem. *IEEE Trans. Med. Imag.* **8** (1989) 43–49
8. Delaney, A.H., Bresler, Y.: A fast and accurate fourier algorithm for iterative parallel-beam tomography. *IEEE Trans. Image Processing* **5** (1996) 740–753
9. Papoulis, A.: A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Trans. Circuits Syst.* **22** (1975) 735–742
10. Happonen, A.P., Alenius, S.: Sinogram filtering using a stackgram domain. In: Proc. of the Second IASTED International Conference: Visualization, Imaging and Image Processing, Malaga, Spain (2002) 339–343
11. Happonen, A.P., Ruotsalainen, U.: Three-dimensional alignment of scans in a dynamic PET study using sinusoidal trajectory signals of a sinogram. *IEEE Trans. Nucl. Sci.* **51** (2004) 2620–2627
12. Daube-Witherspoon, M.E., Carson, R.E.: Investigation of angular smoothing of PET data. *IEEE Trans. Nucl. Sci.* **44** (1997) 2494–2499
13. Happonen, A.P., Alenius, S.: Investigation of sinogram filtering using stackgram domain. *IEEE Trans. Med. Imag.* (revised and awaiting the final decision) (2005)
14. Unser, M., Thevenaz, P., Yaroslavsky, L.: Convolution-based interpolation for fast, high quality rotation of images. *IEEE Trans. Image Processing* **4** (1995) 1371–1381
15. Sabri, M.S., Steenaart, W.: An approach to band-limited signal extrapolation: The extrapolation matrix. *IEEE Trans. Circuits Syst.* **25** (1978) 74–78

# A Classification of Centres of Maximal Balls in $\mathbb{Z}^3$

Robin Strand

Centre for Image Analysis, Uppsala University,  
Lägerhyddsvägen 3, SE-75237 Uppsala, Sweden  
[robin@cb.uu.se](mailto:robin@cb.uu.se)

**Abstract.** A classification of centres of maximal balls (CMBs) in  $\mathbb{Z}^3$  derived from generalizations of the chessboard and city block metrics to 3D, a weighted metric, and the Euclidean metric is presented. Using these metrics, the set of CMBs (the medial axis) can be extracted. One difficulty with skeletonization in 3D is that of guaranteeing reversibility. A reversible skeleton generally consists of both surfaces and curves. Previous attempts to construct connected skeletons including the CMBs uses conditions based on local neighbourhood configurations. However, a local neighbourhood might be too small and, most important, does not allow a consistent definition for surface- and curve-parts of the skeleton. The classification of the CMBs presented in this paper will be a tool for defining which parts of a 3D skeleton are surfaces and curves.

## 1 Introduction

The medial axis transform – the detection of centres of maximal balls (CMBs) in a 2D binary shape – was proposed by Blum, [1]. A ball included in an object is a maximal ball if it is not completely covered by any other singel ball also included in the object. The CMBs can be used to construct reversible skeletons. Skeletons are widely used in image analysis within many applications. Many different approaches for constructing skeletons have been developed.

The set of CMBs is a thin connected set in  $\mathbb{R}^3$ , but in  $\mathbb{Z}^n$  it is in general neither thin nor connected. To construct a connected digital skeleton representing the original object, these CMBs have to be connected. One approach to construct a connected skeleton was presented in [2]. With this approach, a reversible skeleton, i.e., a thin, centered, and topologically equivalent representation of the object from which the original object can be reconstructed, is generated. The set of CMBs are anchor-points, i.e., voxels that are not allowed to be removed. The rules are designed to connect the CMBs in such a way that the resulting skeleton consists of surfaces and curves. These rules are usually based on local configurations of the object. The problem is that there is no definition of “surface” or “curve” skeletal voxels. Basically, the rules check a local neighbourhood of the voxels and from that voxel configuration decides if the voxel should belong to the skeleton as a part of a surface or a curve, or be assigned to the background.

A local neighbourhood might, however, not be sufficient; CMBs corresponding to “surface”-parts of the object can be located at a larger distance from any other CMB than what can be detected by a local neighbourhood.

A classification of the skeletal points (the CMBs) in  $\mathbb{R}^3$  into sheets, curves, and points has been made recently, [3]. The formal classification is based on the number of tangencies and the order of the tangency for each maximal ball. The classes are denoted  $A_k^n$ , where  $n$  is the number of tangencies and  $k$  is the order of the tangency between the border of the maximal ball and the border of the object. Using this approach, each CMB can be classified as belonging to a sheet, a curve, or a point part of the skeleton.

- . . . consist of skeletal points where the maximal balls have two distinct tangencies, i.e.,  $A_1^2$  points.
- , . . . can be divided into intersection curves of three sheets,  $A_1^3$ , and the boundary of sheets, i.e.,  $A_3$  points (higher order tangency points).
- , . . . are either centers of quad-tangent spheres,  $A_1^4$ , or centers of balls with one regular tangency and one higher order tangency,  $A_1 A_3$ .

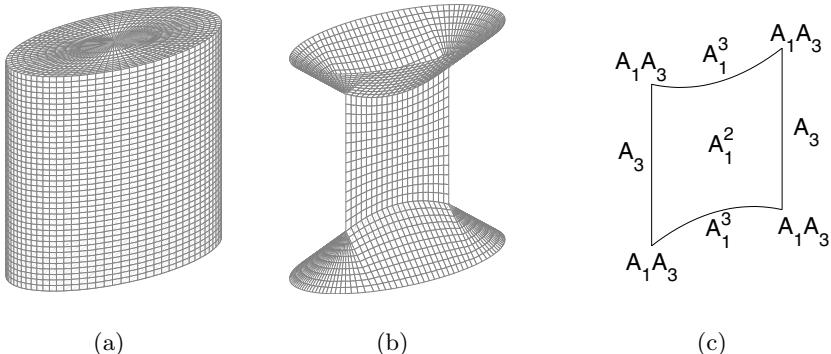
In Figure 1(a), an elliptic cylinder with two planar ends in  $\mathbb{R}^3$  is shown. The medial axis representation (set of CMBs) of the cylinder is shown in Figure 1(b). The CMBs have been divided into sheets, curves, and points according to the classification above, see Figure 1(c) for a classification of the flat sheet (the middle part of the medial axis representation).

In this paper,  $\mathbb{Z}^3$  equipped with four different metrics is considered; the simple metrics  $D^6$  and  $D^{26}$ , a weighted metric based on the weights 3, 4, and 5, and the Euclidean metric. The CMBs are computed and classified based on the number of detected regions where the balls touch the border of the object. Classes similar to the classes previously defined in  $\mathbb{R}^3$  are thus now defined in  $\mathbb{Z}^3$ .

## 2 Basic Concepts and Notations

Two voxels in  $\mathbb{Z}^3$  are . . . if they share a face and . . . if they share at least a vertex. Adjacent voxels will also be denoted . . . . A set of voxels is 6-connected if there is a path consisting of pairwise 6-adjacent voxels in the set between any two voxels in the set. The definition of 26-connected sets is analogous. A binary image  $I$  consists of two sets, the set of object voxels  $X$  and the set of background voxels  $\bar{X}$ . In this paper,  $X$  is assumed to be 26-connected. The . . . of  $X$  consists of the voxels in  $X$  6-adjacent to a voxel in  $\bar{X}$ .

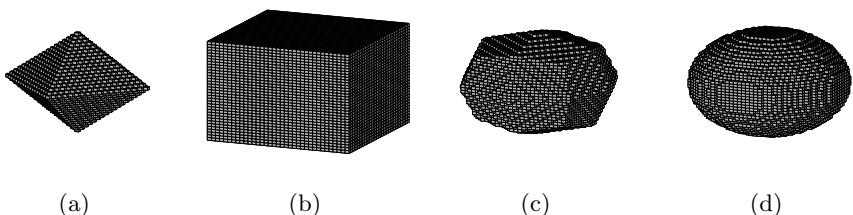
The distance between two voxels using the metrics  $D^6$ ,  $D^{26}$ , or  $<3, 4, 5>$  is defined as the shortest path between the voxels allowing only adjacent voxels, [4]. With unit distance between adjacent voxels, the resulting distance is based on the metrics  $D^6$  and  $D^{26}$ , using 6-adjacency and 26-adjacency, respectively. To get the distance between two voxels using the metric  $<3, 4, 5>$ , the local



**Fig. 1.** An elliptic cylinder  $\mathbb{R}^3$  with two planar ends (a), its medial axis representation (b), and the classification of the flat sheet in the middle of the medial axis representation (c)

distance between face, edge, and vertex neighbours are weighted with 3, 4, and 5, respectively. With this definition, it is obvious that the triangle inequality is fulfilled. It follows that the distance functions are metrics. The digital Euclidean metric is defined using the usual Euclidean distance between the centers of the voxels. A  $\text{..}$  is denoted  $B(x, r) = \{y : d(x, y) < r\}$ , where  $d(x, y)$  is the distance between  $x$  and  $y$  using  $D^6$ ,  $D^{26}$ ,  $< 3, 4, 5 >$ , or the Euclidean metric.

When computing the distance transform (DT), each voxel in  $X$  is assigned a label corresponding to the distance from the voxel to the closest voxel in  $X$ . The DTs based on the metrics  $D^6$ ,  $D^{26}$ , and  $\langle 3, 4, 5 \rangle$  are computed by propagating distance information locally in a two-scan Chamfer algorithm and are denoted  $DT^6$ ,  $DT^{26}$ , and  $WDT$ . To compute the Euclidean DT ( $EDT$ ), four scans are needed, [5]. Using this method, small errors in the  $EDT$  will be produced. For an error-free algorithm, see, e.g., [6]. In Figure 2, balls of radius 18 (18 · 3 for  $WDT$ ) using the different metrics are shown. See [5, 4] for further information about DTs in 3D images.



**Fig. 2.** Balls in  $\mathbb{Z}^3$  using  $D^6$  (a),  $D^{26}$  (b),  $\langle 3, 4, 5 \rangle$  (c), and the Euclidean (d) metrics

### 3 Extracting the CMBs

Using  $DT^6$  and  $DT^{26}$ , the CMBs are easy to identify. In the same way as for the city block and chessboard metrics in  $\mathbb{Z}^2$ , they appear as local maxima in the DTs, [7]. A voxel in  $X$  is a CMB if none of the adjacent voxels (using the same adjacency relation as when computing the DT) has a higher distance label. With a slight modification, this method can be used also for  $WDT$ ; to avoid detection of false CMBs, the distance label 3 has to be changed to 1 when using the  $WDT$ . This is in analogy with the weighted distance transform in  $\mathbb{Z}^2$  with weights 3 and 4, [8]. The local check to be performed is:  $x$  is a CMB if it has a distance label strictly larger than all its neighbours minus the corresponding weight.

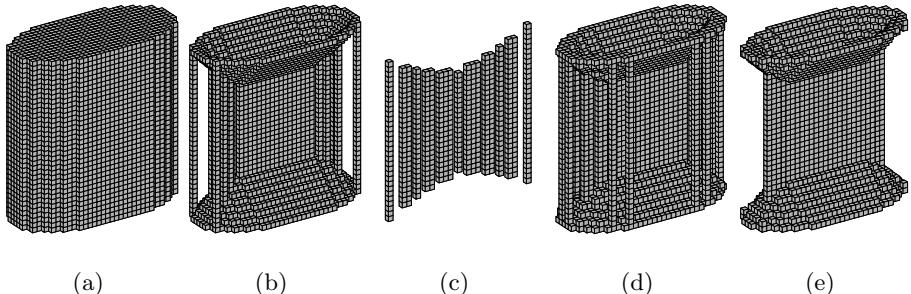
It is not enough to identify local maxima to extract CMBs in an  $EDT$ . To find the CMBs, a look-up table is created. For a center  $x$  of a ball  $B(x, r)$  with a given radius  $r$ , it is computed in advance how big the radii of the balls with centers at positions that are adjacent to  $x$  have to be to cover the ball  $B(x, r)$ . The values are stored in the look-up table. To check if a voxel  $x$  with distance label  $r$  is a CMB, the distance labels of the neighbours to  $x$  are compared with the values in the look-up table in Table 1. Voxel  $x$  is ... a CMB if any neighbour has a distance label strictly greater than the value in the look-up table, otherwise it is a CMB. Observe that some distance labels are not considered in Table 1 since they cannot appear in a  $EDT$  in  $\mathbb{Z}^3$ . This method, however, produces some false CMBs, at least for larger  $r$ , [9]. All the true CMBs are present and therefore the method is accurate enough for the classification in this paper. To only get the true CMBs, larger neighbourhood than  $3 \times 3 \times 3$  must be used.

In Figure 3(a), a digitization of the elliptic cylinder in Figure 1(a) is shown. This object will be used as a running example throughout the paper. The sets of CMBs of the elliptic cylinder using the different metrics are shown in Figure 3(b)-(e).

From the set of CMBs, the original object can be reconstructed by using a distance transform (rDT). A voxel belongs to the original object iff it is at a distance from a CMB less than the distance label of that CMB. For the  $D^6$ ,  $D^{26}$ , and  $< 3, 4, 5 >$  metrics, the rDT can be computed by an algorithm similar to the algorithm used for computing the DT. The CMBs are initially labeled with their distance label and non-CMB voxels are labeled zero. The distance values are propagated by assigning to each voxel the maximum value of its distance label and the distance label of the adjacent voxels minus the

**Table 1.** Look-up table with values of the radii for face-neighbours, flut, edge-neighbours, elut, vertex-neighbours, vlut corresponding to a voxel  $x$  with distance label  $r$ . The first 21 entries in the look-up table are shown. The values in the table should be compared with the squared distance label

$r^2$	1	2	3	4	5	6	8	9	10	11	12	13	14	16	17	18	19	20	21	22	24
flut	2	5	6	7	10	11	12	14	17	18	19	19	21	22	26	27	28	28	30	31	31
elut	3	6	9	10	11	14	15	19	20	21	22	23	26	27	27	30	33	34	35	36	37
vlut	4	7	10	13	13	15	18	20	23	23	25	28	28	30	30	35	35	37	37	39	42



**Fig. 3.** CMBs of an elliptic cylinder (a) using (b)  $D^6$ , (c)  $D^{26}$ , (d)  $<3,4,5>$ , and (e) the Euclidean metrics

corresponding weight. When computing the rDT based on the Euclidean metric, e.g., the method described in [6] can be used.

#### 4 Classification of CMBs in $\mathbb{Z}^3$

It is difficult to make an applicable definition of the order of tangency in  $\mathbb{Z}^3$ , due to its discrete structure. Therefore, this measure is not considered in the classification of CMBs in  $\mathbb{Z}^3$ . The number of tangencies with the object boundary can, however, be generalized to the digital space in a natural way.

In this classification, the notation  $A_m^n$  will be used, where  $m$  and  $n$  denotes the following:

- $m$  The . . . . . of the CMB, i.e., the number of connected regions where the border of the CMB intersects the border of the object.
- $n$  The . . . . . of the CMB, i.e., the number of connected regions where the border of the CMB is in the interior of the object.

In general, this measure can be approximated for a maximal ball  $B(x, r)$  with centre  $x$  and radius  $r$  in an object  $X$  by considering the number of connected components of the intersection of the object and the border of a ball with a slightly larger radius. The number of connected components of  $\partial B(x, r + \epsilon) \cap X$  and  $\partial B(x, r + \epsilon) \cap \bar{X}$  are considered for some sufficiently small tolerance  $\epsilon > 0$ . By keeping the tolerance  $\epsilon$  as small as possible, small variations in the border of  $X$  will be detected. Since many different metrics are considered in this paper, finding the minimal  $\epsilon$  such that  $B(x, r + \epsilon) \setminus B(x, r)$  is a connected set is not trivial. Using  $\epsilon = 1$  will make  $B(x, r + \epsilon) \setminus B(x, r)$  26-connected for  $D^6$  and 6-connected for  $D^{26}$ . For the  $<3,4,5>$  and the Euclidean metrics, general properties of the resulting set are not easily derived. Instead of being enlarged by increasing the radii, the balls will be dilated with the set  $C = \{(x, y, z) \in \mathbb{Z}^3 : \max(|x|, |y|, |z|) \leq 1\}$ . In this way, an approach that is applicable on all metrics is achieved since using a dilation guarantees that  $(B(x, r) \oplus C) \setminus B(x, r)$  is connected.

If the thickness of the object at some point equals an even number of voxels, the set of CMBs might be two voxels thick in places, [2, 10]. This is compensated for by allowing CMBs with the same distance label adjacent (using the same adjacency relation as when computing the DT and 26-adjacency for the EDT) to  $x$  participate in the computation of  $A_m^n$  for  $x$ .

In the present implementation,  $B(x, r)$  is obtained by a rDT. Since the radii are known, these computations can be restricted to a small subset of the image. In pseudocode, the algorithm is

```

for each CMB  $x$  with radius  $r$ , do
    IMAGE =  $B(x, r)$ 
    for each adjacent CMB  $y$  to  $x$  do
        IMAGE = IMAGE  $\cup B(y, r)$ 
    end
    IMAGE = (IMAGE  $\oplus C$ ) \ IMAGE
     $m$  = the number of 26-connected components in  $X \cap$  IMAGE
     $n$  = the number of 26-connected components in  $\bar{X} \cap$  IMAGE
end
```

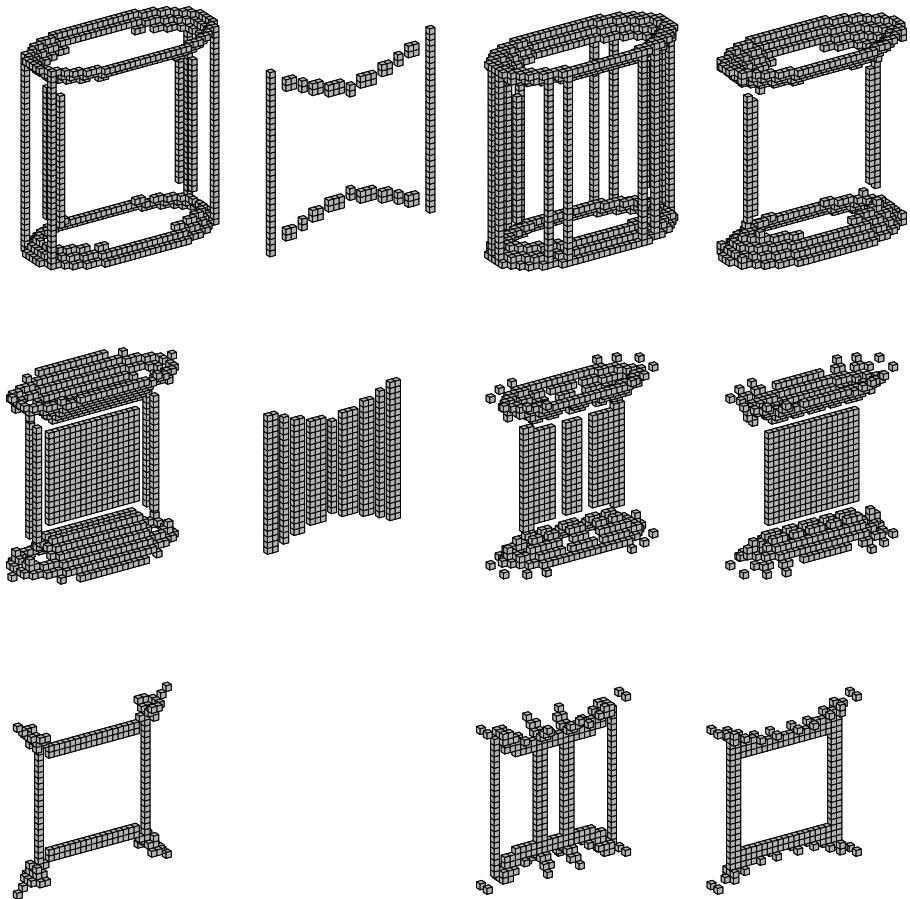
With each CMB labeled with values of  $m$  and  $n$ , the classification of the CMBs into the sets  $A_m^n$  is complete.

## 5 Interpretation of the Classes

A CMB that has two tangency regions and one non-tangency region corresponds to a surface-part of the skeleton. Such a CMB is surrounded by CMBs in directions where the tangencies do not occur. This corresponds to the situation in a local neighbourhood of a point in a surface. A ball with sufficiently small radius and centre in the interior of a surface will have two connected components in the background and one connected component in the surface. At the border of a surface a sufficiently small ball will have exactly one tangency region and one non-tangency region. A ball with centre at the intersection of surfaces has at least three tangency regions and one non-tangency region. With this reasoning, we get the following classification: A CMB classified as

- $A_1^1$  is located at a border of a surface or at an endpoint of a curve of the skeleton.
- $A_2^1$  belongs to a surface part of the skeleton.
- $A_m^1, m > 2$  is located at the intersection curve of  $m$  surfaces.
- $A_1^2$  belongs to a curve part of the skeleton.
- $A_1^n, n > 2$  is located at the point of intersection between  $n$  curves.
- $A_m^n, m, n > 1$  is located at the intersection of both surfaces and curves.

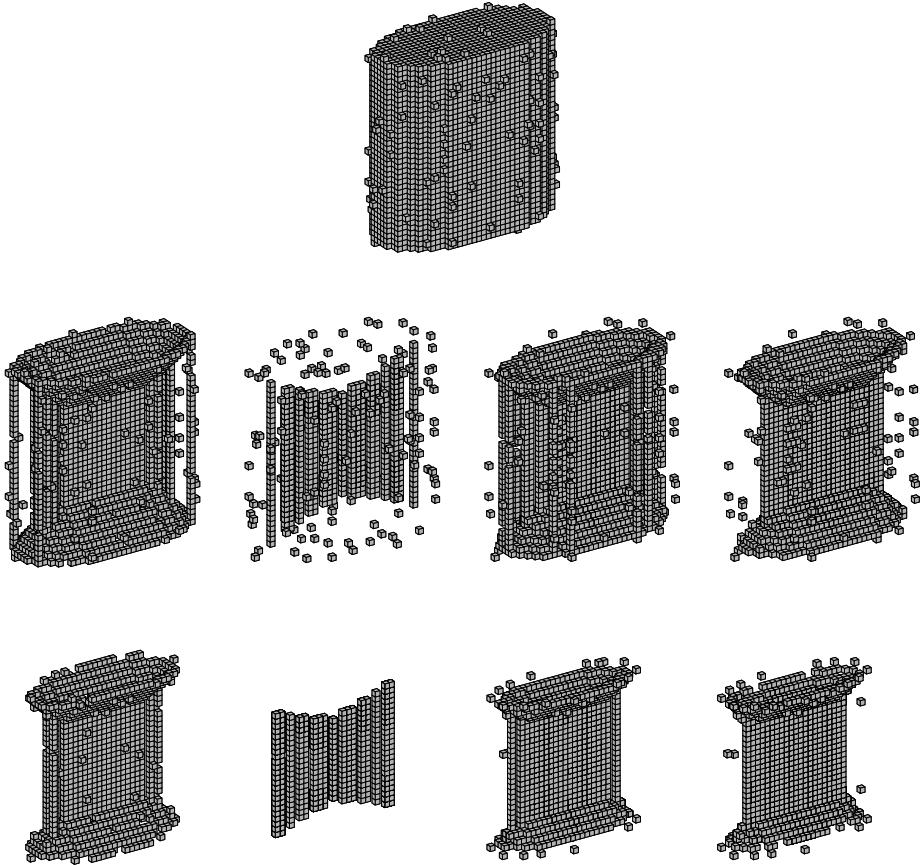
See Figure 4 for a classification of the CMBs in Figure 3.



**Fig. 4.** Classification of the CMBs in Figure 3. Left to right:  $D^6$ ,  $D^{26}$ ,  $<3, 4, 5>$ , and Euclidean metric.

Top row:  $A_1^1$ , second row:  $A_2^1$ , and bottom row:  $A_m^1, m > 2$

It is well-known that the set of CMBs is highly sensitive to noise (e.g., border voxels added to the object). This is not the case for the classification of CMBs into the sets corresponding to surfaces, i.e.,  $A_m^1, m \geq 2$  and  $A_m^n, m, n > 1$ . A noise voxel might increase the number of non-tangency regions by one for some CMB. The CMB will, however, still be assigned to a surface of the skeleton, but with the value of  $n$  increased by one. Often, a border voxel caused by noise will cause a single CMB close to the object border. Such a CMB in general belongs to  $A_1^1$  and does not affect the surface-part of the skeleton. See Figure 5.



**Fig. 5.** Classification of the CMBs of a noisy object. Left to right:  $D^6$ ,  $D^{26}$ ,  $\langle 3, 4, 5 \rangle$ , and Euclidean metric.

Top row: The cylinder in Figure 3(a). Middle row: The set of CMBs. Bottom row: The union of  $A_m^1, m \geq 2$  and  $A_m^n, m, n > 1$

## 6 Conclusions and Future Work

Each CMB is classified as belonging to a surface part, a curve part, the intersection of surfaces and/or curves, or the class  $A_1^1$ . The classification of CMBs is intended for the construction of  $\mathbb{Z}^3$  skeletons. A CMB belonging to  $A_1^1$  should either be assigned to the border of a surface of the skeleton... to the endpoint of a curve. The endpoint of a curve corresponds, in general, to a small detail or a noise voxel in the object resulting in a protrusion of the skeleton. Removing curve end-points corresponds to pruning, an important post-processing step in the skeletonization process, [10]. With the classification proposed in this paper,

CMBs caused by noise or details at the border of the object can be handled consistently and will not cause unwanted surfaces of the skeleton.

Ideally, the skeleton of the running example would consist only of surfaces. By considering the union of  $A_m^1, m \geq 2$  and the CMBs in  $A_1^1$  that is located at the boundary of surfaces consisting of voxels in  $A_m^1, m \geq 2$ , the classes shown in Figure 4 with  $D^6, < 3, 4, 5 >$ , and Euclidean metrics, will constitute sets that are close to the medial axis representation in  $\mathbb{R}^3$ , Figure 1(b). Due to the shape of the balls using  $D^{26}$ , the surface representation using this metric is far from what was obtained in  $\mathbb{R}^3$ .

The main advantage with this classification is that, since the CMBs are divided into surfaces and curves, the rules to construct a connected skeleton from the set of CMBs can be constructed specifically for building surfaces . . . curves. The aim is to use this classification to construct skeletons consisting of surfaces and curves, where the two classes are well-defined.

## Acknowledgement

Many thanks to Prof. Gunilla Borgefors and Dr. Stina Svensson, both Centre for Image Analysis, Uppsala, Sweden, for their valuable comments and suggestions during the development of the classification and preparation of the manuscript.

## References

1. Blum, H.: A transformation for extracting new descriptors of shape. In Wathen-Dunn, W., ed.: Proc. Models for the Perception of Speech and Visual Form, Cambridge, MA, MIT Press (1967) 362–380
2. Sanniti di Baja, G., Svensson, S.: Surface skeletons detected on the  $D^6$  distance transform. In: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition, Springer-Verlag (2000) 387–396
3. Giblin, P., Kimia, B.B.: A formal classification of 3D medial axis points and their local geometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **26** (2004) 238–251
4. Borgefors, G.: On digital distance transforms in three dimensions. *Computer Vision and Image Understanding* **64** (1996) 368–376
5. Ragnemalm, I.: The Euclidean distance transform in arbitrary dimensions. *Pattern Recognition Letters* **14** (1993) 883–888
6. Coeurjolly, D.: d-dimensional reverse Euclidean distance transformation and Euclidean medial axis extraction in optimal time. In: Proceedings of 11<sup>th</sup> Conference on Discrete Geometry for Computer Imagery, Naples, Italy. (2003) 327–337
7. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. *J. ACM* **13** (1966) 471–494
8. Arcelli, C., Sanniti di Baja, G.: Finding local maxima in a pseudo-Euclidean distance transform. *Computer Vision, Graphics, and Image Processing* **43** (1988) 361–367
9. Remy, E., Thiel, E.: Look-up tables for medial axis on squared Euclidean distance transform. In: Proceedings of 11<sup>th</sup> Conference on Discrete Geometry for Computer Imagery, Naples, Italy. (2003) 224–235
10. Svensson, S., Sanniti di Baja, G.: Simplifying curve skeletons in volume images. *Computer Vision and Image Understanding* **90** (2003) 242–257

# 3D Object Volume Measurement Using Freehand Ultrasound

A.L. Bogush and A.V. Tuzikov

United Institute of Informatics Problems,  
National Academy of Sciences of Belarus,  
Surganova 6, 220012 Minsk, Belarus  
[{bogush, tuzikov}@mpen.bas-net.by](mailto:{bogush, tuzikov}@mpen.bas-net.by)  
<http://uiip.bas-net.by>

**Abstract.** Algorithms for volume evaluation of 3D objects on freehand ultrasound images are considered. The position sensor used provides image spatial position and orientation data. The algorithms are based on Watanabe formula for volume computation and use cubic spline interpolation. They allow object volume evaluation on initial image sequence without reconstruction of 3D cube avoiding inevitable data loss at this pre-processing stage. The algorithm accuracy was tested on simulated and real objects.

## 1 Introduction

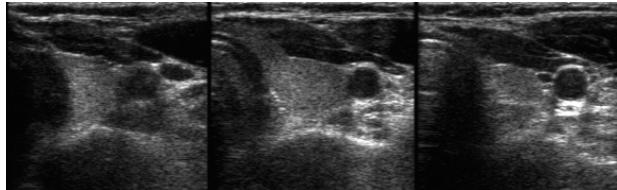
There are known various methods of 3D ultrasound image acquisition. The majority of them compose 3D images as a result of reconstruction from sequential 2D images. Recently special 3D probes have been developed, which are able to produce 3D images of a volume within a body. However, many technical problems need to be solved before such probes will be useful for routine clinical use.

Several different 3D ultrasound imaging approaches have been developed: mechanical scanners, free-hand techniques and 2D arrays [2]. In the last case 2D phased array of transducer elements is used to transmit a broad beam of ultrasound diverging away from the array, sweeping out a volume shaped like a truncated pyramid. However, it is still very expensive to use 2D arrays.

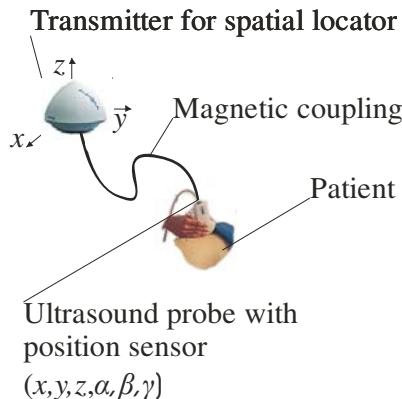
Three basic types of mechanical scanners have been developed: linear, tilt and rotational. They differ only in the manner ultrasound probe is moving, it can be translation, tilting or rotation. But for every type the scanning protocol is predefined. First 3D ultrasound systems based on mechanical movement of the probe were not very suitable for routine clinical applications because of restricted movement and angulations of the transducer.

Fortunately, new 3D ultrasound systems with free-hand acquisition using position sensor systems became available. Free-hand scanning techniques do not require a motorized fixture. In these approaches, a sensor is attached to the transducer to measure its position and orientation. Thus, the transducer can be held by the operator and be manipulated in the usual manner over the anatomy to be imaged. While the transducer is being manipulated, the acquired 2D images (see Fig. 1) are stored by a

computer together with their spatial position and orientation. Several free-hand scanning approaches have been developed, making use of different sensing approaches: articulated arms, acoustic sensing, magnetic field sensing, optical sensing and image-based sensing. Most of recently documented systems use either an electromagnetic or an optical position sensor. In our application an electromagnetic position sensor, mounted on ultrasound probe was used (as it is shown in Fig. 2).



**Fig. 1.** An example of sequence of thyroid gland ultrasound images



**Fig. 2.** Obtaining position and orientation of ultrasound images

The majority of freehand 3D ultrasound systems interpolate these data into a 3D regular array before doing any operations.

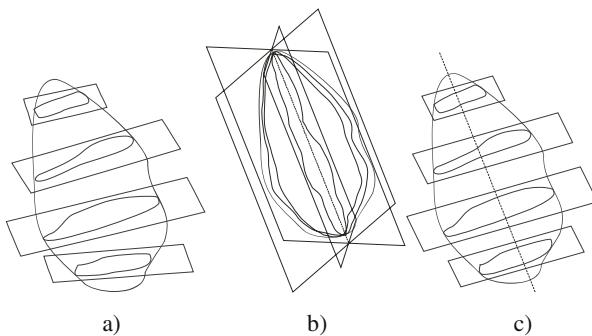
Following the approach proposed in [3] we proposed a volume measurement algorithm from non-parallel cross-sections. This report is devoted to volume estimation algorithm based on Watanabe formula [1] and uses an interpolation by cubic splines. The same splines are applied also for computation of object area and centroid in every cross-section. One can outline object under interest by cubic spline in several non-parallel cross-sections (Fig. 3). We use non-parallel ultrasound images directly without reconstruction to the regular 3D array, and therefore we can overcome data loss at interpolation stage. We implemented several options for object segmentation. The basic option is direct using of non-parallel planes representing initial ultrasound images, as shown in Fig. 4a). Another method is using planes passing through a common pre-defined axis (Fig 4b). This axis is chosen manually. In this case plane images are interpolated from initial cross-section images. This case is

better for visual detecting of object boundaries than the first one. One outlines the object boundaries in these planes and then gets object volume estimation. The third option is derived from the second one. Object outlines for the case with the chosen axis generate outlines in the parallel sections orthogonal to the axis (see Fig 4c).



**Fig. 3.** Thyroid gland outlining in cross-sections

The method proposed is developed for improvement of earlier diagnostics of various diseases, and thyroid gland cancer in particular. This goal can be achieved by decreasing the error level of volume computation from ultrasound data.



**Fig. 4.** Various methods of object outlining: a) non-parallel cross-sections, b) planes with common axis, c) parallel cross-sections

## 2 Method Description

A new method for volume computation from non-parallel cross-sections was suggested and investigated in [3,4]. This method uses a cubic spline interpolation [5]. The method was discussed in details for 2D representation (so called 2D cubic planimetry) and it was mentioned how it can be implemented for the 3D

representation. However, no concrete formulae were given there for the 3D representation. We proposed in [6] the explicit formulae for volume estimation. Suppose that object boundary outlines are obtained in sections  $1, 2, \dots, N$ . Denote by  $V_i$  the object volume between cross-sections  $i$  and  $i+1$ . Assuming that cross-section area is changing according to cubic interpolating spline, for every object part between two cross-sections one has the following formula for volume computation [6]:

$$V_i = \mathbf{x}_i A \mathbf{u}_i^t + \mathbf{y}_i A \mathbf{v}_i^t + \mathbf{z}_i A \mathbf{w}_i^t \quad (1)$$

where

$$A = \frac{1}{240} \begin{pmatrix} 0 & -11 & 12 & -1 \\ 11 & -120 & -143 & 12 \\ -12 & 143 & 120 & -11 \\ 1 & -12 & 11 & 0 \end{pmatrix} \quad (2)$$

Here we use the notation  $\mathbf{x}_i = (x_{i-1}, x_i, x_{i+1}, x_{i+2})$ . Similarly are defined  $\mathbf{y}_i, \mathbf{z}_i, \mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i$ . Note that  $\vec{c}_i = (x_i, y_i, z_i)$  denotes coordinates of the  $i$ -th object cross-section centroid and  $\vec{s}_i = (u_i, v_i, w_i)$  are coordinates of vector, orthogonal to the  $i$ -th cross-section and having the length equal to the object area in the cross-section. It is also assumed that index 0 is replaced by 2 and  $N+1$  is replaced by  $N-1$ .

The volume of the whole object is computed as follows:

$$V = \left| \sum_{i=1}^{N-1} V_i \right| \quad (3)$$

In the case of parallel cross-sections (see Fig 4c)) the volume does not depend on cross-section centroid coordinates and equals to the integral of the cross-section area:

$$V = \left| \int s(h) dh \right|.$$

Here  $s(h)$  is a function of cross-section areas parameterized by a position  $h$  of an intersection point of planes with some orthogonal line.

Denote as before by  $\mathbf{h}_i = (h_{i-1}, h_i, h_{i+1}, h_{i+2})$  and  $\mathbf{s}_i = (s_{i-1}, s_i, s_{i+1}, s_{i+2})$  the values of the parameter and the corresponding cross-section area values used for generating the  $i$ -th spline segment. In this case scalar values of areas are interpolated and the following formula is valid for volume  $V_i$ :

$$V_i = \mathbf{h}_i A \mathbf{s}_i^t, \quad (4)$$

with the same matrix  $A$  as in (2).

Another approach called multiplanar volume approximation is implemented in FreeScan software [7]. Using this method, the longest diameter of a 3D object is determined manually and this diameter is used as a rotation axis. Then nine planes,

rotated around this axis with a fixed angle are automatically generated. The border of the object in each plane is outlined manually and the final volume is evaluated based on these outlines. The evaluation approach used creates equidistant planes orthogonal to the rotation axis. In every equidistant plane a cubic spline is created automatically based on intersection points of outlines with this plane. Then the areas of regions bounded by cubic splines are calculated. The volume results from the areas multiplied by the distance between planes.

We extended this idea in two ways. Firstly, our method using formula (1) can be applied directly to planes, rotated around axis as shown in Fig 4b). One can choose number of planes and their rotation angles around the axis, in order to see better the organ under investigation.

Secondly, parallel cross-sections can be reconstructed automatically from outlines done in planes, rotated around the common axis. The points defining the object boundary are generated for every cross-section. These points are used as spline control points.

We proposed in [8] the explicit formulae for computation of area and low order geometric moments for object bounded by a uniform spline curves. For interpolating spline curve the following formulae hold.

Let us denote

$$\begin{aligned} p_{i1} &= x_i y_{i+1} - x_{i+1} y_i, \quad p_{i2} = x_i y_{i+2} - x_{i+2} y_i, \\ p_{i3} &= x_i y_{i+3} - x_{i+3} y_i, \quad p_{i4} = x_{i+1} y_{i+3} - x_{i+3} y_{i+1}. \end{aligned}$$

where  $(x_i, y_i)$  are a control point coordinates in the local plane coordinate system.

Then the area  $S$  (up to sign) and first order moments  $m_{01}$  and  $m_{10}$  are computed as follows:

$$S = \frac{1}{240} \sum_{i=1}^n (-165 p_{i1} + 24 p_{i2} - p_{i3}) \quad (5)$$

$$\begin{aligned} m_{10} = & \frac{1}{6720} \sum_{i=1}^n (-p_{i1} 1643(x_i + x_{i+1}) \\ & + p_{i2} (302x_{i+1} + 15x_{i+3} + 136(x_i + x_{i+2})) \quad (6) \\ & + p_{i3}(8(x_i + x_{i+2}) - x_i - x_{i+3}) - p_{i4} 15x_i) \end{aligned}$$

$$\begin{aligned} m_{01} = & \frac{1}{6720} \sum_{i=1}^n (-p_{i1} 1643(y_i + y_{i+1}) \\ & + p_{i2} (302y_{i+1} + 15y_{i+3} + 136(y_i + y_{i+2})) \quad (7) \\ & + p_{i3}(8(y_i + y_{i+2}) - y_i - y_{i+3}) - p_{i4} 15y_i) \end{aligned}$$

Centroid coordinates are computed using the first order moments  $m_{01}$  and  $m_{10}$ :

$$(x_c, y_c) = \left( \frac{m_{10}}{S}, \frac{m_{01}}{S} \right) \quad (8)$$

### 3 Volume Computation Algorithm

Below we summarize the algorithm for volume computation from non-parallel cross-sections.

#### Algorithm 1

*Input: A sequence of planes  $\Pi_i$  with normal vectors  $\vec{n}_i, i=1,2,\dots,N$ . Position of every plane is fixed in a global coordinate system. In a local coordinate system of every plane  $\Pi_i$  there are given control points  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n_i}$  which define a spline representing the object boundary. These control points are obtained as a result of a segmentation procedure performed, for example, manually.*

*Output: Volume V of the object.*

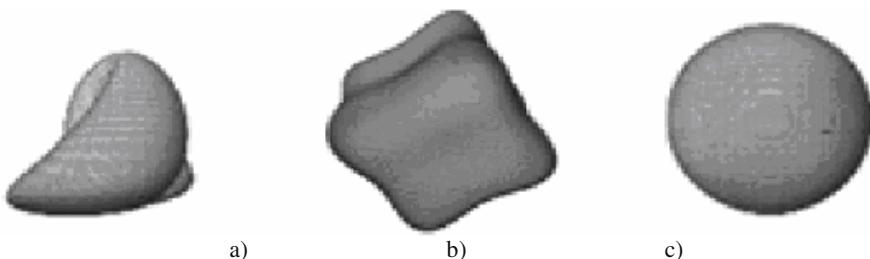
1. For every plane  $\Pi_i$  in a local coordinate system compute object area  $S_i$  and centroid coordinates  $(x_{c_i}, y_{c_i})$  using formulae (5) and (8).
2. Compute global spatial coordinates of centroids  $\vec{c}_i, i=1,2,\dots,N$ , using the plane coordinates and orientation in the global coordinate system.
3. Given a set of vectors  $\vec{s}_i = S_i \vec{n}_i$  and centroids  $\vec{c}_i, i=1,2,\dots,N$  compute the volume using formulae (1) and (3).

It is clear that Algorithm 1 can be applied directly for the case when 2D images have a common axis. One has to take into account that outlined cross-sections are divided into two parts separated by the axis and sorted according to changing of the rotational angle.

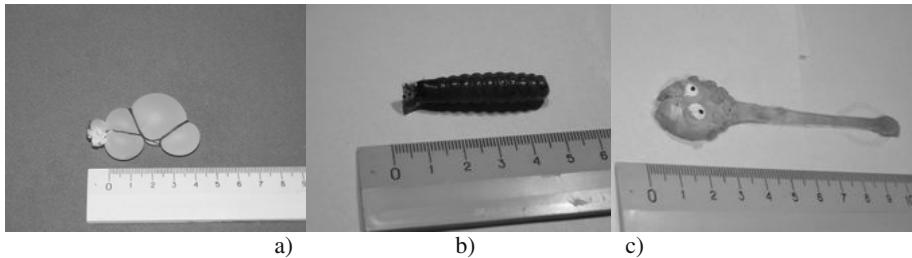
### 4 Experiments Results

To test the accuracy of Algorithm 1 using formulae (1) and (3) we used simulated 3D objects, physical rubber and silicon phantoms.

The tested simulated 3D images (see Fig. 5) have size  $200 \times 200 \times 200$ . The exact volumes of these objects were computed in advance. Non-parallel cross-sections were generated randomly for any given number of sections. We have chosen randomly the.



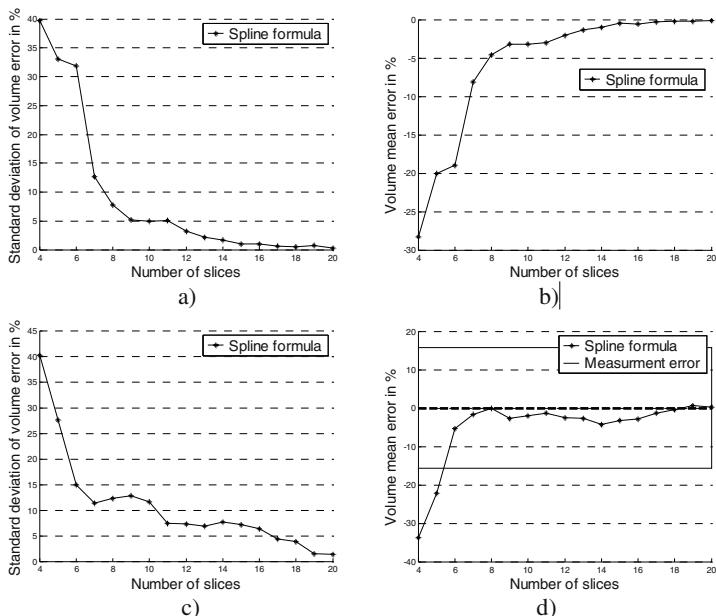
**Fig. 5.** Simulated 3D objects used for volume evaluation



**Fig. 6.** Examples of physical phantoms used for volume evaluation

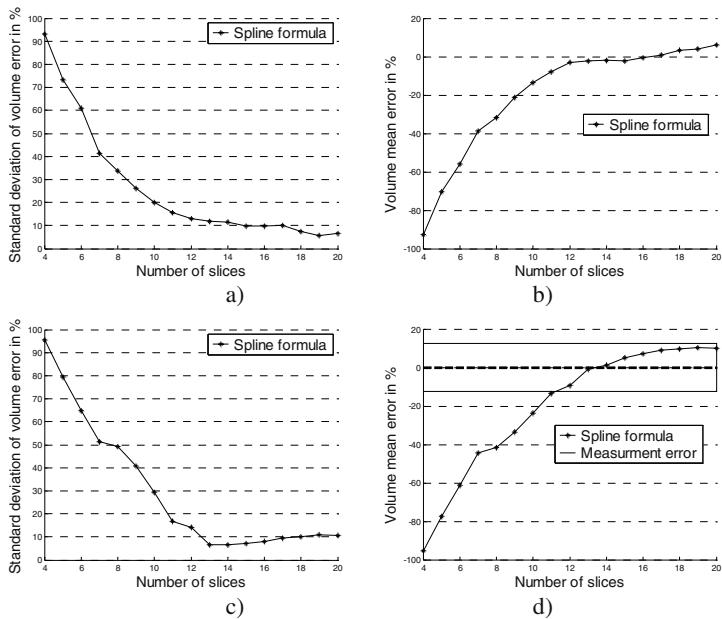
distance between cross-sections and rotation angle around initial positions (up to 15°). The object volume was computed using Algorithm 1 for every realization. Then volume mean and standard deviation errors were calculated for these realizations.

At the next step, we used rubber balls and silicon phantoms, as shown in Fig. 6. They were scanned several times with ultrasound scanner and position sensor mounted on it. The original object volume was measured using measuring glasses with accuracy of 0.25-2.5 ml depending on object sizes.

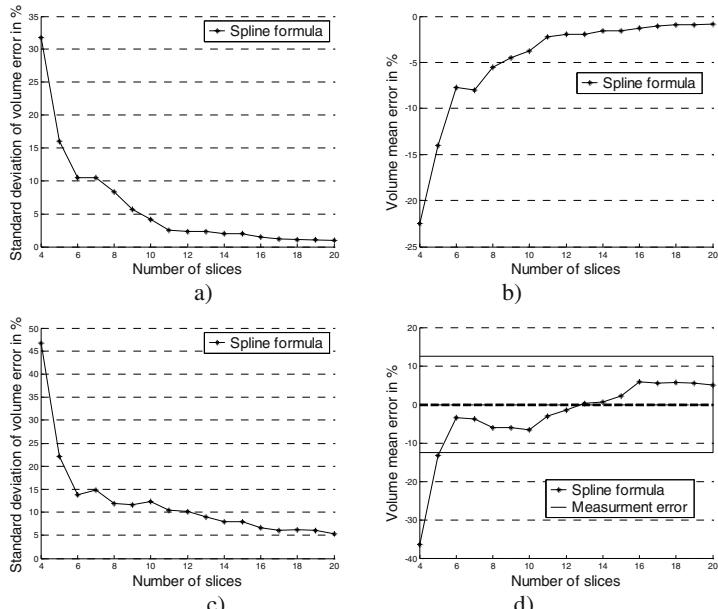


**Fig. 7.** Standard deviation and volume mean errors vs number of cross-sections computed for simulated object shown in Fig. 5a (a,b)) and fantom shown in Fig 6a (c,d))

The plots of the data obtained for simulated 3D object shown in Fig 5a) are presented in Fig. 7-9a) and b). The example of resulting plots for object shown in



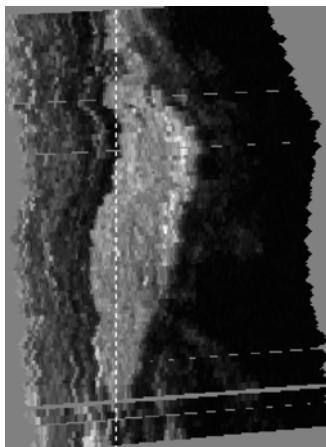
**Fig. 8.** Standard deviation and volume mean errors for objects shown in Fig. 5a) (a, b)) and Fig 6a) (c, d)) for cross-sections with common axis



**Fig. 9.** Standard deviation and volume mean errors for objects shown in Fig. 5a) (a, b)) and Fig 6a) (c, d)) for parallel cross-sections

Fig 6a) are shown in Fig 7-9c) and d). The measurement accuracy limits are shown by solid lines.

When using the current approach volume is under-estimated for low cross-section count. As one may see in Fig. 7, error level of 5% is reached at 8-10 outlined cross-sections. It follows from Fig. 8, that volume measurement results with a common cross-sections are more sensitive to the number of cross-sections. This property should be explained and verified more carefully. One has to take also into account both measurement data acquisition errors. However, using this method one can detect more valuable tissue properties and outline object boundaries more accurately (compare for example a radial cross-section in Fig. 10 and the corresponding initial images of the same object in Fig. 1). Results for parallel cross-sections (Fig. 9) are similar to original, but slightly noisier due to conversion error.



**Fig. 10.** Radial cross-section

## 5 Conclusion

We proposed several algorithms for the 3D object volume estimation from a series of non-parallel 3D freehand ultrasound images.

The algorithms were tested for a number of simulated and real objects with known volumes. The dependence of volume errors on the number of cross-sections was studied. For these quite simple objects it was usually sufficient for 8 cross-sections to obtain a mean volume error less than 5%.

We plan to use our algorithm as a tool in early diagnostics of thyroid cancer with 3D ultrasound because it can allow more accurate determination of progressive growth of cancer nodules as compared with benign ones.

The work was done in framework of the ISTC B-517 project.

## References

1. Y. Watanabe. A method for volume estimation by using vector areas and centroids of serial cross sections. *IEEE Trans. Biomed. Eng.*, 29:202–205, 1982.
2. A. Fenster, D. Downey, and H. Cardinal. Three-dimensional ultrasound imaging. *Physics in Medicine and Biology*, 46:67–99, 2001.
3. G.M. Treece, R.W. Prager, A.H. Gee, and L. Berman. Fast surface and volume estimation from non-parallel cross-sections for freehand 3-D ultrasound. *Medical Image Analysis*, 3(2):141–173, 1999.
4. G. Treece. *Volume measurement and surface visualisation in sequential freehand 3D ultrasound*. PhD thesis, University of Cambridge, Department of Engineering, 2000.
5. E. Catmull and R. Rom. A class of local interpolating splines. In R. Barnhill and R. Risenfeld, editors, *Computer Aided Geometric Design*, pages 317–326. Academic Press, San Francisco, 1974.
6. S.A. Sheynin, A.V. Tuzikov, A.L. Bogush. Improvements of Volume Computation from Non-parallel Cross-sections. *17th International Conference on Pattern Recognition ICPR'2004*, 23-26 August 2004, Cambridge, UK, vol. 4, 815-818.
7. S. Schlogl, E. Werner, M. Lassmann, J. Terekhova, S. Muffert, S. Seybold, and C. Reiners. The use of three-dimensional ultrasound for thyroid volumetry. *Thyroid*, 11(6):569-574, 2001.
8. S. Sheynin and A. Tuzikov. Moment computation for objects with spline curve boundary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25 no. 10:1317-1322, 2003.

# **Modeling, Evaluation and Control of a Road Image Processing Chain**

Yves Lucas<sup>1</sup>, Antonio Domingues<sup>2</sup>, Driss Driouchi<sup>2</sup>, and Pierre Marché<sup>3</sup>

<sup>1</sup> Orleans University, Vision and Robotics Lab,

IUT Mesures Physiques 63 av. de Lattre 18020 Bourges cedex, France

yves.lucas@bourges.univ-orleans.fr

<http://www.bourges.univ-orleans.fr/rech/lvr>

<sup>2</sup> ENSI of Bourges, Vision and Robotics Lab

10 Bd. Lahitolle, 18000 Bourges, France

{Antonio.Domingues, Pierre.Marche}@ensi-bourges.fr

<sup>3</sup> Pierre & Marie Curie University, Theoretical and Applied Statistics Lab

175 rue du Chevaleret 75013 Paris, France

driouchi@ccr.jussieu.fr

<http://www.ccr.jussieu.fr/lsta>

**Abstract.** Tuning a complete image processing chain (IPC) remains a tricky step. Until now researchers focused on the evaluation of single algorithms, based on a small number of test images and ad hoc tuning independent of input data. In this paper we explain how, by combining statistical modeling with design of experiments, numerical optimization and neural learning, it is possible to elaborate a powerful and adaptive IPC. To succeed, it is necessary to build a large image database, to describe input images and finally to evaluate the IPC output. By testing this approach on an IPC dedicated to road obstacle detection, we demonstrate that this experimental methodology and software architecture ensure a steady efficiency. The reason is simple: the IPC is globally optimized, from a large number of real images (180 out of a sequence of 30 000) and with adaptive processing of input data

## **1 Adaptive Processing in Vision Systems**

Designing an image processing application involves a sequence of low and medium level operators (filtering, edge detection and linking, corner detection, region growing ...) in order to extract relevant data for decision purpose ( pattern recognition, classification, inspection ...). At each step of the processing, tuning parameters have a significant influence on algorithm behavior and the ultimate quality of results. As the processing power of micro computers has reached a very high level, artificial vision systems are now developed for demanding applications such as video surveillance or car driving where scene contents is uncontrolled, versatile and rapidly changing. The automatic tuning of the IPC has to be solved there, as the quality of low level vision processes should be continuously preserved to guarantee high level task robustness. The first problem to solve in order to design adaptive vision systems is the evaluation of image processing tasks. Since a few years, researchers are interested in it and proposed rather empirical solutions [1,2,3,4,5,6,7]. When a ground truth is available, it is possible to compare directly this reference to the results using a specific metric. Sometimes

no ground truth exists or remains uncertain and application experts are needed for qualitative visual assessment or empirical numerical criteria are searched for. All these methods consider only one operator at the same time [8,9,10,11]. However the separate tuning of each operator can not often lead to an optimal setting of the complete IPC. Moreover, image operators are generally tested on a too small number of test images, even on artificial noised images, to test algorithms efficiency. This can not replace a large real image base, for IPC testing. But how to evaluate on a great number of images a sequence of image processing operators involving numerous parameters ?

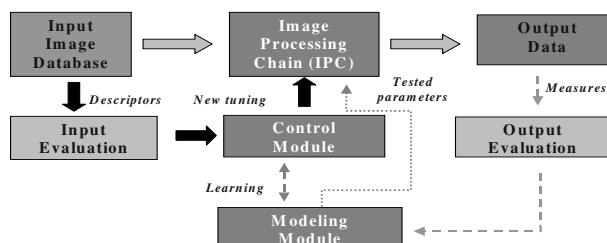
A second problem remains unsolved: how to find the good tuning and so, how to adapt image processing to keep up constant quality of results ? Real time processing being executed by electronic circuits, this hardware must incorporate programmable facilities so that operator parameters can be modified at any time. Artificial retinas and also intelligent video cameras already enable the tuning of some acquisition parameters. Concerning the processing parameters, the amount of computing which is necessary to discern the effect on the results of the modification of several parameters seems at the first glance dissuasive, all the more because each separate image requires different parameters. We note that the choice of operators here still appeals to the experimenter but other works examine also the possibility of its automation [12,13].

We show how to overcome these problems with an experimental approach combining statistical modeling, numerical optimization and learning. We illustrate it in the case of an IPC dedicated to line extraction for road obstacle detection.

## 2 Architecture

### 2.1 General Overview

The software is composed of six modules (Fig.1): IPC, image database, input and output evaluation, modeling and control; The IPC includes a series of low and medium level operators with configurable parameters; The database of input images is specific for an application and covering all situations; The module for IPC output evaluation measures the quality of resulting image entities. The evaluation can be supported by a ground truth or be unsupervised if empirical criteria are used instead; The module for IPC input evaluation enables the control module to tune the processing parameters for each input image. The evaluation should extract from the input images relevant



**Fig. 1.** Architecture of an adaptive IPC

descriptors for IPC tuning; The modeling module; it is required firstly to model the influence of processing parameters on IPC output data and secondly to find an optimal tuning of these parameters for a given input image; Finally, a control module, based on input image description enables to provide for each input image suitable processing parameters after a learning step.

## 2.2 Image Database and Evaluation

Building a specific and exhaustive database for the target application is a crucial step to achieve good tuning of the IPC. Indeed, this database is required during modeling, optimization and control learning. From a statistical point of view, selected images should reflect the frequency of any image contents during the IPC operation and express all its versatility. Input images and output data of the IPC must be absolutely evaluated:

- Output evaluation is necessary during off-line IPC adjustment. This type of evaluation is largely discussed by researchers, even if the studies involve a single algorithm at each time. It remains a critical step, all the more because each IPC is specific and should dispose of its own evaluation criteria.

- Input evaluation is much less investigated. It is required during the IPC inline operation. Achieving adaptive and automatic IPC tuning implies to extract from input images relevant descriptors, that is to say they are closely related with IPC optimal tuning for each image. Image descriptors also enable to lower the initial dimension of the tuning problem (image size in  $n^2$  pixels), as each image pixel contributes to the tuning. Practically, a parameter vector lowers this dimension to the gray level number ( $\approx n$ ), using an histogram computed over the image.

## 2.3 IPC Modeling and Optimization

Modeling parameter influence is carried out through the design of experiments [14]. This tool is common in the industry but has been only recently introduced for machine vision applications [15]. It consists in modeling in a minimal number of trials the effects of simultaneous changes of IPC parameters.

During the experiments, the IPC is considered as a black box whose factors (tuning parameters) influence the effects (values of the criteria for output image evaluation). Tuning only one parameter at the same time can not lead to an optimal setting and testing all parameter configurations would lead to combinatorial explosion. So the goal is to identify the parameters which are really influent (high values for  $|a_i|$ ) and their strong interactions (high values for  $|a_{ij}|$ ) in relation to the effects. These notations refer to a polynomial model whose coefficients are estimated by a least square method:

$$y = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_k x_k + a_{12} x_1 x_2 + \dots + a_{1k} x_1 x_k + \dots + a_{1..k} x_1 \dots x_k \quad (1)$$

A preliminary task consists in specifying for each factor an interval which bounds the experimental domain. This implies a real knowledge of how each algorithm works on image data. Practically, at each trial, every factor is set to its low or high mode, depending on -1 or +1 value in the normalized experiment matrix. So, each design of

experiments is well defined by its experiment matrix whose line number refers to the number of trials and column number refers to the number of tested parameters. The interpretation of the experiments by variance analysis will confirm if the model obtained is really meaningful.

The amount of computing remains very high as the same trials must be repeated on a large number of test images to obtain statistical evidence. So, no optimal tuning is obtained for a given image, only an average tuning for the IPC itself. The parameter influencing significantly the quality of results are identified and the strong interactions between them are also detected, so that only the latter are considered for further IPC programming tasks.

After that, for each particular test image of the database, the optimal tuning of the IPC parameters still need to be searched. This is typically an optimization process. The average tuning obtained before provide valid initial conditions to the search process and the high and low modes of the influent parameters bound the exploration domain

To obtain the optimal parameter tuning for the IPC, we have to look for methods not based on the local gradient computing as it is not available here. The simplex method enables to explore the experimental domain and to reach maxima using a simple cost function to guide the search direction [16]. Practically, a figure of  $n+1$  points of a  $n$ -dimension space is moved and warped through geometric transformations in the parameter space, until a stop condition on the cost function is verified.

## 2.4 IPC Control

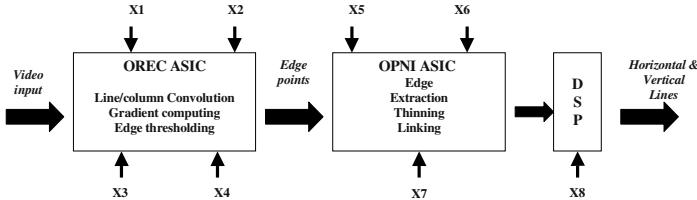
When the optimal tuning of all IPC parameters has been found for a large subset of the image database, the descriptors of these input images are computed. Together with the optimal IPC tuning parameters corresponding to the test images, it constitutes a learning base for a neural network. The other part of the database is reserved for the test of the neural network. It is a convenient tool for estimating the complex relation between the input image descriptors and the computed optimal values of the tuning parameters. At this time, if the selected descriptors are relevant for tuning purpose, the neural network should converge. Finally, after the preceding steps devoted to statistical modeling, numerical optimization and learning, the IPC is toggled into operational mode and image processing tuning parameters are continuously adapted to the characteristics of new input images.

# 3 Application to a Road Image Processing Chain

## 3.1 IPC Overview

This application is part of the French PREDIT program and has been integrated in the SPINE project (intelligent passive security) intended to configure an intelligent airbag system in pre-crash situations. An on board multi-sensor system (EEV high speed camera + SICK scanning laser range finder) integrated in a PEUGEOT 406 experimental car classifies potential front obstacles and estimates their collision course in less than 100 ms [17,18,19]. To respect this drastic real time constrain, low and medium image processing have been implemented in hardware with the support of MBDA

company. It consists in two ASIC circuits [20] embedded with a DSP into an electronic board interfaced with the vehicle CAN bus. As the first tests realized by the industrial car part supplier FAURECIA demonstrated that a static tuning is ineffective against road image variability, an automatic and adaptive tuning based on the approach presented above has been successfully adopted. Eight re-configurable parameters can be modified at any time (Fig. 2).

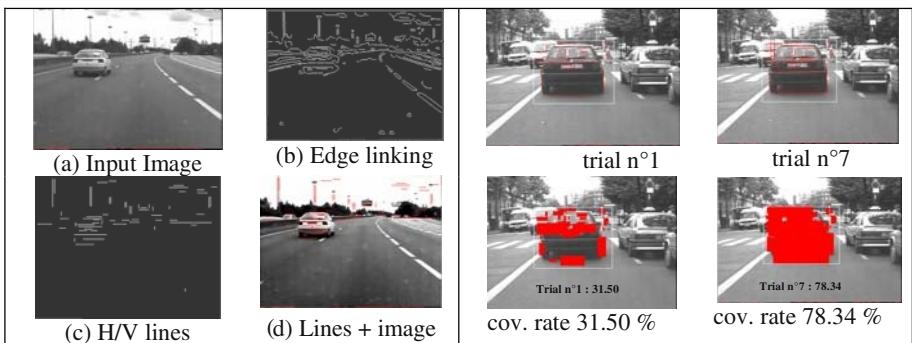


**Fig. 2.** Tunable parameters of the road image processing chain : Canny-Deriche filter coefficients ( $X_1$ ), image amplification coefficients ( $X_2$ ), edge low and high threshold values ( $X_3$ ,  $X_4$ ), the number of elementary automata for contour closing ( $X_5$ ), polygonal approximation threshold ( $X_6$ ), small segments elimination threshold ( $X_7$ ) and the approximation threshold for horizontal and vertical lines ( $X_8$ )

### 3.2 Output Evaluation

The IPC should extract from the image horizontal and vertical lines (Fig.3), which, after perceptual grouping, well describe the potential obstacles in front of the experimental vehicle. Then, output evaluation is based on the number, repartition and length of these segments inside a window of interest specified by the scanning laser range finder. We have proposed a quality evaluation criteria called covering rate, which can be computed for different parameter tunings (Fig.3).

The covering rate is defined as follows: for each horizontal or vertical segment S, we introduce a rectangular-shaped mask  $M_S$ , centered on this segment and whose width is



**Fig. 3.** H/V line Extraction (left) and IPC Output evaluation (right)

proportional to the length of that segment. For each image pixel  $(i,j)$  in the  $W$  window of interest ( $n_x$  and  $n_y$  dimensions), we define a function  $f(i,j)$  by:

$$f(i,j) = 1 \text{ if } \exists S \in W \mid (i,j) \in M_S \text{ and } f(i,j) = 0 \text{ otherwise} \quad (2)$$

The covering rate is given by:

$$r = 1 \setminus n_x.n_y. \sum_{i,j} f(i,j) \quad (3)$$

and it is clear that  $0 \leq r \leq 1$ . An intuitive graphical interpretation exists for the covering rate: it is simply the part of the window of interest which is covered by the superimposition of the masks associated to the set of segments detected by the IPC and so, it will be expressed in this paper as a percentage.

### 3.3 Statistical Modeling

Three design of experiments have been implemented inside the modeling module: a  $2^{k-p}$  factorial fractional design with 16 trials [21] to select the really influent parameters ( $X_1$ ,  $X_6$ ,  $X_8$ ), a Rechschaffner design [22] with 37 trials and finally a quadratic design with 27 trials, by adding an intermediate zero mode to detect non linearity. By using two modes for the tuning of each parameter (Tab.2), 256 different IPC output can be compared from any given input image.

To give an example, we present just below the experiment matrix and the covering rate for the set of trials of the first design of experiments (Tab.1). These designs have been tested on 180 input images selected from a video sequence of over 30 000 city and motorway frames. Selected images should correspond to representative road scenarios

**Table 1.** Experiment matrix (fact.  $2^{8-3}$  design) and averaged outputs. It can be observed that the so called generators are  $X_5=2.3.4$ ,  $X_6=1.3.4$ ,  $X_7=1.2.3$ ,  $X_8=1.2.4$ , where the numbers refer to the columns 1 to 4

and reveal enough different input descriptors. A statistical model has been deduced and validated by measuring R-Square (0.897) and Mallow C(p) indicator . It is given by :

$$Y = 40.2 + 2.06 X_1 + 0.74 X_2 - 2.47 X_6 + 5.30 X_8 - 0.92 X_1 X_2 + 0.95 X_6 X_8 \quad (4)$$

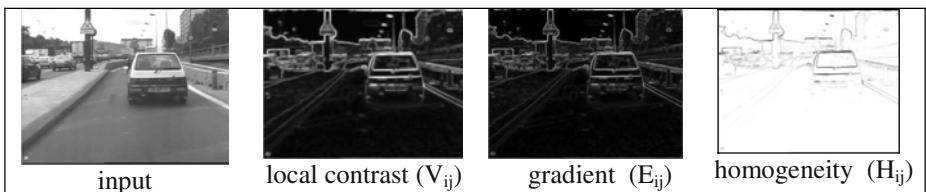
Higher order interactions between the parameters are practically neglected and non significant parameters should not generate significant interactions. The covering rates obtained for the different trials provides an average tuning for the IPC parameters. This static tuning can not be optimal for each given input image but it allows to initialize the optimization of the Nelder & Mead algorithm based on the simplex method. This algorithm then computes all the parameter optimal values corresponding to each tested input image.

**Table 2.** Modes for all the design of experiments

Factor	Parameter	Mode	
X <sub>1</sub>	Canny-Deriche ( $\alpha$ )	0.5	1
X <sub>2</sub>	Image amplification	33	63
X <sub>3</sub>	Edge low threshold	5	15
X <sub>4</sub>	Edge high threshold	15	30
X <sub>5</sub>	Elementary automata (edge closing)	26	30
X <sub>6</sub>	Polygonal approx. threshold	5	6
X <sub>7</sub>	Small segments suppression threshold	5	10
X <sub>8</sub>	Horizontal/ Vertical approx. threshold	1	3

### 3.4 Input Evaluation

Before to start the learning of the control module, input descriptors should be found to characterize input images. The homogeneity histogram [23] of the input image has been selected to take in account regions with uniform shade (vehicle paintings) as well as homogeneous texture (road surface) (Fig.4). The homogeneity measure combines two



**Fig. 5.** Homogeneity measure. Each pixel (i,j) with a  $H_{ij}$  measure verifying  $H_{ij} > 0.95$  is taken in account in the histogram computed on the 256 gray levels of the input image

local criteria: the local contrast  $\sigma_{ij}$  in a  $d \times d$  ( $d=5$ ) window centered on the current pixel (i,j) using  $\mu_{ij}$ , the average of the gray levels computed inside the same window and a gradient measure  $e_{ij}$  in another  $t \times t$  ( $t=3$ ) window. The measure of intensity variations

$e_{ij}$  around a pixel (i,j) is computed by Sobel operator based on the components of the gradient at pixel (i,j) in x and y directions. These measures are normalized using  $V_{ij} = \sigma_{ij} / \max \sigma_{ij}$  and  $E_{ij} = e_{ij} / \max e_{ij}$ . The homogeneity measure is finally expressed by:

$$H_{ij} = 1 - E_{ij} \cdot V_{ij} \quad (5)$$

### 3.5 IPC Control

We have used a simple multi layer perceptron as a control module. It is composed of 256 input neurons (homogeneity histogram levels over the 256 gray levels), 48 hidden neurons (maximum speed convergence during the learning) and 3 output neurons corresponding to the 3 influent tuning parameters of the IPC. During the learning step carried out on 75 % of the 180 input images, we can observe the decrease of the mean absolute error (MAE) between optimal parameters and that computed by the network. (convergence over 400 iterations)

## 4 Results

The separate validation set is constituted by the 25 % remaining test images. An essential task is to control on these images that the tuning parameter computed by the network, not only are close enough to the optimal values, but also enable to obtain really good results at the IPC output, that is to say line groups are well detected (covering rate is close to that derived from simplex algorithm) (Tab. 3).

**Table 3.** Neural network performance

<b>Neural network</b>	<b>Hidden neurons</b>	<b>Learning</b>		<b>Test</b>	
		Parameter MAE (%)	Covering rate Abs. error (%)	Parameter MAE (%)	Covering rate Abs. error (%)
RN3	48	1.4	8.06	23.7	9.53
RN8	80	0.8	3.55	28.6	13.17

We can notice that the neural network only based on influent tuning parameters (RN3) is the most robust during the test step although errors are larger during the learning step. We compare in (Tab.4) the covering rates (output quality evaluation) averaged on the set of 180 test images extracted from the 30 000 image sequence, depending on the tuning process adopted. Without any adaptive tuning facility (static averaged tuning issued from the design of experiments), the results are low; when the best trial obtained from a design of experiments is used for the tuning, the results are high enough but this method can not be applied in real time situations; the results obtained with the simplex (SPL) method are naturally optimal but the price for that is the prohibitive time for parameter space exploration; finally, the neural networks (RN) obtain high results, especially the 3 output network, with a negligible computing cost ( $\approx$  computation of the input image descriptors).

**Table 4.** Comparison of several tuning methods

	Static	RN8	RN3	Fact. design	Rech. design	Quad. design	SPL8	SPL3
<b>Mean Cov. Rate(%)</b>	34.84	45.17	<b>49.64</b>	50.68	51.06	55.02	58.34	59.17
<b>Cost by Image</b>	0	Hist. Comp.	Hist. Comp.	16 trials	37 trials	27 trials	100 trials	60 trials

We have intentionally mentioned in this table the results obtained for a 8 parameter tuning: we can easily verify that the tuning of the 5 parameters considered little influent by the design of experiments, is useless.

## 5 Conclusion

These interesting results obtained in the context of an image processing chain (IPC) dedicated to road obstacle detection highlight the interest of the experimental approach for the adaptive tuning of an IPC. We demonstrated that only three parameters have to be precisely tuned and that with IPC knowledge combined with input image description an automatic tuning can be obtained in real time for each sequence image. The main reasons for this efficiency are simple: contrary to previous works, the IPC is globally optimized, from a great number of real test images, and by adapting image processing to each input image. We are currently testing this approach on other applications in which the image typology, image processing operators and data evaluation criteria for inputs as well as outputs are still specific. This should enable us to unify and generalize this methodology for better IPC performance.

## Acknowledgments

This research program is supported by the French PREDIT Program and by Europe FSE grant.

## References

1. R.M. Haralick, " Performance characterization protocol in computer vision ", *ARPA Image Understanding Workshop*, Monterey, CA, 667-673, 1994.
2. P.Courtney, N.Thacker, A. Clark "Algorithmic modeling for performance evaluation" *Workshop on perf. characteristics of vision algorithms* Cambridge, April 19 1996- 13p.
3. W. Forstner, "10 Pros and cons against performance characterization of vision algorithms", in *Workshop on Performance Characteristics of Vision Algorithms*, April 1996.
4. Kevin W. Bowyer, P. Jonathon Phillips "Empirical Evaluation Techniques in Computer Vision" June 1998, Wiley-IEEE Computer Society Press ISBN: 0-8186-8401-1 262 pages
5. P. Meer, B. Matei, K. Cho, " Input Guided Performance Evaluation ", *Theoretical Foundations of Computer Vision*, pp. 115-124, 1998.
6. I.T. Phillips and A.K. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems", *IEEE PAMI*, vol. 21, pp. 849-870, 1999.

7. J. Blanc-Talon, V. Ropert "Evaluation des chaînes de traitement d'images" *Revue Scientifique et Technique de la Défense* n°46 2000 p.29-38
8. S. Philipp-Foliguet, " Evaluation de la segmentation ", ETIS, Cergy-Pontoise, Mars 2001.
9. N.Sebe, Q Tian, E. Loupias, M. Lew, T. Huang "Evaluation of salient point techniques" *CIVR 02* July 10-15 2002 London
10. P.Rosin, E. Ioannidis "Evaluation of global image thresholding for change detection" *Pattern Recognition Letters* 24 (2003) 2345-2356
11. Y. Yitzhaky, E. Peli "A method for objective edge detection evaluation and detection parameter selection" *IEEE PAMI* vol 25 n°8 Aug. 2003 p.1027-1033.
12. V. Ropert, " Proposition d'une architecture de contrôle pour un système de vision ", *Thèse de l'Université René Descartes* (Paris 6), France, Décembre 2001.
13. I. Levner, V. Bulitko "Machine learning for adaptive image Interpretation" *The Sixteenth Innovative Appl. of Art. Intell. Conf.* (IAAI-04) July 27-29, 2004 San Jose, (CA) USA- 7p.
14. P. Schimmerling, J-C. Sisson, A. Zaïdi, "Pratique des Plans d'Expériences", *Lavoisier Tec & Doc*, ISBN 2-743-00239-5, 1998.
15. S. Treuillet, " Analyse de l'influence des paramètres d'une chaîne de traitements d'images par un plan d'expériences ", *19<sup>e</sup> colloque GRETSI'03*, 8-11 sept. 2003.
16. Margaret H. Wright The Nelder-Mead Simplex Method: Recent Theory and Practice *Int. Symposium on Mathematical Programming Lausanne*, EPFL, August 24-29, 1997
17. A. Domingues, Y. Lucas, D. Baudrier, P. Marché, " Détection et suivi d'objets en temps Détection et suivi d'objets en temps réel par un système embarqué multi capteurs ", *GRETSI'01*, Toulouse, Septembre 2001
18. A. Domingues, "Système embarqué multicapteurs pour la détection d'obstacles routiers - Développement du prototype et réglage automatique de la chaîne de traitement d'images ", *Thèse de l'Université d'Orléans*, France 15 Juillet 2004.
19. Y. Lucas, A. Domingues, M. Boubal, P. Marché, "Système de vision embarqué pour la détection d'obstacles routiers " *Techniques de l'Ingénieur – Recherche Innovation*. 2005
20. P. Lamaty, " Opérateurs de niveau intermédiaire pour le traitement temps réel des images ", *Thèse de Doctorat*, France 2000.
21. A. Fries, J. Hunter, "Minimum aberration  $2^{k-p}$  designs", *Technometrics*, vol. 22, pp. 601-608, 1980.
22. R. L. Rechtschaffner, "Saturated fractions of  $2n$  and  $3n$  factorial designs", *Technometrics*, 9, pp. 569-575, 1967.
23. H. Cheng, Y. Sun, "A hierarchical approach to color image segmentation using homogeneity", in *IEEE transactions on image processing* 9(12): 2071-2082, 2000.

# A Graph Representation of Filter Networks

Björn Svensson, Mats Andersson, and Hans Knutsson\*

Department of Biomedical Engineering, Medical Informatics  
Center for Medical Image Science and Visualization  
Linköping University, Sweden  
`{bjosv, matsa, knutte}@imt.liu.se`

**Abstract.** Filter networks, i.e. decomposition of a filter set into a layered structure of sparse subfilters has been proven successful for e.g. efficient convolution using finite impulse response filters. The efficiency is due to the significantly reduced number of multiplications and additions per data sample that is required. The computational gain is dependent on the choice of network structure and the graph representation compactly incorporates the network structure in the design objectives. Consequently the graph representation forms a framework for searching the optimal network structure. It also removes the requirement of a layered structure, at the cost of a less compact representation.

## 1 Introduction

Filter networks for efficient convolution [1] is a technique for designing and implementing sets of multidimensional finite impulse response (FIR) filters with significantly lower complexity compared to standard convolution. Successful design and practical use of filter networks is shown in e.g. [2], where local 3-D structure is extracted from visual data. This paper shows how elementary graph theory can be used to compactly represent a filter network. The graph representation aims to form a framework for future work on a more general strategy for design of filter networks. Several other design techniques fits within this framework and a few samples are given to illustrate the concept.

Filter networks for efficient convolution are single-rate systems and are not to be confused with the widely used multi-rate systems. The purpose of multi-rate system is data compression using signal decimation rather than efficient filtering. Identical filtering can be performed by a single-rate system, using sparse subfilters. This is a classic example of trading memory for speed. Interpolation is not necessary for single-rate systems which increase the computational efficiency, while multi-rate systems require less amount of memory due to signal decimation. Design of multi-rate systems often starts out from the condition of perfect reconstruction, which is not a requirement for general single-rate systems designed for efficient convolution.

---

\* The financial support from The Swedish Agency for Innovation, VINNOVA, The Swedish Foundation for Strategic Research, SSF, and ContextVision AB is gratefully acknowledged.

## 2 FIR Filter Design

The amount of research devoted to the classic problem of FIR filter design indicates its importance as a fundamental operation in signal processing applications. Much interest has been directed towards the design of 1-D equiripple low-pass filters. An equiripple filter is obtained by finding the Chebyshev approximation to the desired frequency response. This problem was solved by McClellan-Parks [3], using the Remez exchange algorithm.

The Remez exchange algorithm is based on the alternation theorem, which is only applicable on 1-D filters. Due to this, focus for design of multidimensional filters mainly turned towards the weighted least mean squares (WLMS) technique [4] and the eigenfilter approach [5]. Both approaches allow constraints or objectives in the spatio-temporal domain, which is not the case for the traditional approach presented by McClellan-Parks.

### 2.1 Least Squares Design

The problem of FIR filter design is to choose the complex coefficients  $\mathbf{c} \in \mathbb{C}^N$  of the discrete impulse response  $\tilde{f}(\xi)$ , with the closest fit to, in the general case, a number of desired functions. In this paper two objectives,  $\alpha$  in the frequency domain and  $\beta$  in the spatio-temporal domain are used. Each coefficient  $c$  is associated with a discrete spatio-temporal position  $\xi \in \mathbb{Z}^n$  on a Cartesian grid. The efficiency of a filter, i.e. the number of multiplications and additions per data sample, is determined by the number of nonzero filter coefficients  $N$ , which grows exponentially with the signal dimensionality  $n$ .

A direct implementation yields a frequency response  $\tilde{F}(\mathbf{u})$ , which is linear w.r.t. the nonzero coefficients  $\mathbf{c} = [c_1, c_2, \dots, c_N]^T \in \mathbb{C}^N$  of the impulse response  $\tilde{f}(\xi)$  due to the Fourier transform  $\tilde{F}(\mathbf{u}) = \mathcal{F}\{\tilde{f}(\xi)\}$  in Eq. 1.

$$\tilde{F}(\mathbf{u}) = \sum_{\mathbb{Z}^n} \tilde{f}(\xi) \exp(-i\xi^T \mathbf{u}) = \sum_{k=1}^N c_k \exp(-i\xi_k^T \mathbf{u}) \quad (1)$$

The frequency objective  $\alpha(\mathbf{c})$  in Eq. 2 describe a WLMS-error between the frequency response  $\tilde{F}(\mathbf{u})$  of  $\tilde{f}(\xi)$  and the desired frequency response  $F(\mathbf{u})$ . Similarly the spatio-temporal objective  $\beta$  in Eq. 3 is expressed as the WLMS-error between the impulse response  $\tilde{f}(\xi)$  and the desired impulse response  $f(\xi)$ .

$$\alpha(\mathbf{c}) = \int_{\mathbb{U}} W(\mathbf{u}) \left| F(\mathbf{u}) - \tilde{F}(\mathbf{u}) \right|^2 d\mathbf{u}, \quad \mathbb{U} = \{ \mathbf{u} \in \mathbb{R}^n : |u_i| \leq \pi \} \quad (2)$$

$$\beta(\mathbf{c}) = \sum_{\mathbb{Z}^n} w(\xi) \left| f(\xi) - \tilde{f}(\xi) \right|^2 \quad (3)$$

The optimal impulse response  $\tilde{f}^*$  with nonzero coefficients  $\mathbf{c}^*$  is here obtained by simultaneously minimizing  $\alpha(\mathbf{c})$  and  $\beta(\mathbf{c})$  in Eq. 4.

$$\mathbf{c}^* = \arg \min_{\mathbf{c} \in \mathbb{C}^N} \alpha(\mathbf{c}) + \beta(\mathbf{c}) \quad (4)$$

The Chebyshev approximation corresponds to having a frequency objective  $\alpha(\mathbf{c})$ , where the  $l_2$ -norm is replaced by the  $l_\infty$ -norm, while ignoring the spatio-temporal objective. For 2-D filters [6] these equiripple designs can still be achieved by in an iterative manner updating the weighting functions. The equiripple property is however rarely desired in image processing applications.

## 2.2 Multispace Design and Weighting

Filtering, using FIR filters is of course application dependent. The choice of spaces and associated weighting functions should therefore naturally be based on a priori information. As opposed to most design methods, this is here incorporated in the filter design. In this paper the design objectives are restricted to the Fourier space and the spatio-temporal space, but the least squares approach can easily be extended to arbitrary number of objectives in multiple design spaces [7].

In the Fourier space, the weighting function preferably favors a close fit to the desired frequency response for the signal frequencies most critical for the application in mind. Consequently, the errors are distributed among the less critical frequencies. A natural approach is to use a weighting function, which favors a close fit to the most common frequencies, i.e. the expected signal and noise spectra. The spatio-temporal objective can be used to favor locality, an important property to prevent the filters for mixing up different events present in the data.

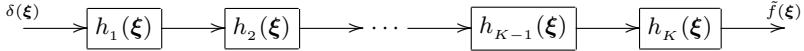
## 3 Design of Cascaded Filters

The idea of decomposing filters into cascaded sparse subfilters  $h_k(\xi)$  (see Fig. 1) for computationally efficiency is far from new and an overview of early work on this topic is given in [8]. Early research mostly concerns narrowband 1-D filters, since sharp transition bands are hard to implement efficiently using standard FIR filters. Cascaded filters for other purposes have not been very popular, since it in general offers no improvement of computational efficiency for 1-D filters.

The approaches are divided into those that reuse the same subfilter (see e.g. [9]) and those who use different subfilters. The two single-rate approaches with most impact, the frequency-response masking technique [10] and interpolated FIR filters are closely related. The frequency response masking technique is actually a small filter network rather than cascaded subfilters.

Interpolated FIR filters, known as IFIR, was introduced in [11]. The basic idea is to use one sparse subfilter followed by a nonsparse subfilter acting as an interpolator in the spatio-temporal domain. By recursively applying this strategy, the interpolator itself can be divided into a new pair of subfilters, a sparse one and an interpolator. In this way a filter sequence with length larger than 2 is obtained.

As opposed to design of 1-D filters, cascaded multidimensional filters generally improve the computational efficiency. Just consider convolution between two filters of spatial size  $N^n$ . The resulting filter is of size  $(N + N - 1)^n$  and



**Fig. 1.** The computational load of filtering, i.e. the number of nonzero coefficients, can be reduced by designing and implementing the impulse response  $\tilde{f}(\xi)$  using cascaded subfilters  $h_k(\xi)$ ,  $k = 1, 2, \dots, K$

the computational gain for this particular filter is  $(N + N - 1)^n / (2N^n)$ . The main problem is how to decompose the desired filter response into subfilters that accurately enough can be described using sparsely scattered filter coefficients. For certain classes of filters, like for instance Cartesian separable filters, there is natural way to decompose the filters. But multidimensional filters are in general Cartesian nonseparable and no general factorization theorem exists for decomposition into sparse subfilters. Still heuristic approaches show examples of efficient decompositions of Cartesian nonseparable filters into cascaded subfilters (see e.g. [7]).

### 3.1 Objective Functions

Replacing  $\tilde{F}(\mathbf{u})$  in Eq. 2, 3 with the product of all subfilter frequency responses  $H_k(\mathbf{u})$  yields the least squares objectives Eq. 2 valid for arbitrary choice of subfilters. To simplify notation, the impulse response  $\tilde{f}(\xi)$  in Eq. 3 is now expressed as  $\tilde{f}(\xi) = \mathcal{F}^{-1}\left\{\tilde{F}(\mathbf{u})\right\}$ , i.e. the inverse Fourier transform of the frequency response.

$$\alpha(\mathbf{c}) = \int_{\mathbb{U}} W(\mathbf{u}) \left| F(\mathbf{u}) - \prod_k H_k(\mathbf{u}) \right|^2 d\mathbf{u} \quad (5)$$

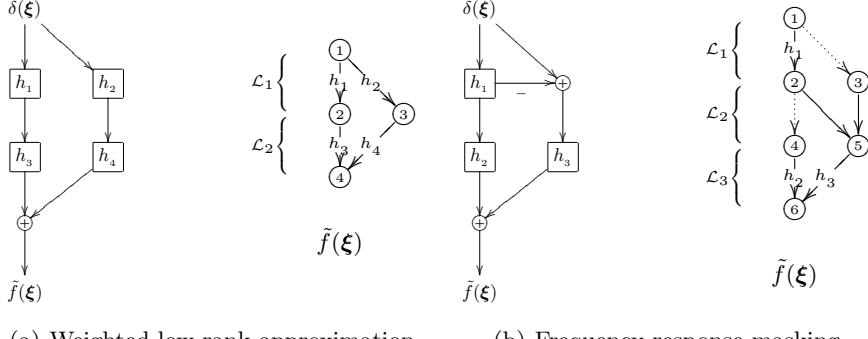
$$\beta(\mathbf{c}) = \sum_{\mathbb{Z}^n} w(\xi) \left| f(\xi) - \mathcal{F}^{-1}\left\{ \prod_k H_k(\mathbf{u}) \right\} \right|^2 \quad (6)$$

## 4 Filter Networks

There are three fundamental properties of the filter networks that contribute to computational efficiency. Firstly, intermediary results may contribute to multiple output, when designing a set of filters. Then, cascaded subfilters admit a lower number of filter coefficients compared to a direct implementation. Finally, sparse subfilters further decrease the number of nonzero filter coefficients, which contribute to a lower computational load. In software implementations a convolver that exploits sparsity is required [12].

### 4.1 Graph Representation

A general FIR filter network can be represented as a directed acyclic graph  $\mathcal{G} = (\mathcal{S}, \mathcal{H})$ , where the nodes  $s \in \mathcal{S}$  are summation points and an arc  $(s_i, s_j) \in \mathcal{H} \subseteq \mathcal{S} \times \mathcal{S}$  are a subfilter connecting  $s_i$  and  $s_j$ . Fig. 2 shows two small examples of such graph representations.



**Fig. 2.** To the left in both (a) and (b) the standard representation of filter networks is shown. The corresponding graphs are shown to the right. Subfilters  $h$  are represented by arcs and the nodes are summation points. The layers are denoted  $\mathcal{L}_k$ . The frequency-response masking example requires the use of dummy filters  $H(u) = 1$  (the dotted arcs), to be able to draw a graph structured in layers. Weighted low rank approximations for 2-D are implemented as a sum of parallel branches with cascaded subfilters. The second order low rank approximation, i.e. two branches, is represented as a filter network with two layers

Elementary graph theory defines two nodes  $s_i, s_j$  as  $\dots \dots$ , when  $(s_i, s_j)$  is an arc. The entire graph can then be described by an adjacency matrix  $\mathbf{A}$ , with elements  $a_{ij}$  defined by Eq. 7.

$$a_{ij} = \begin{cases} 1, & (s_i, s_j) \in \mathcal{H} \\ 0, & (s_i, s_j) \notin \mathcal{H} \end{cases} \quad (7)$$

$A, \dots P$  is defined as a sequence of distinct nodes  $s_1, s_2, \dots, s_k$  such that  $s_i$  and  $s_{i+1}$  are adjacent for all  $i = 1, 2, \dots, k - 1$ . The length of a path  $P$  is for unweighted graphs defined as the number of arcs in  $P$ . Thus the adjacency matrix  $\mathbf{A}$  describes the paths of length 1 between every pair of nodes. Further on  $\mathbf{A}^k$  describes the all paths of length  $k$  between any pair of nodes. Consequently a path matrix  $\mathbf{P}$  as in Eq. 8 contains all paths between every pair of nodes.

$$\mathbf{P} = \sum_{k=0}^{\infty} \mathbf{A}^k \quad (8)$$

Let us now instead of just saying there is a relation between two nodes, label the relation  $(s_i, s_j)$  by  $H(u)$  representing the transfer function from node  $s_i$  to node  $s_j$ . Each element  $a_{ij}, p_{ij}$  in  $\mathbf{A}, \mathbf{P}$  then represent the transfer function between the nodes  $s_i$  and  $s_j$ . When studying filter networks, these transfer functions are of great interest, especially the ones relating the input node to the output nodes. For the small network example in Fig. 2(a),  $\mathbf{A}$  and  $\mathbf{P}$  is given by Eq. 9. For all examples in this paper row-wise numbering of the nodes from left to right is used as shown in Fig. 2.

$$\mathbf{A} = \begin{bmatrix} 0 & H_1 & H_2 & 0 \\ 0 & 0 & 0 & H_3 \\ 0 & 0 & 0 & H_4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 1 & H_1 & H_2 & H_1H_3 + H_2H_4 \\ 0 & 1 & 0 & H_3 \\ 0 & 0 & 1 & H_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

The transfer functions, with the numbering used, from the input to the output are represented by the rightmost elements on the first row  $p_{1o}$ , where  $o$  denotes the indices of the output nodes. For the example in Fig. 2(a) the output  $\tilde{\mathbf{F}}(\mathbf{u})$  given by Eq. 10 is found as element  $p_{14}$  in  $\mathbf{P}$ .

$$\tilde{\mathbf{F}}(\mathbf{u}) = \mathcal{F}\{\tilde{f}(\xi)\} = H_1(\mathbf{u})H_3(\mathbf{u}) + H_2(\mathbf{u})H_4(\mathbf{u}) = p_{14} \quad (10)$$

Clearly there is a more compact representation for the filter network output  $p_{1o}$ . For layered structured networks  $p_{1o}$  is obtained by decomposing  $\mathbf{A}$  into  $\mathbf{A}_k$ , where  $\mathbf{A}_k$  denotes the adjacency between nodes from layer  $k$  to layer  $k+1$ . The reverse relations is not necessary, since there are no relations between nodes from layer  $k+1$  to layer  $k$ . Thus  $\mathbf{A}_k$  is not quadratic and contains fewer zero elements. The output  $\tilde{\mathbf{F}} = [\tilde{F}_1(\mathbf{u}), \tilde{F}_2(\mathbf{u}), \dots, \tilde{F}_K(\mathbf{u})]^T$  of the design example is then given by Eq. 11.

$$\tilde{\mathbf{F}} = \mathbf{A}_2^T \mathbf{A}_1^T = [H_3 \ H_4] \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = H_1(\mathbf{u})H_3(\mathbf{u}) + H_2(\mathbf{u})H_4(\mathbf{u}) \quad (11)$$

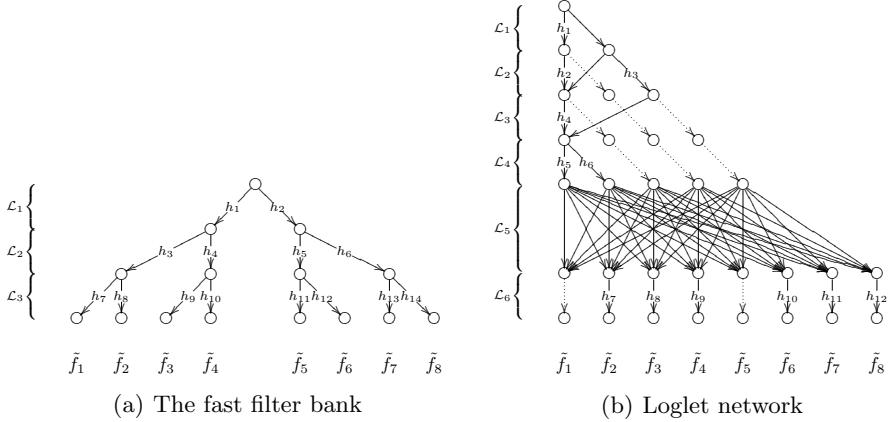
The compact representation can be used for arbitrary filter networks, since a layered structure can be obtained by inserting dummy filters with transfer function  $H(\mathbf{u}) = 1$  to extend all paths to have an equal length. The example in Fig. 2(b) requires dummy filters to be represented compactly as in Eq. 12.

$$\tilde{\mathbf{F}} = \mathbf{A}_3^T \mathbf{A}_2^T \mathbf{A}_1^T = [H_2 \ H_3] \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} H_1 \\ 1 \end{bmatrix} = H_1(\mathbf{u})H_2(\mathbf{u}) + (1 - H_1(\mathbf{u}))H_3(\mathbf{u}) \quad (12)$$

## 4.2 Network Examples

The graph representation forms a general framework and many approaches fit within this framework. A few samples, conceptually close to the filter network approach in [1] are given. Firstly, frequency-response masking and the fast filter bank are briefly presented as two closely related 1-D samples. Then the widely used weighted low rank approximation constitute an example of decomposition of 2-D filters into sparse subfilters. Finally a 2-D version of the 3-D loglet network in [2], forms an example to show the use of graph representation on a larger filter network.

The frequency-response technique [10] can be thought of as generalizations of the IFIR approach. Frequency-response masking (FRM), shown in Fig. 2(b) allows an even sparser representation compared to the IFIR technique. This is achieved by reusing the filter response  $h_1(\xi)$  forming  $1 - h_1(\xi)$  (if causality is ignored). Applying interpolators  $h_2, h_3$  to each of these subfilter output filters  $F(\mathbf{u})$  in Eq. 12 with sharp transition bands with arbitrary bandwidth can be designed.



**Fig. 3.** The fast filter bank to the left is a single-rate system implemented as a tree-structured network. Here an 8 channel example is shown. This structure is also equivalent to a graph representation of an 8-point fast Fourier transform butterfly. The 2-D loglet network is represented by the network with 6 layers to the right. Arcs not annotated are single coefficient subfilters or dummy filters (dotted)

FRM can be applied in a multi-stage approach, by recursively designing the interpolators using FRM. It is also possible to analyze the branches separately i.e. having a multiple output network by not performing the summation after filtering with  $h_2$ ,  $h_3$  in Fig. 2(b). This approach is used to derive the tree-structured fast filter bank (FFB) for efficient convolution in [13]. In fact, also the fast Fourier transform butterfly implemented with only 2 sparsely scattered subfilter coefficients (see e.g. [14]) can be represented using this structure shown in Fig. 3(a).

Most work on efficient convolution in multidimensional applications concerns factorization of the  $n$ -D desired frequency responses to achieve approximations in branching network structures using 1-D subfilters (see Fig. 2(a)). Weighted low rank approximations [15, 16] using singular value decomposition (SVD) to find the desired frequency responses for the 1-D subfilters is the most common technique. The parallel branches corresponding to the largest singular values then forms the implemented filter as in Eq. 11. Due to the lack of non-iterative SVD for  $n$  larger than 2 most research on WLRA is limited to 2-D.

The loglet network presented is represented by the graph in Fig. 3(b). The output  $\tilde{\mathbf{F}} = [\tilde{F}_1(\mathbf{u}), \tilde{F}_2(\mathbf{u}), \dots, \tilde{F}_8(\mathbf{u})]^T$  is given in Eq. 13 and constitute a basis for extracting features like orientation, velocity and local structure [17] in two different scales. The upper part of the network, from input to the output of layer  $L_4$ , forms 5 radial basis functions denoted  $s_5$  in Eq. 13, 14. The 5 basis functions and their relation to preceding subfilters are visualized in Fig. 4. The notation  $c^{(i,j)}$ , used represent single coefficient subfilters from node  $s_i$  to node  $s_j$ .

$$\tilde{\mathbf{F}} = \mathbf{A}_6^T \mathbf{A}_5^T \underbrace{\mathbf{A}_4^T \mathbf{A}_3^T \mathbf{A}_2^T \mathbf{A}_1^T}_{s_5} \quad (13)$$

**Fig. 4.** The input to layer  $\mathcal{L}_5$  in the loglet network in Fig. 3(b) is computed as in Eq. 14. The transfer functions  $H_k(\mathbf{u})$  in these matrices are here visualized in the Fourier domain

The network output are then composed by filtering a linear combination (layer  $\mathcal{L}_5$ ) of these 5 basis functions with directional filters in layer  $\mathcal{L}_6$ . This computation is shown in Fig. 5.

$$\mathbf{s}_5 = \begin{bmatrix} H_5 & 0 & 0 & 0 \\ H_6 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H_4 & 0 & c^{(6,7)} \\ c^{(4,8)} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H_2 & c^{(3,4)} \\ c^{(2,5)} & 0 \\ 0 & H_3 \end{bmatrix} \begin{bmatrix} H_1 \\ c^{(1,1)} \end{bmatrix} \quad (14)$$

### 4.3 General Design Objectives

Actually, the problem of designing layered filter networks is similar to that of designing cascaded filters. Each layer in the filter network can be thought of as one subfilter in a filter sequence. As a consequence of the graph representation the objectives  $\alpha(\mathbf{c})$  in Eq. 5,  $\beta(\mathbf{c})$  in Eq. 6 used for cascaded subfilters can be generalized to represent filter networks in Eq. 15, 16. Note that the argument order, using the product operator can not be changed since  $\mathbf{A}_i$  and  $\mathbf{A}_j$  are not commutative.

$$\alpha(\mathbf{c}) = \int_{\mathbb{U}} \left( \mathbf{F} - \prod_k \mathbf{A}_k \right)^T \mathbf{W} \left( \mathbf{F} - \prod_k \mathbf{A}_k \right) d\mathbf{u} \quad (15)$$

$$\beta(\mathbf{c}) = \left( \mathbf{f} - \mathcal{F}^{-1} \left\{ \prod_k \mathbf{A}_k \right\} \right)^T \mathbf{w} \left( \mathbf{f} - \mathcal{F}^{-1} \left\{ \prod_k \mathbf{A}_k \right\} \right) \quad (16)$$

Sequential convolution fits within this approach, since cascaded filters constitute a directed acyclic graph, where each  $\mathbf{A}_k$  is a  $1 \times 1$  matrix. Note that Eq. 15, 16 is only valid for layered structured networks, but since all filter networks can be redrawn in such a way this is not a limitation.

$$\left[ \begin{array}{c} \text{filter 1} \\ \text{filter 2} \\ \text{filter 3} \\ \text{filter 4} \\ \text{filter 5} \\ \text{filter 6} \\ \text{filter 7} \\ \text{filter 8} \end{array} \right] = \left[ \begin{array}{ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{filter 1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{filter 2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{filter 3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{filter 4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{filter 5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{filter 6} \end{array} \right] \mathbf{A}_5^T \mathbf{s}_5$$

$\mathbf{A}_6$

**Fig. 5.** The output  $\tilde{\mathbf{F}}$  of the loglet network in 3(b) is given by Eq. 13. Here the frequency responses are showing how the directional filters in layer  $\mathcal{L}_6$  forms the output from linear combinations of the input to layer  $\mathcal{L}_5$  shown in Fig. 4

## 5 Discussion

Design of filter sets using filter networks offers a manifold of opportunities to increase the computational efficiency. If similarities between the filters in the set can be exploited, subfilters can contribute to multiple output. To fully exploit this property it is necessary to search for a good network structure.

Choosing the network structure optimally is however a very difficult task, since it require joint optimization of the network structure, the discrete spatio-temporal positions of the filter coefficients and the coefficient values.

The graph representation presented in this paper forms a framework for design of filter networks, which incorporates the network structure in the design objectives and removes the requirement of the layered network structure previously used. The framework simplifies analysis and visualization of how the network structure influences the objective functions.

## References

1. M. Andersson, J. Wiklund, and H. Knutsson. Filter networks. In *Proceedings of Signal and Image Processing (SIP'99)*, Nassau, Bahamas, October 1999. IASTED. Also as Technical Report LiTH-ISY-R-2245.
2. B. Svensson, M. Andersson, and H. Knutsson. Filter networks for efficient estimation of local 3d structure. In *Proceedings of the IEEE-ICIP*, Genoa, Italy, September 2005.

3. J. H. McClellan and T. W. Parks. A unified approach to the design of optimum FIR linear phase digital filters. *IEEE Trans. Circuit Theory*, CT-20:697–701, 1973.
4. D. W. Tufts and J. T. Francis. Designing digital low pass filters - Comparison of some methods and criteria. *IEEE Trans. Audio Electroacoust.*, AU-18:487–494, August 1970.
5. P. Vaidyanathan and T. Nguyen. Eigenfilters: A new approach to least-squares FIR filter design and applications including nyquist filters. *IEEE Transactions on Circuits and Systems*, 34(1):11–23, 1987.
6. J.L. Aravena and Guoxiang Gu. Weighted least mean square design of 2-d fir digital filters: the general case. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 44(10):2568 – 2578, Oct. 1996.
7. H. Knutsson, M. Andersson, and J. Wiklund. Advanced filter design. In *Proceedings of the Scandinavian Conference on Image analysis*, Greenland, June 1999. SCIA.
8. T. Saramäki and A. Fam. Subfilter approach for designing efficient fir filters. In *Proceedings IEEE International Symposium on Circuits and Systems*, pages 2903–2915, June 1988.
9. J. Kaiser and R. Hamming. Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, ASSP-25:415–422, Oct. 1977.
10. Y. C. Lim. Frequency-response masking approach for the synthesis of sharp linear phase digital filters. *IEEE Trans. Circuits and Systems*, CAS-33:357–364, Apr. 1986.
11. Y. Neuvo, Dong Cheng-Yu, and S. Mitra. Interpolated finite impulse response filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(3):563 – 570, June 1984.
12. J. Wiklund and H. Knutsson. A generalized convolver. Report LiTH-ISY-R-1830, Computer Vision Laboratory, SE-581 83 Linköping, Sweden, April 1996.
13. Yong Ching Lim and B. Farhang-Boroujeny. Analysis and optimum design of the ffb. In *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'94)*, volume 2, pages 509 – 512, May 1994.
14. Y.C. Lim and B. Farhang-Boroujeny. Fast filter bank (ffb). *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(5):316 – 318, May 1992.
15. J. Shanks, S. Treitel, and J. Justice. Stability and synthesis of two-dimensional recursive filters. *IEEE Transactions on Audio and Electroacoustics*, 20(2):115 – 128, June 1972.
16. R. Twogood and S. Mitra. Computer-aided design of separable two-dimensional digital filters. *IEEE Trans. Acoust. Speech, Signal Processing*, 25:165–169, 1977.
17. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.

# Optimal Ratio of Lamé Moduli with Application to Motion of Jupiter Storms

Ramūnas Girdziušas and Jorma Laaksonen

Helsinki University of Technology, Laboratory of Computer and Information Science,  
P.O. Box 5400, FI-02015 HUT, Espoo, Finland,  
`Ramunas.Girdziusas@hut.fi, Jorma.Laaksonen@hut.fi`

**Abstract.** Fluid dynamics models the distribution of sources, sinks and vortices in imaged motion. A variety of different flow types can be obtained by specifying a key quantity known as the ratio of the Lamé moduli  $\lambda/\mu$ . Special cases include the weakly elliptic flow  $\lambda/\mu \rightarrow -2$ , often utilized in the Monge-Ampère transport, the Laplacian diffusion model  $\lambda/\mu = -1$ , and the hyper-elliptic flow  $\lambda/\mu \rightarrow \infty$  of the Stokesian dynamics. Bayesian Gaussian process generalization of the fluid displacement estimation indicates that in the absence of the specific knowledge about the ratio of the Lamé moduli, it is better to temporally balance between the rotational and divergent motion. At each time instant the Lamé moduli should minimize the difference between the fluid displacement increment and the negative gradient of the image mismatch measure while keeping the flow as incompressible as possible. An experiment presented in this paper with the interpolation of the photographed motion of Jupiter storms supports the result.

## 1 Introduction

Fluid-based estimation of motion in images covers two major application areas: medical image registration [2] and optical flow estimation [3]. In this work we focus on the latter type of applications, in particular, we attempt to interpolate the cloud movement in Jupiter by means of fluid motion estimation.

The fluid dynamics framework [2] extends the previous works based on linear elasticity aiming at the estimation of large nonlinear deformations. A common problem here is that the displacement estimation qualitatively depends on the ratio of the Lamé moduli  $\lambda/\mu$ , usually defined in an  $\dots$  way. Various spatial models of the Lamé moduli have already been considered [7]. However, the possibility of modifying these parameters adaptively in time has not gained considerable attention so far.

Research on fluid dynamics applications often avoids the two key concepts which are in modern analysis known as the  $\dots$  and  $\dots$  of the Lamé functional. A safe constraint  $\mu > 0$  and  $\lambda > 0$  is often employed, which, however, misses weakly elliptic modes of the fluid flow. On the other hand, ellipticity constraint  $\mu > 0$  and  $\lambda + 2\mu > 0$  is not enough to ensure that a numerical algorithm will inherit essential properties of the continuous problem.

This work utilizes the ellipticity and point-wise stability concepts and indicates how two resulting sets of constraints on the Lamé moduli affect the models built to estimate a fluid flow.

After simplifying a fluid dynamics model, we view the estimation of displacement increment at each time instant as obtaining the average of a certain Gaussian process (GP). As a result, the Lamé moduli can be interpreted as hyperparameters that maximize the Bayesian evidence criterion [6]. This criterion has already been used to select the weighting factors of some deformable surface energies in order to correct three-dimensional SPECT images [4], but its implications to time-dependent fluid flow are very much unexplored.

The paper is organized into four remaining sections. A concise summary of fluid dynamics models and key difficulties in identifying fluid motion are presented in Section 2. Section 3 develops the Bayesian evidence-based optimality criterion and a realistic algorithm which could be used in temporal adaptation of the Lamé moduli. Numerical results are presented in Section 4. Here we first depict a diversity of deformations caused by a variety of different Lamé moduli and later present the result of an experiment with the interpolation of cloud movement in Jupiter atmosphere. Finally, conclusions are drawn in Section 5.

## 2 Fluid Dynamics Models of Image Motion

Let us assume that the template and target images are smooth real-valued functions  $\varrho_0(\mathbf{x}) \in \mathbb{H}^1(\Omega)$  and  $\varrho_T(\mathbf{x}) \in \mathbb{H}^1(\Omega)$ , respectively, defined on the spatial domain  $\mathbf{x} \in \Omega = [0, 1] \times [0, 1]$ , i.e. they belong to the space of functions which are square-integrable up to their first order spatial derivatives. The goal will be to find the displacement vector field  $\mathbf{u}_T(\mathbf{x}) \in [\mathbb{H}_{\partial\Omega}^1(\Omega)]^2$ , defined over the spatial domain  $\mathbf{x} \in \Omega$ , which maps the target image  $\varrho_T(\mathbf{x})$  onto the template image  $\varrho_0(\mathbf{x})$  via  $\varrho_T(\mathbf{x}) = \varrho_0(\mathbf{x} - \mathbf{u}_T(\mathbf{x}))$  and also satisfies some additional requirements.

Consider first the functionals  $[\mathbb{H}_{\partial\Omega}^1(\Omega)]^2 \rightarrow \mathbb{R}$ , defined by

$$\mathcal{J}[\mathbf{u}] = \|\varrho_0(\mathbf{x} - \mathbf{u}) - \varrho_T(\mathbf{x})\|^2, \quad (1)$$

$$\mathcal{R}_{\lambda,\mu}[\mathbf{u}] = \|\nabla^\perp \cdot \mathbf{u}_t\|^2 + (\lambda/\mu + 2)\|\nabla \cdot \mathbf{u}_t\|^2, \quad (2)$$

where  $\|\cdot\|^2$  denotes the squared Euclidean norm integrated over the domain  $\Omega$  and  $\nabla^\perp \cdot \mathbf{u} = \partial_{x_1} u^{(2)} - \partial_{x_2} u^{(1)}$  is a two-dimensional curl operator. We next define the fluid motion model which: (i) sets initial displacement values to  $\mathbf{u}_0 = \mathbf{0}$  and (ii) updates them by adding small displacement increments  $\mathbf{u}_{t+1} = \mathbf{u}_t + \Delta \mathbf{u}_t$  according to

$$\Delta \mathbf{u}_t \approx \arg \inf_{\delta \mathbf{u}} (\delta \mathcal{J}[\mathbf{u}_t] + \delta^2 \mathcal{R}_{\lambda,\mu}[\mathbf{u}_t]). \quad (3)$$

Here the functionals  $\delta \mathcal{J}[\mathbf{u}_t]$  denote the first and second variation of Eqs. (1) and (2) respectively, where the latter one is also weighted with  $c_t > 0$ . The first functional is linear in  $\delta \mathbf{u}$  whereas  $\delta^2 \mathcal{R}[\mathbf{u}_t]$  is quadratic in  $\delta \mathbf{u}$ . The term  $\delta^2 \mathcal{J}[\mathbf{u}_t]$  encompasses the second-order effects of the optical flow and can be neglected.

The existence and uniqueness of the stationary state  $\mathbf{u}_{t \rightarrow T}$  then entirely depends on the Lamé functional  $\mathcal{R}_{\lambda,\mu}[\mathbf{u}_t]$ .

Eq. (3) models a fluid motion without inertial terms. Here we further simplified it by neglecting the multiplying matrix  $\mathbf{I} - \nabla \mathbf{u}_t$  which would normally appear in Eq. (3) due to the Eulerian reference frame [2]. Our experience indicates that in the applications considered below this multiplying factor is not important.

It is the absence of the linear-in- $\delta\mathbf{u}$  or  $\nabla \delta\mathbf{u}$  term  $\delta\mathcal{R}$  that makes Eq. (3) crucially different from variational regularization problem solving [5]. This is the key defining feature of fluid motion: whenever a fluid particle gets pushed, there will be no re-action to force it back. The smoothness here is only due to the particle's interaction with its neighbors via  $\delta^2\mathcal{R}[\mathbf{u}_t] \neq 0$ .

Eq. (3) possesses unique minima if the Lamé moduli satisfy either one of the following requirements:

$$\text{Ellipticity: } \mu > 0, \quad \lambda + 2\mu > 0, \quad (4)$$

$$\text{Point-wise stability: } \mu > \mu_0, \quad \lambda + \frac{3}{2}\mu > \lambda_0, \quad (5)$$

for some positive constants  $\mu_0$  and  $\lambda_0$ . A computational analysis of these two sets of constraints is presented in Section 3.2. We emphasize that Eqs. (4) and (5) are different in the three-dimensional case [5].

Numerous fluid motion estimation algorithms essentially differ in the application-dependent image error measure  $\mathcal{J}$  and in the variants of the Lamé functional. The case  $\lambda/\mu = -1$  can be implemented by performing the steepest descent on the image mismatch with Gaussian smoothing of the displacement increments. Compressible Stokes models and Monge-Ampère transport allow a more diverse range of displacements by considering  $\lambda/\mu \rightarrow \infty$  and  $\lambda/\mu = -2$  cases, respectively.

We will apply Eq. (3) with the boundary conditions  $\delta\mathbf{u}|_{\partial\Omega} = \mathbf{0}$ . Another constraint will be imposed on the maximum Euclidean norm of the displacement increment  $\max \|\delta\mathbf{u}\|_2 = \kappa$ , which removes the need to consider time-dependent regularization weight  $c \equiv c(\kappa)$ . In general, the absolute values of the Lamé moduli depend on the constant  $\kappa$ . However, we believe that the choice of  $\kappa$  does not produce qualitatively different types of flow, it rather affects its stability. In what follows we replace the ratio  $\frac{\lambda}{\mu}$  by a single parameter  $\lambda$ .

Finding the optimal ratio of the Lamé moduli is challenging for many reasons. If Eq. (3) includes the extra term  $\delta\mathcal{R}$ , e.g. when it becomes linear elasticity equation, this ratio can be chosen according to the material properties of the imaged objects [7]. A more general methodology considers the joint probability distribution of both: the displacements and the Lamé moduli. A research example in this direction is the application of the Kalman filter to estimation of material properties of the myocardial tissues [9].

Nice compressibility patterns hardly exist or it is impossible to track them in the imaged fluid motion. Moreover, the functional  $\mathcal{J}$  is always a very crude approximation to how image intensity values change along the motion path. These discrepancies would require adequate changes in the Lamé moduli. The

weaknesses in choosing the Lamé moduli based on the physics of the problem alone can be especially well seen from Bayesian viewpoint.

### 3 Image Motion with Adaptive Lamé Constants

#### 3.1 Gaussian Process of Fluid Displacement Change

In order to understand the ellipticity implications and to reduce Eq. (3) to a computationally feasible level, let us approximate the Lamé functional with central finite differences. Eq. (3) then becomes

$$\mathbf{P}_{\lambda_t} \Delta \mathbf{u}_t^* = \mathbf{f}_t + \mathbf{b}_{\lambda_t}, \quad (6)$$

where the vector  $\Delta \mathbf{u}_t^*$  consists of the unknown displacement increments at the nodes  $\mathbf{x}_{1:n} = \{\mathbf{x}_i | i = 1, 2, \dots, n = (w - 2)^2, \mathbf{x}_i \in \mathbb{R}^2\}$  of the discrete domain  $\Omega$  with  $w^2$  spatial nodes. The vector  $\mathbf{f}_t \in \mathbb{R}^{2n}$  denotes the components of the image mismatch gradient w.r.t. the displacement at time  $t$ . The elements of the vector  $\mathbf{b}_{\lambda_t}$  depend on the boundary conditions and in our case  $\mathbf{b}_{\lambda_t} = \mathbf{0}$ .

The matrix  $\mathbf{P}_{\lambda_t} \in \mathbb{R}^{2n \times 2n}$  can be expressed via finite-difference matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$ :

$$\mathbf{P}_{\lambda_t} = \begin{pmatrix} (\lambda_t + 2)\mathbf{T}_1^T \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{T}_2 & (\lambda_t + 1)\mathbf{T}_1^T \mathbf{T}_2 \\ (\lambda_t + 1)\mathbf{T}_2^T \mathbf{T}_1 & (\lambda_t + 2)\mathbf{T}_2^T \mathbf{T}_2 + \mathbf{T}_1^T \mathbf{T}_1 \end{pmatrix}. \quad (7)$$

The role of the matrix  $\mathbf{T}_1$  can be interpreted as follows. Each displacement component can be considered as a separate image, whose columns could be joined into a single vector. Multiplying such a vector by the matrix  $\mathbf{T}_1$  from the left produces a vector which contains the central-difference approximation of the derivatives of the displacement component at every internal node in the  $x_1$ -direction. The matrix  $\mathbf{T}_2$  estimates the derivatives in the  $x_2$ -direction. Eq. (7) indicates that the matrix  $\mathbf{P}_{\lambda_t}$  will be positive definite at least when  $\lambda_t \in [-1, \infty)$ .

Intuitively, we can view Eq. (6) as a filter of the noisy vector  $\mathbf{f}_t$ . The following probabilistic interpretation of Eq. (6) helps to develop criterion for the Lamé moduli in a systematic way:

$$p(\mathbf{f}_t | \Delta \mathbf{u}_t, \sigma^2) = \mathcal{N}(\Delta \mathbf{u}_t, \sigma^2 \mathbf{I}), \quad (8)$$

$$p(\Delta \mathbf{u}_t | \mathbf{x}, \lambda_t, \sigma^2) = \mathcal{N}(\mathbf{0}, \sigma^2 (\mathbf{P}_{\lambda_t} - \mathbf{I})^{-1}). \quad (9)$$

Eq. (8) assumes that the vector  $\mathbf{f}_t$  is contaminated by additive Gaussian noise, whose variance is denoted by  $\sigma^2$ . Eq. (9) states that the displacement increments follow zero-mean Gaussian random process  $\Delta \mathbf{u}_t$  with the covariance matrix  $\mathbf{K} \equiv \sigma^2 (\mathbf{P}_{\lambda_t} - \mathbf{I})^{-1} \in R^{2n \times 2n}$ . Eq. (6) now can be re-obtained by applying conditioning [6] of the actual displacement increment  $\Delta \mathbf{u}_t$  on the observation vector  $\mathbf{f}_t$ :

$$\Delta \mathbf{u}_t^* \equiv E[\Delta \mathbf{u}_t | \mathbf{f}_t] = \mathbf{K}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{f}_t = \mathbf{P}_{\lambda_t}^{-1} \Delta \mathbf{f}_t. \quad (10)$$

Eq. (6) defines the mean of the displacement increment  $\Delta \mathbf{u}_t$ , whereas Eqs. (8) and (8) complete it to a full probability distribution. The kernel matrix  $\mathbf{K}$  is not defined uniquely in  $\mathbf{P}_{\lambda_t}$ , but choosing  $\sigma^2 = \|\mathbf{f}_t - \Delta \mathbf{u}_t^*\|^2 / (2n)$  removes the ambiguity.

This interpretation of fluid motion is useful because we can impose a non-informative improper hyper-prior on the Lamé ratio and maximize the loglikelihood (log-evidence)  $p(\mathbf{f}_t | \lambda)$  of the vector  $\mathbf{f}_t$  w.r.t. the Lamé ratio  $\lambda$ . This procedure can be shown to be equivalent to the following minimization problem:

$$\lambda_t^* = \arg \min_{\lambda} \left\{ \underbrace{\frac{1}{2} \ln \frac{|\mathbf{P}_{\lambda}|}{|\mathbf{P}_{\lambda} - \mathbf{I}|}}_{\text{'Incompressibility'}} + \underbrace{n \ln 2\pi\sigma^2}_{\text{'Deregularization'}} + \underbrace{\frac{1}{2\sigma^2} (\Delta \mathbf{u}_t^*)^T \mathbf{P}_{\lambda} \overbrace{(\mathbf{f}_t - \Delta \mathbf{u}_t^*)}^{\text{'Noise'}}}_{\text{'Decorrelation'}} \right\}. \quad (11)$$

We emphasize that Eq. (6) defines the mean function, and thus there are infinitely many possibilities for its probabilistic interpretation. As an example, one could consider a degenerate variant of the GP model Eqs. (8) and (9) which would result in the mean function  $\Delta \mathbf{u}_t^* \equiv E[\Delta \mathbf{u}_t | \mathbf{f}_t]$  equivalent to Eq. (6) in the same way as Eq. (10):

$$p(\Delta \mathbf{u}_t | \mathbf{x}, \lambda_t) = \mathcal{N}(\mathbf{0}, \mathbf{P}_{\lambda_t}^{-1}), \quad \mathbf{f}_t = \mathbf{P}_{\lambda_t} \Delta \mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\lambda_t}), \quad (12)$$

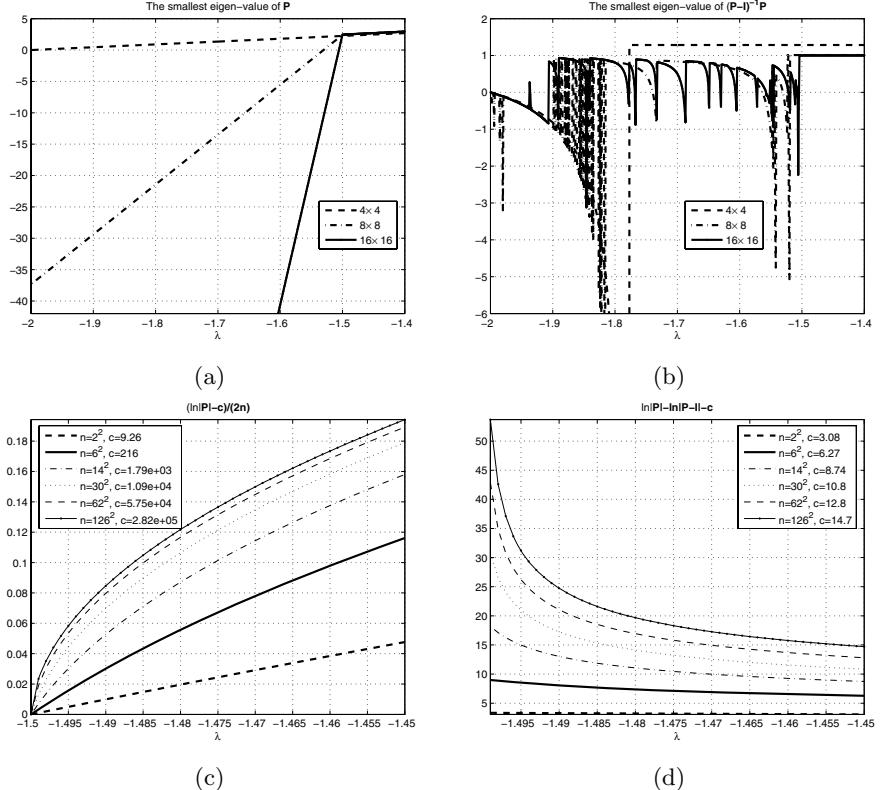
$$\lambda_t^* = \arg \min_{\lambda} \left\{ \frac{1}{2} \ln |\mathbf{P}_{\lambda}| + \frac{1}{2} \mathbf{f}_t^T \Delta \mathbf{u}_t^* + n \ln 2\pi \right\}. \quad (13)$$

The GP model in Eqs. (8) and (9) is more justified than its degenerate counterpart, cf. the argument presented in [8]. A more thorough analysis of Eq. (13) would show that it could be qualitatively similar to Eq. (11), but the utilization of Eq. (11) would require a proper scaling of the matrix  $\mathbf{P}_{\lambda_t}$ .

### 3.2 Computational Analysis

Let us analyze Eq. (7) and the above-derived Bayesian evidence criterion Eq. (11). Fig. 1 indicates how the smallest eigenvalues of the matrices  $\mathbf{P}_{\lambda}$  and  $\mathbf{K} + \sigma^2 \mathbf{I} = (\mathbf{P}_{\lambda} - \mathbf{I})^{-1} \mathbf{P}_{\lambda}$  depend on the Lamé ratio  $\lambda$  and grid size. The eigenvalues are estimated on several small grids, where the interplay between the ellipticity and point-wise stability is the most notable. Fig. 1 shows that the range of allowed Lamé ratio values is confined by the point-wise stability of the Lamé functional stated in Eq. (5) to the interval  $\lambda \in (-1.5, \infty)$ . Notably, the ellipticity constraint  $\lambda \in (-2, \infty)$  does not ensure the positive definiteness of the matrix  $\mathbf{P}_{\lambda}$ , yet the smallest singular value is never equal to zero and therefore a unique solution of Eq. (6) exists even when  $\lambda \in (-2, -1.5]$ . However, Fig. 1b indicates that the covariance matrix of the GP model in Eq. (9) is surely positive definite only when  $\lambda \in (-1.5, \infty)$ .

The estimates of the log-determinants of the matrix  $\mathbf{P}_{\lambda}$  and the covariance matrix of the GP model are presented in Figs. 1c,d. In the case of a larger  $126 \times 126$  grid, the values are estimated by first applying sparse LU decomposition and then summing all the logarithms of the diagonal elements in the U-matrix.



**Fig. 1.** (a) The dependence of the smallest eigenvalue of  $\mathbf{P}_\lambda$  on the ratio of the Lamé moduli  $\lambda$ , and (b) similar analysis concerning the range of positive-definiteness of the covariance matrix of the GP model Eqs. (8) and (9). (c) The dependence of the log-determinant  $\ln|\mathbf{P}_\lambda|$  on the ratio of the Lamé moduli  $\lambda$ , (d) the difference between the log-determinants in Eq. (11)

In connection with Eq. (11), Fig. 1d indicates that we should give preference to large values  $\lambda$ , i.e. the flow is supposed to be incompressible. However, at the same time we seek for the displacement increment  $\Delta \mathbf{u}_t$  most similar to the force field  $\mathbf{f}_t$ , which naturally would require smaller values of  $\lambda$ . The implications of this balance to an example problem will be analyzed in Section 4.

### 3.3 Extended Fluid Motion Estimation

Below we will present a skeleton of the fluid motion estimation algorithm with adaptable Lamé moduli:

Initialize  $\epsilon, \kappa$ , set  $t = 0$ ,  $\mathbf{u}_0 = \mathbf{0}$ .

WHILE  $\|\varrho_t(\mathbf{x}) - \varrho_T(\mathbf{x})\|_2 > \epsilon$ ,

    Compute the vector  $\mathbf{f}_t$ ,

$$(\Delta \mathbf{u}_t, \lambda_t, \sigma_t^2) = \arg \min_{\Delta \mathbf{u}, \lambda, \sigma^2} \left\{ \frac{1}{2} \ln \frac{|\mathbf{P}_\lambda|}{|\mathbf{P}_\lambda - \mathbf{I}|} + n \ln 2\pi\sigma^2 + \frac{1}{2\sigma^2} (\Delta \mathbf{u}_t^*)^T \mathbf{P}_\lambda (\mathbf{f}_t - \Delta \mathbf{u}_t^*) \right\},$$

subject to  $\begin{cases} \mathbf{P}_\lambda \Delta \mathbf{u} = \mathbf{f}_t, \quad \Delta \mathbf{u}|_{\partial\Omega} = \mathbf{0}, \\ \|\Delta \mathbf{u}\|_2 \leq \kappa, \quad \lambda > -1.5, \\ \sigma^2 = \|\mathbf{f}_t^T - \Delta \mathbf{u}\|^2 / (2n). \end{cases}$

$$\mathbf{u}_t = \mathbf{u}_t + \Delta \mathbf{u}_t,$$

$$t \leftarrow t + dt,$$

END.

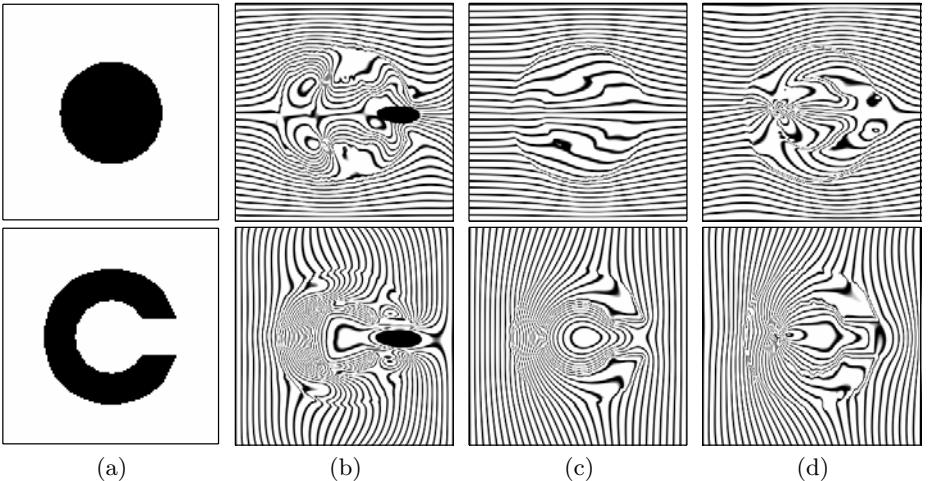
This optimization problem is hard to solve due to the large number of equality constraints and the strong nonlinear coupling of the Lamé moduli with the displacement increments. For example, Section 4 states a problem that requires  $2n = 2 \cdot 400^2$  equality constraints which is far beyond the limits of any general-purpose constrained optimization routine.

To circumvent these difficulties, we perform the minimization in Eq. (11) by direct evaluation of the logevidence over a small number of characteristic values of the Lamé moduli that we believe sufficiently cover the optimization region  $\lambda \in [-2, \infty]$ . We note that the point  $\lambda = -1$  splits the ellipticity region of the Lamé functional into sub-elliptic and super-elliptic smoothing. Thus, we first solve  $\mathbf{P}_\lambda \Delta \mathbf{u} = \mathbf{f}_t$  for a certain  $\Delta \mathbf{u}_t$  dependent on  $\lambda$  by applying pseudo-inverse techniques based on the sparse LU decomposition, and later set  $\sigma_t^2 = \|\mathbf{f}_t - \Delta \mathbf{u}\|^2 / (2n)$ . The logevidence is then evaluated and the computation is repeated for different values of the Lamé ratio. Finally the best displacement increment and the optimal ratio  $\lambda_t$  are chosen which minimize the criterion in Eq. (11).

## 4 Experiments

### 4.1 Circle to Letter ‘c’ Matching

Finding the deformation field between a circle and a letter ‘c’ is a synthetic problem frequently used to assess the quality of various image registration algorithms [2]. The template circle  $\varrho_0(\mathbf{x})$  and target ‘c’  $\varrho_T(\mathbf{x})$  images were generated as described in [2]. The fluid registration algorithm has been applied three times, each time with fixed Lamé parameter values arising from one of the three zones of the ellipticity condition: (i) nearly incompressible flow, (ii) Laplacian smoothing, and (iii) weakly-elliptic flow. The results presented in Fig. 2 show that the displacements strongly depend on the Lamé moduli. Fig. 2a depicts the flow of the nearly incompressible fluid flow. The divergence of displacement increments is penalized by a large  $\lambda$  value, and the flow generates two vortices which can be seen in the left part of the letter ‘c’ image. The relatively smoothest transformation is achieved in the Laplacian case, shown in Fig. 2b, whereas sub-elliptic flow results in source points shown in Fig. 2c. It is important to note that all of the final displacement maps carry the circle into ‘c’ perfectly, while being very different in their nature, depending on what prior knowledge we assume about the flow.

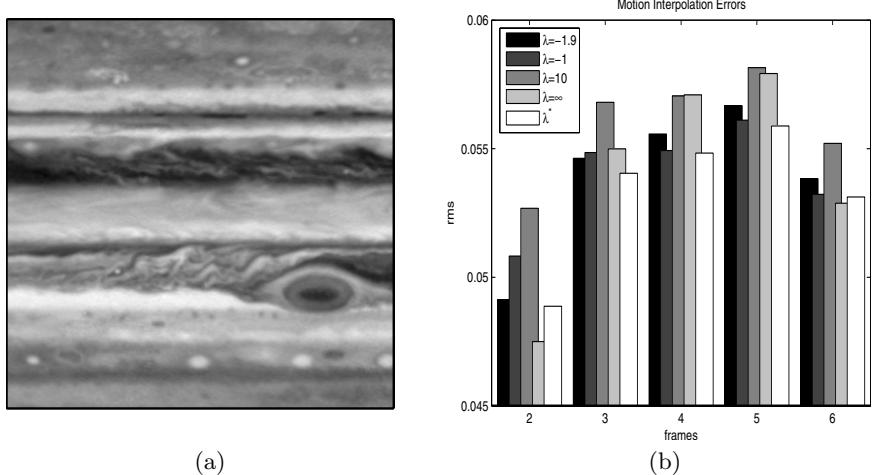


**Fig. 2.** Diversity of fluid motions. (a) Circle to ‘c’ matching results with: (b)  $\lambda = 10$  strong ellipticity (nearly incompressible) fluid flow, (c)  $\lambda = -1$  Laplacian smoothing, (d)  $\lambda = -1.9$  sub-elliptic flow. (b)-(d) The columns from left to right depict deformations of horizontal and vertical grids. All maps deform circle into letter ‘c’ perfectly

The experiments indicate that the use of the Lamé functional  $\mathcal{R}_{\mu,\lambda}$  does not ensure diffeomorphic transformations in general. In fact, all maps develop singularities at relatively early stages of image matching. Symmetry breaking is also notable in all the three cases. The divergence of the displacement increments is often advocated as a term responsible for the image intensity appearance/disappearance rate. If one superimposed the deformed circles in all three cases after few iterations, one could see that the circle expands to the borders of letter ‘c’ faster in the cases of the Laplacian and weak ellipticity models than in the case of the hyper-elliptic model. However, such a fact is of minor importance compared to the solenoidal nature of the flow imposed by large  $\lambda$  values.

#### 4.2 Interpolation of Motion in Jupiter Storms

Interpolation of time events occurring in the atmosphere of astronomic bodies is a very potential application area of fluid motion estimation algorithms. Below we present the problem of interpolating the cloud motion around the Great Red Spot of the planet Jupiter. The first one of seven frames taken by the Cassini-Huygens spacecraft in one rotation period of the planet of Jupiter is shown in Fig. 4.2a. We assume that the exact time of photographing of the images is not known. For simplicity, we estimate the optical flow from the frame No. 1 to the frame No. 7, and try to re-produce the frames No. 2–6 by evaluating deformations which are made equidistant in space. The purpose



**Fig. 3.** An image of Jupiter storms near the Great Red Spot taken by Cassini-Huygens spacecraft in November 2000. Each image is of a resolution  $400 \times 400$  pixels and span 100 degrees around the equator both in East-West and North-South directions [1]. (a) the first frame and (b) the root-mean-square errors between the five estimated and real frames No.2–6

of this example is to show that the optimization of the Bayesian evidence criterion reduces the model errors even when our model makes the strong, possibly invalid, assumption on the constant brightness of the Jupiter storm motion.

The resulting root-mean-squared errors are indicated in Fig. 4.2b. They show that at different time instances different parameter settings are optimal. In the beginning, the motion appears to be incompressible, but later divergent flow produces smaller errors in the frames No. 3–5. Also, there is a notable difference between the results obtained with the ratio  $\lambda = -1$  and the divergence-free flow of  $\lambda = \infty$ , the latter simulated by using the classical projection of the displacement increment onto the divergence-free subspace. The Laplacian flow  $\lambda = -1$  is inferior to our approach in all frames.

It is notable that the optimal temporal reshaping of the Lamé moduli, denoted by  $\lambda^*$ , produces better image interpolation results than any one of the fixed value cases. However, the difference is rather small for several reasons. First, the constant image brightness assumption in Eq. (1) is hardly valid, at least because of the bright dots appearing in the left of the Great Red Spot which are believed to be lightnings [1]. Second, it is clear that the flow around the Great Red Spot is very rotational whereas in other areas it is divergent or simply affine. The problem of the interpolation of Jupiter cloud motion would therefore require spatial models for the Lamé moduli. The task of associating different areas with different flow regimes corresponding to particular Lamé moduli will for now remain as an open question.

## 5 Conclusions

This work has focused on the analysis of fluid-based estimation of motion in images, put into a probabilistic form. The negative spatial gradient of the image mismatch has been interpreted as a time-dependent Gaussian process whose temporal covariance structure depends on the second variation of the Lamé functional. Under this model, the ratio of the Lamé moduli has become a hyperparameter that maximizes the Bayesian evidence of the postulated temporal GP model. More precisely, in the absence of any specific knowledge, the Lamé ratio can be chosen to minimize at each time instant the difference between the fluid flow increment and the external force field, i.e. the negative image gradient. At the same time, the flow should be kept almost incompressible.

The above-discussed optimal reshaping of the Lamé moduli is only an approximative Bayesian inference procedure, and its practical value to the determination of the optimal fluid flow regime remains unclear. The presented method indicates how one can determine the amount of temporal compressibility of the fluid flow by decorrelating the displacement increment with the noise term, which is the difference between the external force field and the estimated displacement increment. When performing the experiments with Jupiter storm images, we have observed that the Lamé ratio introduces a trade-off between the above-mentioned correlation and incompressibility. However, the evidence values are rather flat on the whole region of Lamé values, and the improvements are small.

There are two interesting consequences of this study. First, fixing the compressibility of the imaged motion is always suboptimal. Neither typical incompressible flow  $\lambda/\mu = \infty$ , nor the ‘most assumptionless’  $\lambda/\mu = -1$  Laplacian flow could produce best results. Second, the flatness of the Bayesian evidence criterion can be utilized to detect inadequacies in the model of the image intensity evolution, in particular, validating the optical flow constraint.

We believe that the optimization of the Lamé moduli is important whenever: (i) the postulated model of the image mismatch measure strongly deviates from its true unknown counterpart or (ii) the spatial dependence of the Lamé moduli cannot be neglected but is hard to identify.

## References

1. The Great Red Spot movie. NASA’s Planetary PhotoJournal, PIA02829, NASA/JPL/University of Arizona, November 2002.
2. G. Christensen. *Deformable shape models for anatomy*. Phd thesis, Washington University, 1994.
3. T. Corpetti, É. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):365–380, 2002.
4. G.S. Cunningham, A. Lehovich, and K.M. Hanson. Bayesian estimation of regularization parameters for deformable surface models. *SPIE*, pages 562–573, 1999.
5. B. D.Reddy. *Introductory functional analysis : with applications to boundary value problems and finite elements*. Springer, 1998.

6. M.N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. Ph.d. thesis, Cambridge University, 1997.
7. A. Hagemann. *A Biomechanical Model of the Human Head with Variable Material Properties for Intraoperative Image Correction*. Logos Verlag Berlin, 2001.
8. J. Quiñonero-Candela and C. E. Rasmussen. Analysis of some methods for reduced rank gaussian process regression. In R. Shorten and R. Murray-Smith, editors, *Proc. of Hamilton Summer School on Switching and Learning in Feedback systems*. Springer-Verlag, 2004.
9. Pengcheng Shi and Huafeng Liu. Stochastic finite element framework for simultaneous estimation of cardiac kinematic functions and material parameters. *Medical Image Analysis*, 7:445–464, 2003.

# Extraction and Removal of Layers from Map Imagery Data

Alexey Podlasov and Eugene Ageenko

Department of Computer Science,  
University of Joensuu,  
P.O.Box 111, 80101 Joensuu, Finland  
`{apodla, ageenko}@cs.joensuu.fi`

**Abstract.** Map images are composed of semantic layers depicted in arbitrary color. Layer extraction and removal is often needed for improving readability as well as for further processing. When image is separated into the set of layers with respect to the colors, it results in appearance of severe artifacts because of the layer overlapping. In this way the extracted layers differ from the semantic data, which affects further map image processing analysis tasks. In this work, we introduce techniques for extraction and removal of the semantic layers from the map images. The techniques utilize low-complexity morphological image restoration algorithms. The restoration provides good quality of the reconstructed layers, and alleviates the affect of artifacts on the precision of image analysis tasks.

## 1 Introduction

Nowadays, there exist various services delivering map imagery content on mobile devices. For example, map imaging applications provide user with a view of geographical map for the requested location. It could be also weather, traffic, pollution or any other kind of map. The imagery data is usually obtained from Digital Spatial Libraries [1], and transmitted via wireless network to user's computer or mobile device such as pocket PC, PDA, mobile phone, or similar mobile terminals. Map images need typically only a few color tones but high spatial resolution for representing details such as roads, infrastructure and names of the places. Though maps could be stored in vector format, raster map image is more preferable on a client-side since it is easier to transmit and handle. Raster images are also often used for digital publishing on CD-ROM and in the web.

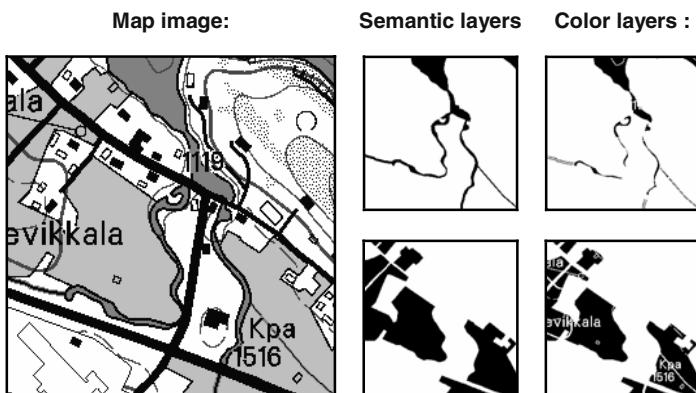
The map images consist of a set of semantic layers, each containing data with distinct semantic content such as roads, elevation lines, state boundaries, water areas, temperature distribution, wind directions, *etc.* Layers are combined and displayed to the user as a generated color image, which is usually produced as follows. First the layers with different semantic nature are combined together by overlapping each other in a predefined order. Then the layers are depicted on the map with appropriate color and finally are transmitted to a client in an image form. After image has been received, client has no knowledge about the initial layer structure.

For example, we consider the topographic images from the NLS topographic database, in particular basic map series 1:20,000 [2]. These images consist of the following semantic layers: *Basic* (roads, contours, labels and other topographic data), *Elevation lines* (thin lines representing elevations levels), *Waters* (solid regions and poly-lines representing water areas and ways), *Fields* (solid polygonal regions), see Figure 1.

Though raster image is well suited for user observation, it cannot be easily used for further processing especially when semantic data is required. For example, when one needs to calculate the area of fields or e.g. the length of sea shore the semantic layer corresponding to the water or field areas must be obtained first. The layers can be extracted from the raster map image through *color separation* process. During this process, the map image is divided into binary layers each representing one color in the original image. The problem is that the separation introduces severe artifacts in places where one layer overlaps another, see Figure 2. These artifacts make separated layer inappropriate for many image analysis tasks. In order to use corrupted layers in further processing a restoration technique should be designed.

Another task is to remove some irrelevant layer(s) from the map image. For example, a car driving user does not need elevation lines on a map. Their presence impairs map readability, which can be improved by the layer removal. Since elevation lines are drawn on the fields and waters, one must apply reconstruction to remove artifacts left by the removed layer.

Moreover, it has been shown that the best compression results for raster map image can be achieved if the image is decomposed into binary semantic layers, which are consequently compressed by the algorithm designed to handle binary data (e.g. JBIG) [3]. Color separation artifacts affect the statistical properties and consistency of the layers, and result in degraded compression performance.



**Fig. 1.** Illustration of map image, its semantic structure, and color layers showing the artifacts due to color separation (with permission of National Land Survey of Finland)

The approximation of the noise-corrupted image to the original is often achieved by using various noise removal techniques, image enhancement or statistical analysis

[4-10]. These approaches are limited to a local neighborhood only, and cannot exploit non-local semantic properties of the image. Semantic approaches exploiting global properties of the image typically have high complexity and are suitable for very special kind of data e.g. thin line graphics or text [11-13]. Existing noise filtering and image enhancement algorithms could not be considered for our problem.

In this work we present two techniques for extraction and, correspondingly, removal of the semantic layers from raster map images using color separation process. The algorithms are based on the morphological restoration algorithms that attempt to recover the semantic layer structure from the separated color layer. The technique is applied for analysis of the semantic data as well as for (on demand) removal of irrelevant semantic content from the map image. The effect of the restoration is limited only to the areas which are degraded due to separation and would be overlapped with other layers during composition. Therefore the color image obtained using combination of the restored layers matches exactly the initial image without any degradation in the quality. Due to simplicity of morphological operations, the method is also fast and simple to implement on the modern mobile devices.

The rest of the paper is organized as follows. Mathematical morphology is briefly introduced in Section 2. Then in Section 3, we introduce new filtering method for layer extraction, and then apply it for layer removal in Section 4. Empirical results are reported in Section 5, and conclusions drawn in Section 6.

## 2 Mathematical Morphology Fundamentals

Mathematical morphology [13] refers to a branch of nonlinear image processing and analysis originally introduced by Georges Matheron [14] and Jean Serra [15]. In mathematical morphology, the binary image space  $E$  is defined as  $E = \mathbb{Z}^2$  (the space of all possible image pixel locations), and the binary image  $X$  as a set  $X \subseteq E$ . The main principle of mathematical morphology is to analyze geometrical and topological structure of an image  $X$  by “probing” the image with another small set  $A \subseteq E$  called a structuring element. The choice of the appropriate structuring element depends on the particular application.

Let us define the dilation of  $X$  by  $A$ , denoted by  $\delta_A(X)$ , as an operator on  $\mathcal{P}(E)$  such as:

$$\delta_A(X) = \bigcup_{a \in A} X_a = \{h \in E \mid \widetilde{A_h} \cap X \neq \emptyset\}, \quad (1)$$

The *erosion* of  $X$  by  $A$ , denoted by  $\varepsilon_A(X)$ , is consequently:

$$\varepsilon_A(X) = \bigcap_{a \in A} X_{-a} = \{h \in E \mid A_h \subseteq X\}, \quad (2)$$

where  $\widetilde{A} = -A = \{-a \mid a \in A\}$  is the reflectance of  $A$  with respect to the origin. Let us also define the translation invariant operator  $\rho_{A,n}$  called *rank operator* as follows:

$$\rho_{A,n}(X) = \left\{ h \in E \mid \text{card}(X \cap A_h) \geq n \right\}. \quad (3)$$

The operator  $\rho_{A,n}(X)$  sets current pixel to be foreground if the amount of foreground pixels in a neighborhood defined by the structuring element is greater than  $n$ . Otherwise the pixel is defined as a background pixel. Since rank operator performs similar to erosion or dilation depending on the value of the rank parameter, it is possible to treat the rank as soft counterpart of classical erosion and dilation operators. In particular

$$\delta_{\tilde{A}}(X) = \rho_{A,1}(X) \text{ and } \varepsilon_A(X) = \rho_{A,n}(X). \quad (4)$$

Sometimes it is important to restrict the area where operator could be applied. This can be accomplished by using conditional operators: if image  $A$  is a subset of image  $M$ , then for any operator  $\psi(A)$  the operator  $\psi(A \mid T)$  is called  $\psi(A)$  conditional relative to mask image  $M$  and is defined as follows:

$$\psi(A \mid T) = \psi(A) \cap T. \quad (5)$$

### 3 Layer Extraction

When original semantic data is unavailable, the task of restoration leaves a lot of freedom for algorithm designer as one can only guess the initial layer structure. The only restriction we have is that the composition of reconstructed layers would be identical to the initial color map. In other words, we can modify the value of the pixels in the layers only if the same pixel value is set in one of the higher priority (overlapping) layers. This means that the change of the pixel value will be seen only in the particular layer, but not in the color image corresponding to the reconstructed layers.

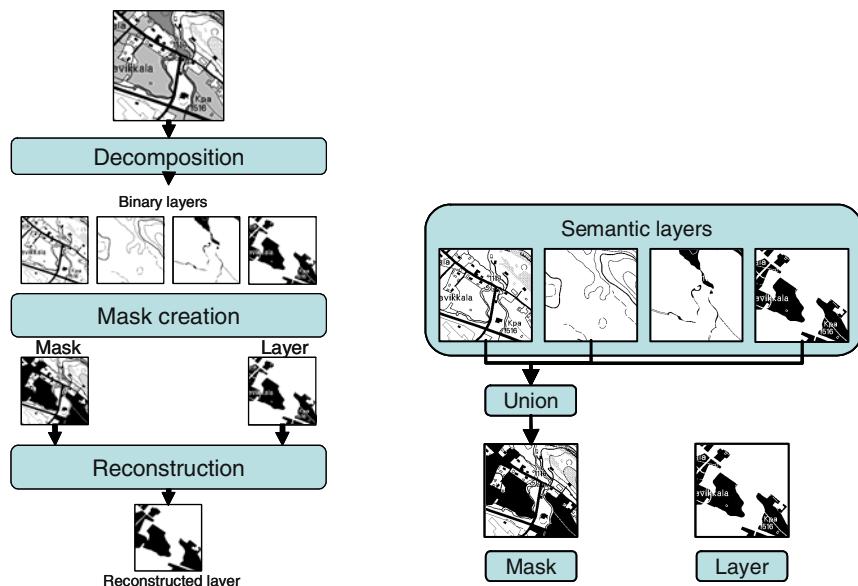
#### 3.1 Algorithm Structure

The algorithm consists of three principal steps: decomposition, mask creation and layer restoration, as outlined in Figure 2. At the first step, the color map image (scanned or obtained from the third party source) is decomposed into a set of binary layers by color separation process. This is done so that each layer represents one color in the original image [3]. On the second step, we define a mask – an area where reconstruction could be performed restricting the reconstruction of the layers to be equal to the original color image. Finally, the layer is extracted and restored using the proposed restoration algorithm. Further we describe in details second and third steps of the algorithm.

#### 3.2 Mask Construction

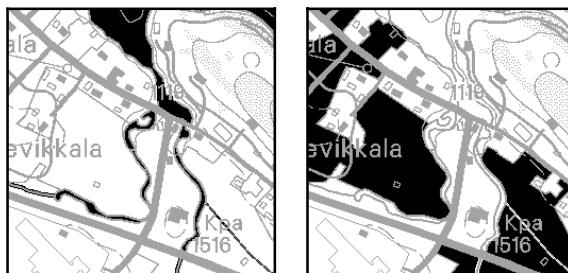
The conditioning mask defines the set of pixels that are allowed to change in the restoration so that the combination of the restored layers would be kept untouched. Since we have assumed that the order of layer overlapping is predefined, the mask for every layer will be a union of all upper-laying layers, see Figure 3. All modifications made to the pixels within the mask area will be overlapped when the combined color image is represented to the user. Depending on the particular case, it is possible to simplify the mask structure by taking into account the nature of the objects represented on the map. For example, we can expect that Waters and Field layers cannot overlap in real-

ity, and therefore, could not overlap on a combined map image. When implementing, we can exclude these layers from the conditioning mask (see Figure 4).



**Fig. 2.** The diagram of the layer extraction algorithm

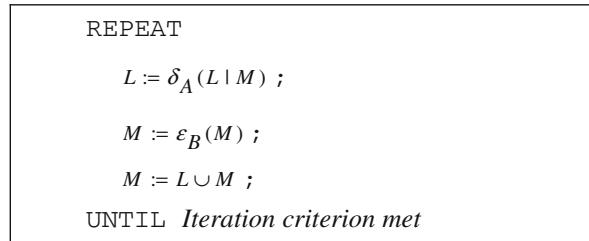
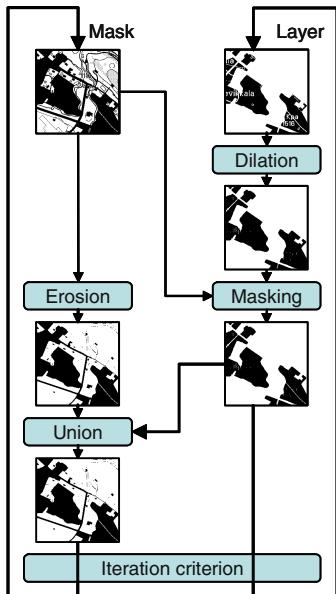
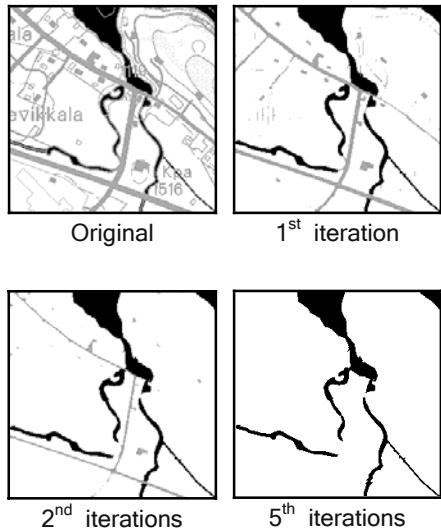
**Fig. 3.** The approach for the mask construction



**Fig. 4.** Water and Fields layers with their masks. Object pixels are shown in black, mask pixels in gray color, and background in white

### 3.3 Layer Restoration

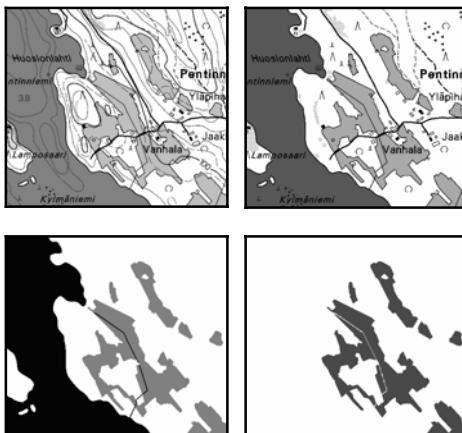
We reconstruct layer iteratively. With every iteration, the object areas spread within the mask, and then the mask area shrinks. The spreading is performed by dilation operator  $\delta A(X)$  and mask shrinking by erosion operator  $\varepsilon A(X)$ . The pseudo-code of the layer restoration algorithm is shown in Figure 5, and its diagram is outlined in Figure 6. The stepwise process of the iterations is illustrated in Figure 7.

**Fig. 5.** Outlined of the layer restoration algorithm**Fig. 6.** Block diagram of the layer restoration algorithm**Fig. 7.** Step-by-step illustration of the dilation with mask erosion. Pixels of the processed object are marked in black, whereas the pixels belonging to the mask – in gray

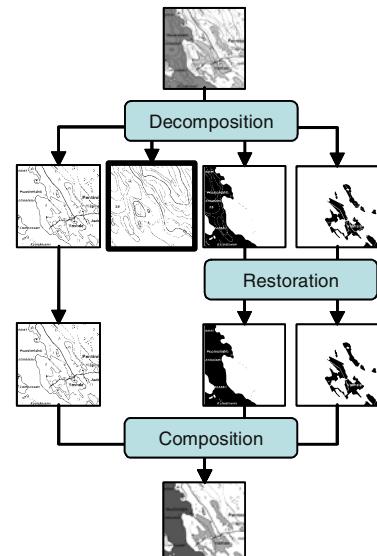
The iterative process is controlled by a stopping criterion. We have investigated two approaches: *Iterate until stability* and *Iterate fixed amount of times*. The first approach assumes that the iterative process will continue until the layer (and mask) converges. The convergence is guaranteed because the erosion sequentially decreases the mask, see Figure 7. We can therefore perform the iterations until the mask equals to the layer itself.

Examination if the mask and layer are equal could be a time consuming operation, especially if the image size is big. To avoid this, we consider the second approach assuming that most of the artifacts being of limited size. Therefore it is sufficient to perform a predefined (small) number of iterations to complete the restoration process. For example, if we suppose that the size of an artifact is less than 4 pixels, on average, only 3 iterations with  $3 \times 3$  block are needed.

As with the conditional closing, an important question is the choice of an appropriate structuring element. There are two structuring elements used in the algorithm. By varying the element used for dilation we can control how fast the object expands over the mask, while varying the element used for erosion we control how fast the mask shrinks. An essential matter is the relationship between the dilation and erosion speeds. Let  $A$  be the structuring element of dilation and  $B$  be the structuring element of erosion. In our investigations, we have tested three cases: objects dilating faster than mask eroding ( $A = \text{block}_{3\times 3}$ ,  $B = \text{cross}_{3\times 3}$ ), objects dilating slower than mask eroding: ( $A = \text{cross}_{3\times 3}$ ,  $B = \text{block}_{3\times 3}$ ), and the case of equal speed ( $A = \text{block}_{3\times 3}$ ,  $B = \text{block}_{3\times 3}$  or  $A = \text{cross}_{3\times 3}$ ,  $B = \text{cross}_{3\times 3}$ ).



**Fig. 8.** Example of the consecutive layer removal (map image fragment, elevation lines removed, basic layer removed, water areas removed)



**Fig. 9.** Block diagram of the layer removal algorithm. Elevation lines layer to be removed is outlined with a black frame

## 4 Layer Removal

The task of layer removal arises when less important layers are needless to the map user, e.g. user driving a car does not need elevation lines. In order to remove a layer, the restoration technique described in Section 3 is first applied to all underlying layers in order of overlapping. Then the restored layers except the removed one are composed into the color image, see Figure 9. The most important criterion here is the quality of the restoration – how closely the restored layer approximates the semantic data. Moreover, in interactive applications the visual appearance of the reconstructed layer becomes essential. Figure 8 illustrates the effect of the successive removal of *Elevation*, *Basic* and *Water* layers.

## 5 Evaluation

The restoration technique has been evaluated on a set of topographic color-palette map images. These images were decomposed into binary layers with distinctive semantic meaning identified by the pixel color on the map. The restoration algorithm has been applied for reconstruction of these semantic layers after the map decomposition process. Both the combined color map images and the binary semantic layers composing these color map images were originally available for testing. This gave us a possibility to compare restored images with their original undistorted counterparts.

The test set consists of five randomly chosen images from the “*NLS Basic Map Series 1:20000*” corresponding to the map sheets No. 431306, 201401, 263112, and 431204. Each image has dimension  $5000 \times 5000$  pixels and corresponds to  $10 \times 10$  km area. Images are composed of four semantic layers:

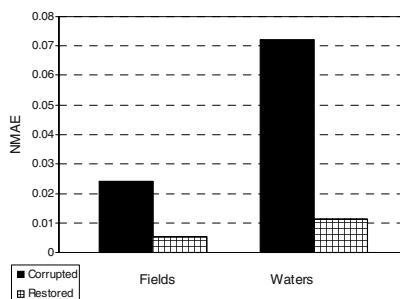
- *Basic* –buildings, protected sites, benchmarks and administrative boundaries;
- *Elevation* – elevation lines;
- *Water* – lakes, rivers, swamps, water streams;
- *Fields* – agricultural areas.

In the following we evaluate the proposed technique by estimating the restoration quality using image *similarity measurement*, *area measurement* and *length of the sea shore*. For image similarity we consider *Normalized Mean Absolute Error* (NMAE), which is a Hamming Distance between images computing the average number of different pixel values. In our context, we compare the original (not affected by decomposition) semantic layers and the layers restored using the proposed technique:

$$NMAE(X, Y) = \frac{\sum_{j=1}^H \sum_{i=1}^W |x_{i,j} - y_{i,j}|}{H \cdot W}, \quad (6)$$

where  $H$  and  $W$  are image dimensions.

We measure NMAE difference between reconstructed *Waters* and *Fields* layers and the original ones and show the improvement comparing to the corrupted layers. We present obtained NMAE difference for every layer separately in total within the test set (see Figure 10).



**Fig. 10.** The average NMAE difference with the original measured for restored. Fields (left) and Waters (right) layers comparing to corrupted ones

In the following, we compare area measured over the original layer with one measured over reconstructed and corrupted layer. The results are presented for Waters and Fields layers separately on average within the test set, see Table 1. Reconstruction reduces the error of the area measurement from near 15-20% to just about 1%. The length of the sea shore is measured as the length of object borders in Waters layer. Results – the length and error over original, corrupted and reconstructed layer are represented in Table 2. Reconstruction reduces the error of shore length calculation from 37% to 1%.

**Table 1.** The area (in pixels) and error comparing to original value (in percents) measured over original, corrupted and reconstructed Waters and Fields layers

Layer	Semantic layers	Corrupted layers		Reconstructed layers	
	Area	Area	%	Area	%
Waters	10 480 893	8 678 605	17.2	10 389 501	0.8
Fields	4 267 983	3 663 960	14.1	4 262 378	0.1

**Table 2.** The length of the sea shore and error comparing to original value (in percents) measured over original, corrupted and reconstructed Waters layer

Semantic layers	Corrupted layers		Reconstructed layers	
Length	Length	%	Length	%
3 115 505	4 279 979	37.3	3 074 954	1.3

## 6 Conclusions

A technique for the extraction and removal of semantic layers from map imagery data has been proposed. The extracted semantic data can be further used for various image analyzing and processing tasks (e.g. area measurement); whereas the layer removal is useful for removing unwanted data from map images due to various reasons (e.g. view cluttering). The proposed technique is based on the separation of the raster map image into color layers and subsequent elimination of the artifacts caused by the color separation process. The iterative restoration algorithm based on the conditional morphological operators is designed for layer reconstruction. The performance of the proposed technique is evaluated qualitatively by comparing the reconstructed layers with the native semantic data, and quantitatively by using standard image analysis tasks. Quality evaluation demonstrates that restoration algorithm can efficiently approximate the map layers. When properly tuned, the algorithm reduces the error in such image analyzing applications as area measurement from 15-20% to about 1%. The reconstructed layers have lesser entropy and can substitute for the color layers in map data storage without any loss of quality. It is possible because the restoration is limited to the area of the images that are overlapped by other layers. Therefore the color raster map image can be obtained by the combination of the reconstructed layers and still remain absolutely identical to the initial non-processed map image.

## References

1. Fox E.A., et al. (Eds.) "Digital Libraries". [Special issue of] *Communications of the ACM* 38 (4), 1995.
2. NLS: National Land Survey of Finland, Opastinsilta 12 C, P.O.Box 84, 00521 Helsinki, Finland. [http://www.nls.fi/index\\_e.html](http://www.nls.fi/index_e.html).
3. Fränti P., Ageenko E., Kopylov P., Gröhn S. and Berger F., "Compression of map images for real-time applications", *Image and Vision Computing*, 22 (13), 1105-1115, November 2004.
4. Pitas, I., Venetsanopoulos A.N., *Nonlinear digital filters: principles and applications*, Boston, Mass.: Kluwer, 1990.
5. Dougherty E.R., Astola J. (eds) *Nonlinear Filters for Image Processing*, SPIE Optical Engineering Press, 1997.
6. Dougherty E.R., "Optimal mean-square n-observation digital morphological filters. Part I: Optimal binary filters", *Computer Vision, Graphics, and Image Processing*, 55: 36-54, 1992.
7. Wah, F.M., "A binary image preprocessor for document quality improvement and data reduction", *Proc. Int. Conf. on Acoustic, Speech, and Signal Processing-ICASSP'86*, 2459-2462, 1986.
8. Ping Z., Lihui C., Alex K.C., "Text document filters using morphological and geometrical features of characters", *Proc. Int. Conf on Signal Processing-ICSP'00*, pp. 472-475, 2000.
9. Randolph T.R., Smith M.J.T., "Enhancement of fax documents using a binary angular representation", *Proc. Int. Symp. on Intelligent Multimedia, Video and Signal Processing*, pp. 125-128, Hong Kong China, 2-4 May 2001.
10. Zheng Q., Kanungo T., "Morphological degradation models and their use in document image restoration", University of Maryland, USA, Technical Report, LAMP-TR-065 CAR-TR-962 N660010028910/IIS9987944, 2001.
11. Ageenko E., Fränti P., "Context-based filtering of document images", *Pattern Recognition Letters*, 21 (6-7), 483-491, Elsevier Science, 2000.
12. Kolesnikov A., Fränti P., Data reduction of large vector graphics, *Pattern Recognition*, 38(3), 2005, pp 381-394.
13. Heijmans H.J.A.M., *Morphological image operators*. Boston: Academic Press, 1994.
14. Matheron G. *Random Sets and Integral Geometry*, J. Wiley & Sons, New York, 1975.
15. Serra J., *Image Analysis and Mathematical morphology*. London: Academic Press, 1982.

# Tensor Processing for Texture and Colour Segmentation

Rodrigo de Luis-García<sup>1</sup>, Rachid Deriche<sup>2</sup>, Mikael Rousson<sup>2</sup>,  
and Carlos Alberola-López<sup>1</sup>

<sup>1</sup> ETSI Telecomunicación, University of Valladolid, Valladolid, Spain  
`{rodlui, caralb}@tel.uva.es`

<sup>2</sup> Projet Odyssée, INRIA Sophia-Antipolis, France  
`{Rachid.Deriche, Mikael.Rousson}@sophia.inria.fr`

**Abstract.** In this paper, we propose an original approach for texture and colour segmentation based on the tensor processing of the nonlinear structure tensor. While the tensor structure is a well established tool for image segmentation, its advantages were only partly used because of the vector processing of that information. In this work, we use more appropriate definitions of tensor distance grounded in concepts from information theory and compare their performance on a large number of images. We clearly show that the traditional Frobenius norm-based tensor distance is not the most appropriate one. Symmetrized KL divergence and Riemannian distance intrinsic to the manifold of the symmetric positive definite matrices are tested and compared. Adding to that, the extended structure tensor and the compact structure tensor are two new concepts that we present to incorporate gray or colour information without losing the tensor properties. The performance and the superiority of the Riemannian based approach over some recent studies are demonstrated on a large number of gray-level and colour data sets as well as real images.

## 1 Introduction

The segmentation of textured images usually relies on the extraction of suitable features from the image. Traditionally, Gabor filters have been used [3, 19], but they yield a lot of feature channels. This drawback was overcome by the use of the structure tensor [12, 1, 2] or its improved versions such as the NLST [6, 4].

After the features have been extracted, a segmentation method that employs this information has to be designed. Lately, level set-based methods [23, 18, 9, 10] have gained much relevance due to their good properties. Besides, they can easily integrate boundary, region and even shape prior information [18, 7, 20, 16, 8].

A very interesting method for the segmentation of textured images was proposed in [21], based on the features extracted by the NLST. The geodesic active regions model is applied to a vector-valued image whose channels are the components of the NLST, obtaining promising results. However, the advantages of the

structure tensor are partially lost because of the vector processing of that information. To our knowledge, no tensor processing has been applied to the structure tensor for texture segmentation. Nevertheless, much work has been done in the field of Diffusion Tensor Imaging (DTI) for the segmentation of tensor-valued images [24, 25, 14, 15, 22, 11]. Level-set based methods were used in [14, 15] for the segmentation of anatomical structures, employing intrinsic tensor dissimilarity measures based on geometric properties of their respective spaces.

In this paper, we propose a novel texture segmentation method which, based on the use of the NLST and its new extended versions for feature extraction, afterwards performs the segmentation in the tensor domain by applying region, level-set based tensor field segmentation tools developed for the segmentation of DTI [14, 15, 22, 25]. This way, the nice properties of the NLST for texture discrimination are fully exploited, as experimental results showed.

Furthermore, new modalities of structure tensor are also proposed that incorporate gray level or colour information while keeping the tensor structure. Altogether, comparative results are shown which indicate that the novel segmentation methods described in this paper yield excellent results and improve the state of the art.

The paper is organized as follows: next section studies the NLST for texture extraction. Afterwards, we review the vector adaptive segmentation methods employed in [21] for texture segmentation, and the tensor schemes proposed in [14, 15, 22] for DTI segmentation. In Section 4, we present the main contribution of this paper, that is, the tensor processing of the NLST for texture segmentation. Besides, we introduce new, improved modalities of the structure tensor incorporating gray or colour information. Section 5 describes the extensive experiments made to test and validate the methods proposed, followed by a discussion of the results. Finally, a brief summary is presented.

## 2 Nonlinear Structure Tensor

For a scalar image  $I$ , the structure tensor is defined as follows [12, 1, 2]:

$$J_\rho = K_\rho * (\nabla I \nabla I^T) = \begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix} \quad (1)$$

where  $K_\rho$  is a Gaussian kernel with standard deviation  $\rho$ , and subscripts denote partial derivatives. For vector-valued images, the following expression is used :

$$J_\rho = K_\rho * \left( \sum_{i=1}^N \nabla I_i \nabla I_i^T \right) \quad (2)$$

The smoothing with a Gaussian kernel makes the structure tensor suffer from the dislocation of edges. To solve this problem, Brox and Weickert [4, 6] propose to replace the Gaussian smoothing by nonlinear diffusion. For vector-valued data, the diffusion equation becomes:

$$\partial_t u_i = \operatorname{div} \left( g \left( \sum_{k=1}^N |\nabla u_k|^2 \right) \nabla u_i \right) \quad \forall i \quad (3)$$

where  $u_i$  is an evolving vector channel, and  $N$  is the total number of vector channels.

The NLST can be obtained, for a scalar image, by applying Eq. 3 with initial conditions  $\mathbf{u} = [I_x^2 \ I_y^2 \ I_x I_y]^T$ . In practice, however, the original image is added as an extra channel because it can provide valuable information, yielding  $\mathbf{u} = [I_x^2 \ I_y^2 \ I_x I_y \ I]^T$ . However, it can be noticed that these components have not the same order of magnitude, which could cause some problems. To solve this problem and force all channels to take values in the same range, a simple normalization is not a good choice, since it would amplify the noise in channels containing no information. Instead, a better solution is to replace the structure tensor by its square root (see [17] for details).

## 3 Adaptive Segmentation

### 3.1 Vector Field Segmentation

In [21], a variational approach was proposed for the segmentation of textured images. Following the work in [19], the image segmentation can be found by maximizing the partition probability  $p(P(\Omega)|I)$  given the observed image  $I$ , where  $P(\Omega) = \{\Omega_1, \Omega_2\}$  is the partition of the image domain  $\Omega$  in two regions. This is equivalent to the minimization of an energy term. Two hypotheses are necessary: all partitions are equally probable, and the pixels within each region are independent. Then, if a Gaussian approximation is used to model each region, let  $\{\mu_1, \Sigma_1\}$  and  $\{\mu_2, \Sigma_2\}$  be the vectors means and the covariance matrices of the Gaussian approximation for  $\Omega_1$  and  $\Omega_2$ . The partition boundary  $\partial\Omega$  can be represented by the level set function  $\Phi$ , and the resulting energy can then be minimized by iteratively estimating the optimal statistical parameters  $\{\mu_i, \Sigma_i\}$  for a fixed level set function  $\Phi$  and evolving the level set function with these parameters. The following system of coupled equations is obtained (see [21] for details):

$$\begin{cases} \mu_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} u(x) dx, \\ \Sigma_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} (u(x) - \mu_i)(u(x) - \mu_i)^T dx \\ \frac{\partial \Phi}{\partial t}(x) = \delta(\Phi) (\nu \operatorname{div}(\frac{\nabla \Phi}{|\nabla \Phi|}) + \log \frac{p_1(u(x))}{p_2(u(x))}) \end{cases} \quad (4)$$

where  $\delta(z)$  is the Dirac function.

Considering identity covariance matrix leads to the well known piece-wise constant Chan-Vese model [7] while considering other covariance matrices (diagonal, full..) allows to discriminate between regions having the same mean but different second order statistics. Finally, if Gaussian approximation for some channels is not appropriate, an estimation of the probability density function based on the parzen window can be performed. [21].

### 3.2 Tensor Field Segmentation

The adaptive segmentation method shown above was designed for vector-valued images, and so the structure tensor has to be converted into a vector leading to the traditional Frobenius norm-based tensor distance. This way, the nice properties of the structure tensor as such are lost. Therefore, a tensor processing of the NLST would be expected to outperform the approach proposed in [21].

For the segmentation of Diffusion Tensor images, a symmetric positive definite (SPD) tensor was interpreted as a covariance matrix of a Gaussian distribution in Wang . . . [24, 25]. Then, the natural distance between two Gaussian pdfs, given by the symmetrized Kullback-Leibler distance, can be a measure of dissimilarity between two Gaussian distributions, represented by SPD tensors.

The symmetrized Kullback-Leibler distance (also called J-divergence) between two distributions  $p$  and  $q$  is given by:

$$d(p, q) = \frac{1}{2} \int (p(x) \log \frac{p(x)}{q(x)} + q(x) \log \frac{q(x)}{p(x)}) dx \quad (5)$$

It is possible to obtain a very simple closed form for the symmetrized Kullback-Leibler distance [24]. Now, let us denote by  $\mathbf{T}_1$  and  $\mathbf{T}_2$  the mean values of the tensor image over the regions  $\Omega_1$  and  $\Omega_2$ . It is possible to model the distribution of the KL distances to  $\mathbf{T}_1$  and  $\mathbf{T}_2$  in their respective domains by the densities  $p_{d,1}$  and  $p_{d,2}$ . Making the assumption that  $p_{d,1}$  and  $p_{d,2}$  are Gaussian of zero mean and variances  $\sigma_1^2$  and  $\sigma_2^2$ , the minimization of the corresponding energy functional can be achieved as follows [14, 15, 22]:

$$\begin{cases} \sigma_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} p_{d,i}^2(x) dx \\ \frac{\partial \Phi}{\partial t}(x) = \delta(\Phi)(\nu \operatorname{div}(\frac{\nabla \Phi}{|\nabla \Phi|}) + \frac{1}{2} \log \frac{p_{d,2}}{p_{d,1}}) \end{cases} \quad (6)$$

This approach has been successfully employed for Diffusion MRI segmentation in [15]. However, as shown in [14, 13], it only considers the parameterized , . . as living in the linear space  $\mathbb{R}^6$  instead of taking into account the Riemannian structure of the underlying manifold, thus being able to define a geodesic distance. It is not possible to find a closed form of the geodesic distance for general distributions, but a closed-form of the geodesic distance between two symmetric positive definite matrices can be found. Indeed, the Riemannian distance, intrinsic to the manifold of the symmetric positive definite matrices, between two SPD matrices  $P_1$  and  $P_2$  is shown to be equal to  $d(P_1, P_2) = \sqrt{\sum_{i=1}^n \ln^2(\lambda_i)}$  where  $\lambda_i, i = 1..n$  are the positive eigenvalues of  $P_1^{-1}P_2$ .

Such an approach and its advantages were presented in [13] where the authors present impressive results on Diffusion Tensor MRI. In this work, we propose to replace the symmetrized KL distance with this Riemannian distance and compare their performance on segmenting textured, coloured images.

## 4 Tensor Processing for Segmentation

The NLST described in Section 2 has shown to be a very suitable way to extract texture information from images. It was employed for texture segmentation in [21] obtaining promising results, but the tensor structure of the texture representation was not exploited. To overcome this limitation, we propose a novel segmentation method for textured images, which, starting from the NLST, applies a tensor adaptive segmentation approach in order to take advantage of the nice properties of the structure tensor as such.

Let us consider an image  $I$ , containing at each pixel, instead of the scalar or vector value, the  $2 \times 2$  nonlinear structure tensor described in Section 2:

$$\mathbf{T} = \begin{pmatrix} \hat{I}_x^2 & I_x \hat{I}_y \\ I_x \hat{I}_y & \hat{I}_y^2 \end{pmatrix} \quad (7)$$

$$\mathbf{T}_C = \sum_i \begin{pmatrix} (\hat{I}_i)_x^2 & (\hat{I}_i)_x (\hat{I}_i)_y \\ (\hat{I}_i)_x (\hat{I}_i)_y & (\hat{I}_i)_y^2 \end{pmatrix} \quad (8)$$

for gray level or colour images, respectively, where by  $\hat{\cdot}$  we denote the nonlinearly diffused components.

For this tensor-valued image, we employ the adaptive segmentation methods based on the Kullback-Leibler and the geodesic distances proposed for the segmentation of DTI images [13, 14, 15] (see Section 3.2).

### 4.1 Advanced Tensor Architectures

The NLST is a very valuable feature for the segmentation of texture images, as will be shown in Section 5. However, when compared with the feature vector  $\mathbf{u} = [\hat{I}_x^2 \quad \hat{I}_y^2 \quad I_x \hat{I}_y \quad \hat{I}]^T$  employed in [21], it is clear that the tensor approach proposed in this paper has the disadvantage of not using any gray information (or colour information, in the case of vector-valued images) at all. Thus, it would be desirable to incorporate this valuable information without losing the nice properties of the NLST. To do so, we propose to use the . . . . . , which, for a scalar image, we define as follows:

$$\mathbf{T}_E = vv^T = \begin{pmatrix} \hat{I}_x^2 & I_x \hat{I}_y & I_x \hat{I} \\ I_x \hat{I}_y & \hat{I}_y^2 & \hat{I}_y \hat{I} \\ I_x \hat{I} & \hat{I}_y \hat{I} & \hat{I}^2 \end{pmatrix} \quad (9)$$

where  $v = [I_x \quad I_y \quad I]^T$ .

With regard to colour images, the extended structure tensor is adapted and becomes  $\mathbf{T}_E = ww^T$ , where  $w = [I'_x \quad I'_y \quad I_R \quad I_G \quad I_B]^T$  and  $I' = \frac{I_R + I_G + I_B}{3}$ .

The . . . . . contains a lot of valuable information for the discrimination between different textures. However, the  $3 \times 3$  tensor ( $5 \times 5$  for colour images) implies that the energy minimization has to be done in a higher

dimensional space, which can be too difficult and result in multiple local minima. To overcome this disadvantage, it would be desirable to reduce the tensor size without losing valuable information. This can be done using principal component analysis (PCA). Using this transformation, it is possible to obtain  $v' = \text{PCA}(v) = [v'_1 \ v'_2]^T$ , which will be afterwards used to construct the structure tensor (see [17] for details):

$$\mathbf{T}_C = v'(v')^T = \begin{pmatrix} (\hat{v'_1})^2 & \hat{v'_1}\hat{v'_2} \\ \hat{v'_1}\hat{v'_2} & (\hat{v'_2})^2 \end{pmatrix} \quad (10)$$

For colour images, the same procedure can be used to reduce the  $5 \times 5$  extended structure tensor to the  $2 \times 2$ .

In some cases, however, valuable information can be lost as the dimension reduction is very marked for the colour case ( $5 \times 5$  to  $2 \times 2$ ). This can be solved by applying a dimensionality reduction to a size that keeps all the eigenvectors responsible for a minimum percentage of the total variance. Using this procedure a structure tensor of variable size is obtained, which is called

## 5 Experimental Results

We first tested the performance of the proposed methods with two synthetic test sets for gray-level and colour images, respectively. Starting from the Brodatz and the CUReT (Columbia Utrecht Reflectance and Texture Database) databases, 100 test images were created for each test set by combining two textures per image.

Different combinations of texture representation schemes and adaptive segmentation methods were tested. First, the vector processing of the NLST [21] was tested and considered as a performance reference, with two slightly different segmentation techniques (see Section 3.1). Next, the gray or colour channels were removed using the earlier vector approach. Afterwards, the tensor segmentation approaches proposed in this work were tested (KL distance and geodesic distance to the Riemannian barycenter, see Section 3.2), combined with the different structure tensors proposed (classical structure tensor, extended structure tensor, compact structure tensor and adaptive compact structure tensor).

The performance of the segmentation was measured in terms of a, defined as the relation between the number of pixels correctly classified and the total number of pixels. Obviously,  $0 \leq S \leq 1$ . In Table 1, we show the median values for all segmentation methods on the gray-value test set. Results for the colour test set are shown in Table 2.

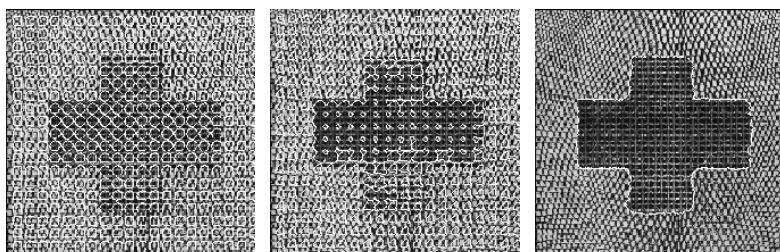
As for the initial contour, small circular contours were placed all over the image. In Figure 1, the evolution of a segmentation process can be seen at different stages. The proposed methods were also tested using real-world images, showing excellent results for gray-level and colour images. Figure 2 shows some results on test images from [21, 5], which prove our method to be fully competitive.

**Table 1.** Results for the different segmentation methods for gray-level images

Texture representation	Segmentation method	Median Value
Feature vector $1 \times 4$	Gaussian full covariance	0.6624
	Gaussian uncorrelated	0.7079
	Gaussian-Parzen	0.7103
Feature vector $1 \times 3$ (no gray information)	Gaussian full covariance	0.6489
	Gaussian uncorrelated	0.5357
Structure tensor $2 \times 2$ (no gray information)	KL distance	0.7040
	Geodesic distance	0.7167
Extended tensor $3 \times 3$	KL distance	0.7405
	Geodesic distance	0.7925
Compact tensor $2 \times 2$	KL distance	0.7800
	Geodesic distance	0.8059

**Table 2.** Results for the different segmentation methods for colour images

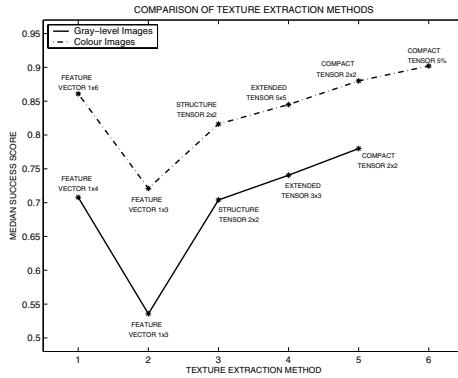
Texture representation	Segmentation method	Median Value
Feature vector $1 \times 6$	Gaussian full covariance	0.7002
	Gaussian uncorrelated	0.8609
Feature vector $1 \times 3$ (no colour information)	Gaussian full covariance	0.6692
	Gaussian uncorrelated	0.7211
Structure tensor $2 \times 2$ (no colour information)	KL distance	0.8162
	Geodesic distance	0.8093
Extended tensor $5 \times 5$	KL distance	0.8459
	Geodesic distance	0.8549
Compact tensor $2 \times 2$	KL distance	0.8807
	Geodesic distance	0.8976
Adaptive Compact tensor 5% of variance	KL distance	0.9023
	Geodesic distance	0.9148

**Fig. 1.** Different samples of the segmentation process for a gray level image belonging to the test set, using the compact and adaptive compact structure tensor (5% of variance), respectively, and KL distance

The results, both for gray-level and colour images, show clearly that the tensor processing of the structure tensor can help improve the accuracy of the segmen-



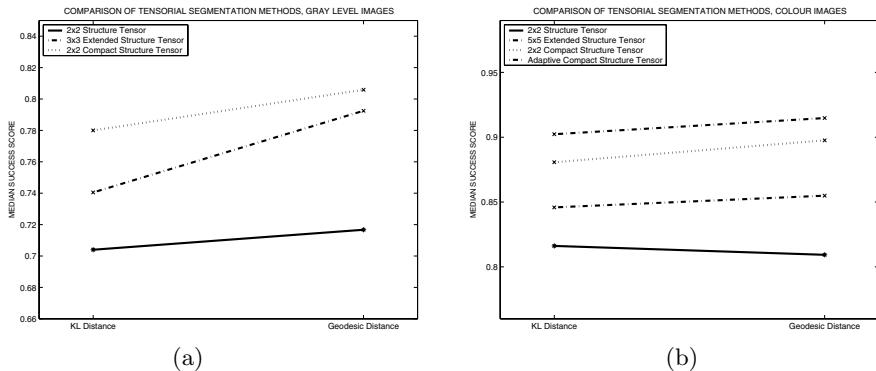
**Fig. 2.** Segmentation results with gray-level and colour real-world images, using the compact structure tensor and KL distance



**Fig. 3.** Graphical comparison of the different texture representation schemes tested, for gray and colour images

tation over the vector processing of that structure tensor. This can be seen in Tables 1 and 2, and even more clearly in Figure 3. Indeed, all the relative performances clearly suggest that the tensor processing is more powerful than the vector counterpart. As with regard to the suitability of the proposed advanced tensor modalities, it can be seen in Figure 3 that, for a fixed tensor segmentation method, the use of the extended structure tensor slightly improves the performance over the classical structure tensor, for gray images. A bigger improvement is obtained for colour images. In both cases, there is a noticeable performance improvement if the compact tensor is used, as it keeps the space dimension low while retaining all the valuable information. For colour images, with the use of the adaptive compact tensor the best results in all can be reached.

Another interesting issue is the choice between the two tensor segmentation methods proposed, which is not so clear. In Figure 4 we show comparisons of the results for both methods, working on the different structure tensor architectures. In general, results favour the use of the geodesic distance to the Riemannian barycenter with respect to the use of the Kullback-Leibler distance. Anyway, the use of the geodesic distance is quite more computationally expensive than the KL option, mainly because the riemannian barycenter has to be computed using an iterative method. This drawback becomes a serious problem for extended tensor architectures, for which the KL distance should be preferred in most cases.



**Fig. 4.** Graphical comparison of the different tensor adaptive segmentation methods tested, for gray (a) and colour images (b)

## 6 Summary

We have presented a NLST based approach for segmenting textured and coloured images. Various tensor field segmentation techniques, recently proposed for DT-MRI, have been employed and tested, showing that the tensor processing of the NLST significantly improves the segmentation performance with respect to more classical approaches based on the vector processing of such tensors. Moreover, it has been shown that the gray or colour information can be incorporated using the extended structure tensor, definitely boosting the segmentation accuracy. One step further was taken with the introduction of the compact structure tensor, which aims at reducing the size of the structure tensor while keeping all the valuable information. An adaptive compact tensor of variable size reaches the maximum refinement and yields results that improve the state of the art.

## Acknowledgments

Our research is partly funded by the Imavis project numbered HPMT-CT-2000-00040 working within the framework of the ..., as well as CICYT TIC 3808-C02-02, FP6-507609. This is gratefully acknowledged.

## References

1. J. Bigun, G. H. Grandlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Trans. on PAMI*, 13(8): 775-790, 1991.
2. J. Bigun, G. H. Grandlund "Optimal orientation detection of linear symmetry," *Proc. 1st IEEE ICCV, London, June 1987*
3. A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. on PAMI*, 12(1): 55-73, 1990.

4. T. Brox and J. Weickert, “Nonlinear matrix diffusion for optic flow estimation,” in *Proc. of the 24th DAGM Symp. , vol. 2449 of LNCS*, Zurich, Switzerland, sep 2002, pp. 446–453.
5. T. Brox, M. Rousson, R. Deriche, and J. Weickert, “Unsupervised segmentation incorporating colour, texture, and motion,” INRIA, Research Rep. 4760, mar 2003.
6. T. Brox, J. Weickert, B. Burgeth, P. Mrázek “Nonlinear structure tensors”, Preprint No. 113, Department of Mathematics, Saarland University, Saarbrücken, Germany. Oct. 2004
7. T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Trans. on IP*, 10(2): 266-277, 2001. *Pattern Recognition* 36(9): 1929-1943. 2003.
8. D. Cremers, F. Tischhauser, J. Weickert and C. Schnorr, “Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional”, *International Journal of Computer Vision* 50(3): 295-313; Dec 2002
9. A. Dervieux and F. Thomasset “A finite element method for the simulation of Rayleigh-Taylor instability” *Lecture Notes in Mathematics*, 771:145–159, 1979
10. A. Dervieux and F. Thomasset “Multifluid incompressible flows by a finite element method” In *International Conference on Numerical Methods in Fluid Dynamics*, 158–163, 1980
11. C. Feddern, J. Weickert, B. Burgeth and M. Welk “Curvature-driven PDE methods for matrix-valued images.”, Technical Report No. 104, Department of Mathematics, Saarland University, Saarbrücken, Germany, Apr. 2004.
12. W. Foerstner and E. Gulch, “A fast operator for detection and precise location of distinct points, corners and centres of circular features”, in: Intercomm. Conf. on Fast Proc. of Photogrammetric. Data, Interlaken, June 1987, pp. 281-305.
13. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras, “Statistics on multivariate normal distributions: A geometric approach and its application to diffusion tensor mri,” INRIA, Research Report 5242, Jun 2004.
14. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras, “Toward segmentation of 3D probability density fields by surface evolution: Application to diffusion mri,” INRIA, Research Rep. 5243, June 2004.
15. C. Lenglet, M. Rousson, and R. Deriche, “Segmentation of 3d probability density fields by surface evolution: Application to diffusion mri,” in *Proc. of the MICCAI*, Saint Malo, France, Sep. 2004.
16. M. E. Leventon, O. Faugeras, W. E. L. Grimson, and W. M. W. III, “Level set based segmentation with intensity and curvature priors,” in *Proc. of the IEEE Workshop on MMBIA*, Hilton Head, SC, USAs, jun 2000, pp. 4–11.
17. R. de Luis-Garcia, R. Deriche, C. Lenglet, and M. Rousson, “Tensor processing for texture and colour segmentation,” INRIA, Research. Rep., in press.
18. R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front propagation: A level set approach,” *IEEE Trans. on PAMI*, 17(2): 158-175, feb 1995.
19. N. Paragios and R. Deriche, “Geodesic active regions and level set methods for supervised texture segmentation,” *The International Journal of Computer Vision*, 46(3): 223-247, 2002.
20. N. Paragios and R. Deriche, “Geodesic active regions: A new framework to deal with frame partition problems in computer vision,” *Journal of Visual Communication and Image Representation*, vol. 13, pp. 249–268, 2002.
21. M. Rousson, T. Brox, and R. Deriche, “Active unsupervised texture segmentation on a diffusion based feature space,” in *Proc. of CVPR*, Madison, Wisconsin, USA, jun 2003.

22. M. Rousson, C. Lenglet, and R. Deriche, “Level set and region based surface propagation for diffusion tensor mri segmentation,” in *Proc. of the Computer Vision Approaches to Medical Image Analysis Workshop*, Prague, May 2004.
23. S. Osher and J. A. Sethian, “Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation,” *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
24. Z. Wang and B. C. Vemuri, “An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation,” in *Proc. of the IEEE CVPR*, Washington DC, USA, 2004, pp. 228–233.
25. Z. Wang and B. C. Vemuri, “Tensor field segmentation using region based active contour model,” in *Proc. of the ECCV*, Prague, Czech Republic, may 2004.

# Cerebrovascular Segmentation by Accurate Probabilistic Modeling of TOF-MRA Images

Ayman El-Baz<sup>1</sup>, Aly Farag<sup>1</sup>, and Georgy Gimelfarb<sup>2</sup>

<sup>1</sup> Computer Vision and Image Processing Laboratory,  
University of Louisville, Louisville, KY 40292, USA  
[{farag, elbaz}@cvip.Louisville.edu](mailto:{farag, elbaz}@cvip.Louisville.edu)  
<http://www.cvip.louisville.edu>

<sup>2</sup> Department of Computer Science, Tamaki Campus,  
University of Auckland, Auckland, New Zealand

**Abstract.** We present a fast algorithm for automatic extraction of a 3D cerebrovascular system from time-of-flight (TOF) magnetic resonance angiography (MRA) data. Blood vessels are separated from background tissues (fat, bones, or grey and white brain matter) by voxel-wise classification based on precise approximation of a multi-modal empirical marginal intensity distribution of the TOF-MRA data. The approximation involves a linear combination of discrete Gaussians (LCDG) with alternating signs, and we modify the conventional Expectation-Maximization (EM) algorithm to deal with the LCDG. To validate the accuracy of our algorithm, a special 3D geometrical phantom motivated by statistical analysis of the MRA-TOF data is designed. Experiments with both the phantom and 50 real data sets confirm high accuracy of the proposed approach.

## 1 Introduction

Accurate cerebrovascular segmentation is of prime importance for early diagnostics and timely endovascular treatment. Unless detected at early stage, serious vascular diseases like carotid stenosis, aneurysm, and vascular malformation may cause not only severe headaches but also a brain stroke or a life-threatening coma [1]. Non-invasive MRA is a valuable tool in preoperative evaluation of suspected intracranial vascular diseases. Three commonly used MRA techniques are TOF-MRA, phase contrast angiography (PCA), and contrast enhanced MRA (CE-MRA). Both TOF-MRA and PCA use flowing blood as an inherent contrast medium, while for CE-MRA a contrasting substance is injected into the circulatory system. PCA exploits phase changes of transverse magnetization when flowing spins move through a magnetic field gradient. This provides good background signal suppression and can quantify flow velocity vectors for each voxel. TOF-MRA relying on amplitude differences in longitudinal magnetization between flowing static spins is less quantitative, however it is fast and provides high contrast images. The fact that it is widely used in clinical practice is the main motivation behind our work.

A variety of today's most popular techniques for segmenting blood vessels from TOF-MRA data can be roughly classified into deformable and statistical models. The former methods iteratively deform an initial boundary surface of blood vessels in order to optimize an energy function which depends on image gradient information and surface smoothness [2]. Topologically adaptable surfaces make classical deformable models more efficient for segmenting intracranial vasculature [3]. Geodesic active contours implemented with level set techniques offer flexible topological adaptability to segment MRA images [4] including more efficient adaptation to local geometric structures represented e.g. by tensor eigenvalues [5]. Fast segmentation of blood vessel surfaces is obtained by inflating a 3D balloon with fast marching methods [6]. Two-step segmentation of a 3D vascular tree from CTA data sets in [7] is first carried out locally in a small volume of interest. Then a global topology is estimated to initialize a new volume of interest. A multi-scale geometrical flow is proposed in [8] to segment vascular tree from MRI images. These methods produce quite good experimental results but have a common drawback. They need manual initialization and are slow comparing to the statistical approaches.

The statistical approach extracts the vascular tree automatically, but its accuracy depends on underlying probability data models. The TOF-MRA image is multi-modal in that signals in each region-of-interest (e.g. blood vessels, brain tissues, etc) are associated with a particular mode of the total marginal probability distribution of signals. To the best of our knowledge, up-to-now there exists only one adaptive statistical approach for extracting blood vessels from the TOF-MRA data proposed by Wilson and Noble [9]. They model the marginal data distribution with a mixture of two Gaussian and one uniform components for the stationary CSF, brain tissues, and arteries, respectively. To identify the mixture (i.e. estimate all its parameters), they use a conventional EM algorithm<sup>3</sup>. Furthermore, a region-based deformable contour for segmenting tubular structures was derived in [10] by combining signal statistics and shape information.

In this paper we precisely identify rather than pre-select the probability models of each region-of-interest for a given TOF-MRA image. The whole empirical gray level distribution is approximated first with an LCDG. Then this latter is split into individual LCDGs for the regions, or modes. Experiments show that such more accurate region models result in significantly better segmentation. It should be noted that probability distributions comprise only a proper subset of the LCDGs which may be negative for some or all  $q \in \mathbf{Q}$ . For simplicity, we do not restrict the identification to only that subset. As will soon become clear, the restrictions may be ignored due to very close approximation provided by the LCDGs.

---

<sup>3</sup> It was called a "modified EM" in [9] after a marginal gray level distribution replaced individual pixel-wise gray levels in their initial EM algorithm. But such a modification simply returns to what was in common use for decades, while the individual pixels emerged only as an unduly verbatim replica of a general EM framework.

## 2 LCDG-Model of a Multi-modal TOF-MRA Image

Let  $q; q \in \mathbf{Q} = \{0, 1, \dots, Q - 1\}$ , denote the  $Q$ -ary gray level. The discrete Gaussian (DG) is defined as the probability distribution  $\Psi_\theta = (\psi(q|\theta) : q \in \mathbf{Q})$  on  $\mathbf{Q}$  such that  $\psi(q|\theta) = \Phi_\theta(q + 0.5) - \Phi_\theta(q - 0.5)$  for  $q = 1, \dots, Q - 2$ ,  $\psi(0|\theta) = \Phi_\theta(0.5)$ ,  $\psi(Q - 1|\theta) = 1 - \Phi_\theta(Q - 1.5)$  where  $\Phi_\theta(q)$  is the cumulative Gaussian probability function with a shorthand notation  $\theta = (\mu, \sigma^2)$  for its mean,  $\mu$ , and variance,  $\sigma^2$ .

We assume the number  $K$  of dominant modes, i.e. regions, objects, or classes of interest in a given TOF-MRA image, is already known. In contrast to a conventional mixture of Gaussians and/or other simple distributions, one per region, we closely approximate the empirical gray level distribution for a TOF-MRA image with an LCDG having  $C_p$  positive and  $C_n$  negative components such that  $C_p \geq K$ :

$$p_{\mathbf{w}, \Theta}(q) = \sum_{r=1}^{C_p} w_{p,r} \psi(q|\theta_{p,r}) - \sum_{l=1}^{C_n} w_{n,l} \psi(q|\theta_{n,l}) \quad (1)$$

under the obvious restrictions on the weights  $\mathbf{w} = [w_{p,.}, w_{n,.}]$ : all the weights are non-negative and

$$\sum_{r=1}^{C_p} w_{p,r} - \sum_{l=1}^{C_n} w_{n,l} = 1 \quad (2)$$

To identify the LCDG-model including the numbers of its positive and negative components, we modify the conventional Expectation-Maximization (EM) algorithm to deal with the LCDG.

First the numbers  $C_p - K$ ,  $C_n$  and parameters  $\mathbf{w}$ ,  $\Theta$  (weights, means, and variances) of the positive and negative DG components are estimated with a sequential EM-based initializing algorithm. The goal is to produce a close initial LCDG-approximation of the empirical distribution. Then under the fixed  $C_p$  and  $C_n$ , all other model parameters are refined with an EM algorithm that modifies the conventional one in [11] to account for the components with alternating signs.

### 2.1 Sequential EM-Based Initialization

Sequential EM-based initialization forms an LCDG-approximation of a given empirical marginal gray level distribution using the conventional EM-algorithm [11] adapted to the DGs. At the first stage, the empirical distribution is represented with a mixture of  $K$  positive DGs, each dominant mode being roughly approximated with a single DG. At the second stage, deviations of the empirical distribution from the dominant  $K$ -component mixture are modeled with other, “subordinate” components of the LCDG. The resulting initial LCDG has  $K$  dominant weights, say,  $w_{p,1}, \dots, w_{p,K}$  such that  $\sum_{r=1}^K w_{p,r} = 1$ , and a number of subordinate weights of smaller values such that  $\sum_{r=K+1}^{C_p} w_{p,r} - \sum_{l=1}^{C_n} w_{n,l} = 0$ .

The subordinate components are determined as follows. The positive and negative deviations of the empirical distribution from the dominant mixture are

separated and scaled up to form two new “empirical distributions”. The same conventional EM algorithm is iteratively exploited to find the subordinate mixtures of positive or negative DGs that approximate best the scaled-up positive or negative deviations, respectively. The sizes  $C_p - K$  and  $C_n$  of these mixtures are found by sequential minimization of the total absolute error between each scaled-up deviation and its mixture model by the number of the components. Then the obtained positive and negative subordinate models are scaled down and then added to the dominant mixture yielding the initial LCDG model of the size  $C = C_p + C_n$ .

## 2.2 Modified EM Algorithm for LCDG

Modified EM algorithm for LCDG maximizes the log-likelihood of the empirical data by the model parameters assuming statistically independent signals:

$$L(\mathbf{w}, \Theta) = \sum_{q \in \mathbf{Q}} f(q) \log p_{\mathbf{w}, \Theta}(q) \quad (3)$$

A local maximum of the log-likelihood in Eq. (3) is given with the EM process extending the one in [11] onto alternating signs of the components. Let  $p_{\mathbf{w}, \Theta}^{[m]}(q) = \sum_{r=1}^{C_p} w_{p,r}^{[m]} \psi(q|\theta_{p,r}^{[m]}) - \sum_{l=1}^{C_n} w_{n,l}^{[m]} \psi(q|\theta_{n,l}^{[m]})$  denote the current LCDG at iteration  $m$ . Relative contributions of each signal  $q \in \mathbf{Q}$  to each positive and negative DG at iteration  $m$  are specified by the respective conditional weights

$$\pi_p^{[m]}(r|q) = \frac{w_{p,r}^{[m]} \psi(q|\theta_{p,r}^{[m]})}{p_{\mathbf{w}, \Theta}^{[m]}(q)}; \quad \pi_n^{[m]}(l|q) = \frac{w_{n,l}^{[m]} \psi(q|\theta_{n,l}^{[m]})}{p_{\mathbf{w}, \Theta}^{[m]}(q)} \quad (4)$$

such that the following constraints hold:

$$\sum_{r=1}^{C_p} \pi_p^{[m]}(r|q) - \sum_{l=1}^{C_n} \pi_n^{[m]}(l|q) = 1; \quad q = 0, \dots, Q-1 \quad (5)$$

The following two steps iterate until the log-likelihood changes become small:

- E– step<sup>[m+1]</sup>:** Find the weights of Eq. (4) under the fixed parameters  $\mathbf{w}^{[m]}$ ,  $\Theta^{[m]}$  from the previous iteration  $m$ , and
- M– step<sup>[m+1]</sup>:** Find conditional MLEs  $\mathbf{w}^{[m+1]}$ ,  $\Theta^{[m+1]}$  by maximizing  $L(\mathbf{w}, \Theta)$  under the fixed weights of Eq. (4).

Considerations closely similar to those in [11] show this process converges to a local log-likelihood maximum. Let the log-likelihood of Eq. (3) be rewritten in the equivalent form with the constraints of Eq. (5) as unit factors:

$$L(\mathbf{w}^{[m]}, \Theta^{[m]}) = \sum_{q=0}^Q f(q) \left[ \sum_{r=1}^{C_p} \pi_p^{[m]}(r|q) \log p^{[m]}(q) - \sum_{l=1}^{C_n} \pi_n^{[m]}(l|q) \log p^{[m]}(q) \right] \quad (6)$$

Let the terms  $\log p^{[m]}(q)$  in the first and second brackets be replaced with the equal terms  $\log w_{p,r}^{[m]} + \log \psi(q|\theta_{p,r}^{[m]}) - \log \pi_p^{[m]}(r|q)$  and  $\log w_{n,l}^{[m]} + \log \psi(q|\theta_{n,l}^{[m]}) - \log \pi_n^{[m]}(l|q)$ , respectively, which follow from Eq. (4). At the E-step, the conditional Lagrange maximization of the log-likelihood of Eq. (6) under the  $Q$  restrictions of Eq. (5) results just in the weights  $\pi_p^{[m+1]}(r|q)$  and  $\pi_n^{[m+1]}(l|q)$  of Eq. (4) for all  $r = 1, \dots, C_p$ ;  $l = 1, \dots, C_n$  and  $q \in \mathbf{Q}$ . At the M-step, the DG weights  $w_{p,r}^{[m+1]} = \sum_{q \in \mathbf{Q}} f(q) \pi_p^{[m+1]}(r|q)$  and  $w_{n,l}^{[m+1]} = \sum_{q \in \mathbf{Q}} f(q) \pi_n^{[m+1]}(l|q)$  follow from the conditional Lagrange maximization of the log-likelihood in Eq. (6) under the restriction of Eq. (2) and the fixed conditional weights of Eq. (4). Under these latter, the conventional MLEs of the parameters of each DG stem from maximizing the log-likelihood after each difference of the cumulative Gaussians is replaced with its close approximation with the Gaussian density (below “c” stands for “p” or “n”, respectively):

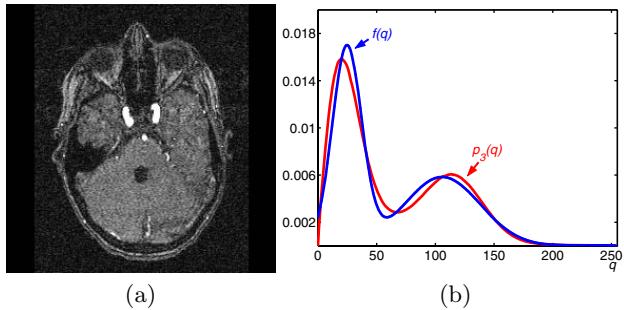
$$\begin{aligned}\mu_{c,r}^{[m+1]} &= \frac{1}{w_{c,r}^{[m+1]}} \sum_{q \in \mathbf{Q}} q \cdot f(q) \pi_c^{[m+1]}(r|q) \\ (\sigma_{c,r}^{[m+1]})^2 &= \frac{1}{w_{c,r}^{[m+1]}} \sum_{q \in \mathbf{Q}} \left( q - \mu_{c,r}^{[m+1]} \right)^2 \cdot f(q) \pi_c^{[m+1]}(r|q)\end{aligned}$$

This modified EM-algorithm is valid until the weights  $\mathbf{w}$  are strictly positive. The iterations should be terminated when the log-likelihood of Eq. (3) does not change or begins to decrease due to accumulation of rounding errors.

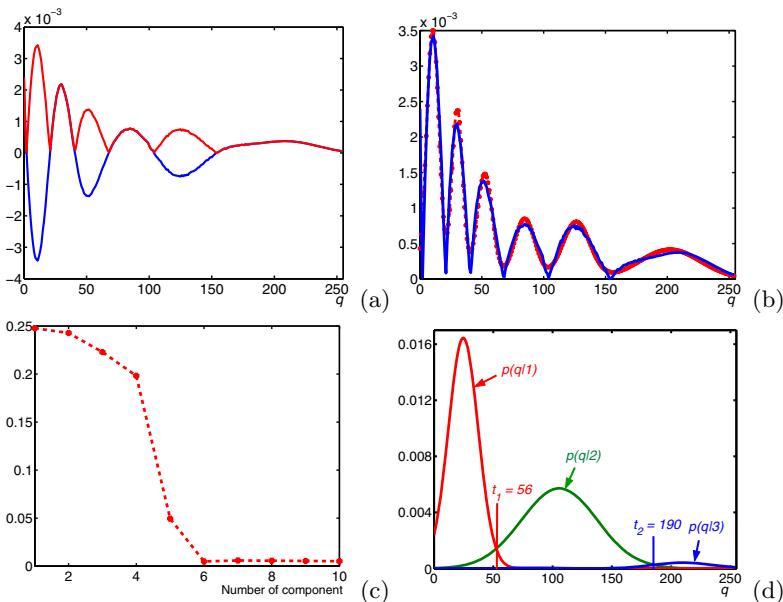
The final mixed LCDG-model  $p_C(q)$  is partitioned into the  $K$  LCDG-submodels  $P_{[k]} = [p(q|k) : q \in \mathbf{Q}]$ , one per class  $k = 1, \dots, K$ , by associating the subordinate DGs with the dominant terms so that the misclassification rate is minimal.

### 3 Experimental Results

Experiments were conducted with the TOF-MRA images acquired with the Picker 1.5T Edge MRI scanner having spatial resolution of  $0.43 \times 0.43 \times 1.0$  mm. The size of each 3D data set is  $512 \times 512 \times 93$ . The TOF-MRA images contain three classes ( $K = 3$ ), namely, darker bones and fat, brain tissues, and brighter blood vessels. A typical TOF-MRA slice, its empirical marginal gray level distribution  $f(q)$ , and the initial 3-component Gaussian dominant mixture  $p_3(q)$  are shown in Fig. 1. Figure 2 illustrates basic stages of our sequential EM-based initialization by showing the scaled-up alternating and absolute deviations  $f(q) - p_3(q)$ , the best mixture model estimated for the absolute deviations (these six Gaussian components give the minimum approximation error), and the initial LCDG-models for each class. The scaling makes the sums of the positive or absolute negative deviations for  $q = 0, \dots, Q - 1$  equal to one. Figure 3 presents the final LCDG-model after refining the initial one with the modified EM-algorithm and shows successive changes of the log-likelihood at the refinement iterations. The final LCDG-models of each class are obtained with the best separation thresholds  $t_1 = 57$  and  $t_2 = 192$ . First nine refining iterations increase the log-likelihood from  $-5.7$  to  $-5.2$ .

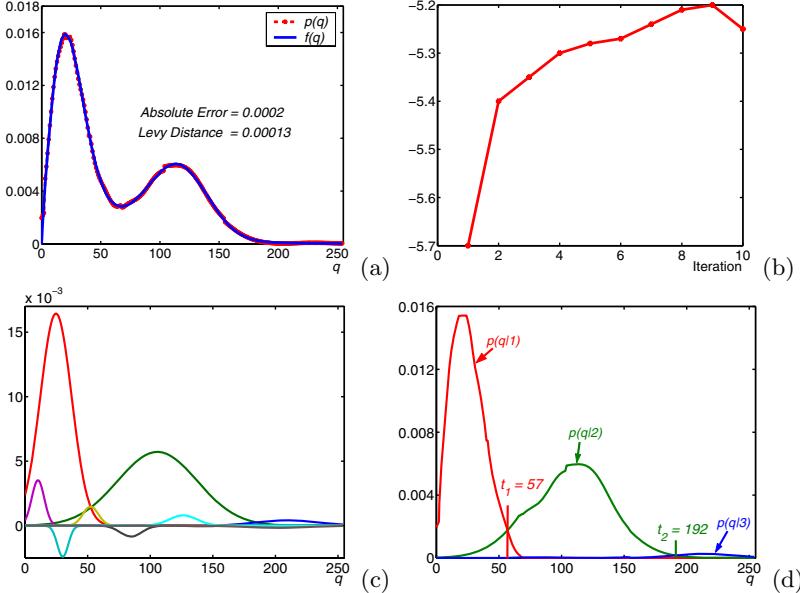


**Fig. 1.** Typical TOF-MRA scan slice (a) and deviations between the empirical distribution  $f(q)$  and the dominant 3-component mixture  $p_3(q)$  (b)

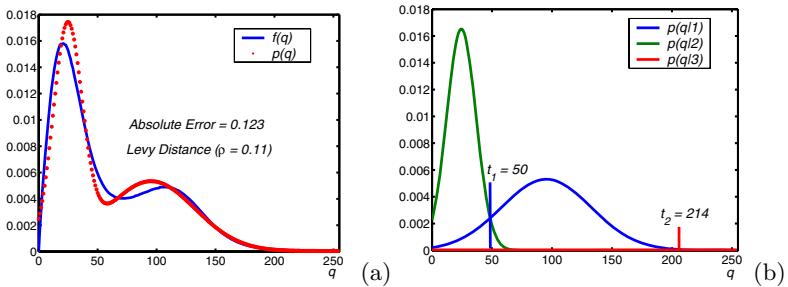


**Fig. 2.** Deviations and absolute deviations between  $f(q)$  and  $p_3(q)$  (a), the mixture model (b) of the absolute deviations in (a), the absolute error (c) as a function of the number of Gaussians approximating the scaled-up absolute deviations in (a), and the initial estimated LCDG-models for each class (d)

To highlight the advantages of our approach over the existing one, Fig. 4 shows results obtained with the model of Wilson and Noble [9]. To measure the estimation quality, we use the Levy distance between two distributions [12] and the absolute error. The Levy distance between the empirical distribution and its estimated model is 0.11 and 0.00013 and the absolute errors are 0.123 and 0.0002 for the Wilson-Noble's and our approach, respectively. The larger Levy distance and absolute error indicate the notably worse approximation which strongly



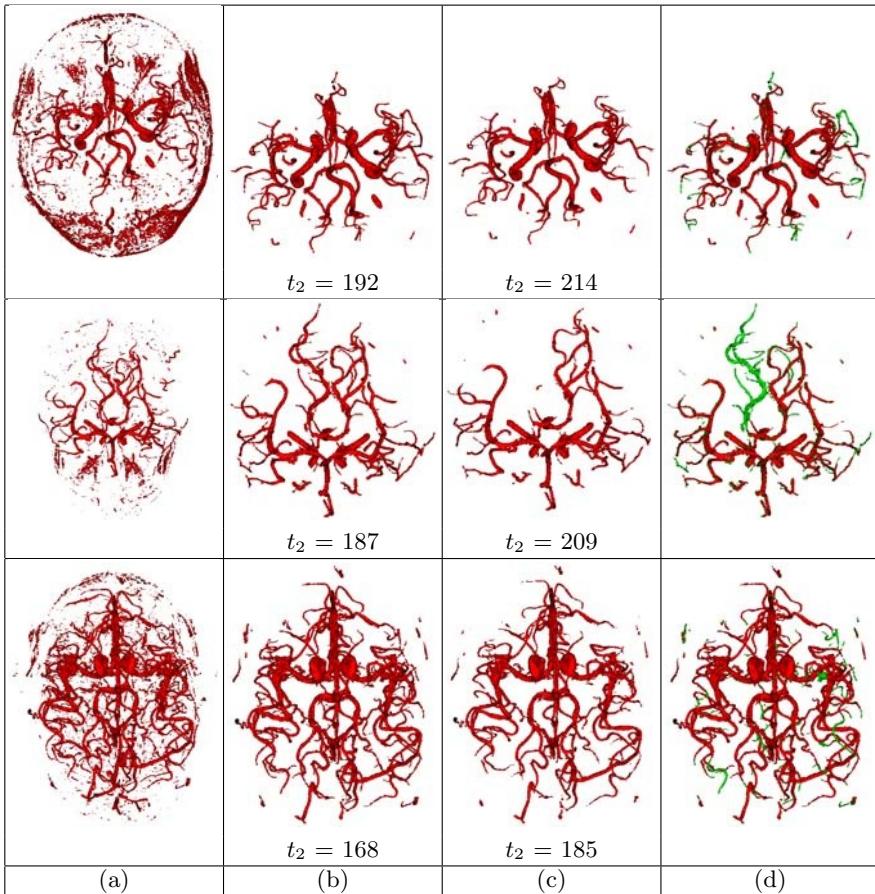
**Fig. 3.** Final 3-class LCDG-model overlaying the empirical density (a), the log-likelihood dynamics (b) for the refining EM-iterations, the refined model components (c), and the class LCDG-models (d)



**Fig. 4.** Wilson-Noble's model [9]: the estimated distribution (a) and the class models (b)

affects the accuracy of separating the blood vessels from the background. Because of a typically higher separation threshold, e.g.  $t_2 = 214$  versus our  $t_2 = 192$  in this particular example, the Wilson-Noble's approach misses some blood vessels, as shown in Fig. 5.

Both the approaches have been compared on 50 data sets. Results of the three tests are depicted in Fig. 5. As the first column, (a), suggests, TOF-MRA is sensitive to tissues like subcutaneous fat with a short T1 response that may obscure the blood vessels in the segmented volume. To eliminate them, the volume is processed with an automatic connectivity filter which selects the largest

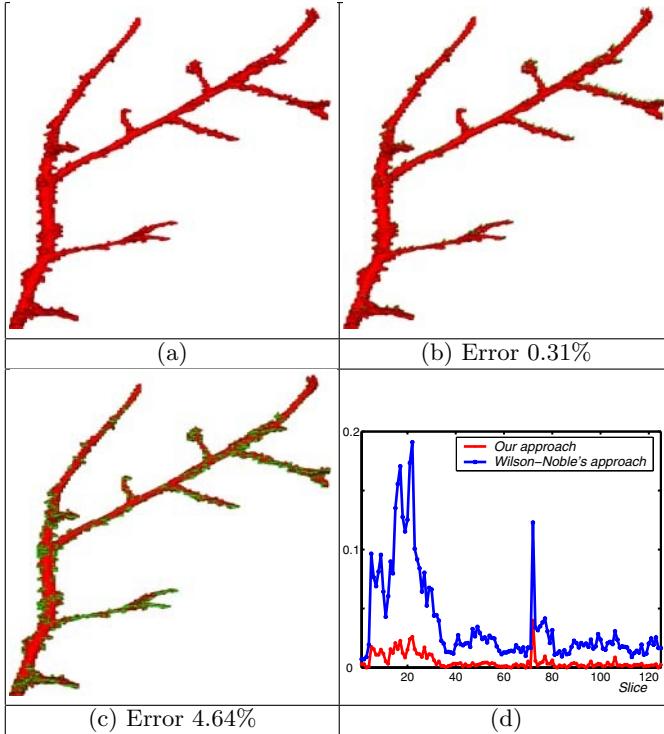


**Fig. 5.** Each row relates to one patient: our segmentation before (a) and after (b) nose and small fat voxels are eliminated with the connectivity filter, the Wilson-Noble's segmentation (c) after the connectivity filter, and the differences between (b) and (c); the green voxels are missed by the Wilson-Noble's approach and the red ones are detected by the both approaches

connected tree structures using a 3D volume growing algorithm [13]. The results after applying such a filter to our and Wilson-Noble's segmentation in Fig. 5 show that the latter approach fails to detect sizeable fractions of the vascular trees which are validated by the expert (radiologist) that the green parts which are detected by our approaches follow the topology of the brain vascular tree.

## 4 Validation

It is very difficult to get accurate manually segmented complete vasculatures to validate our algorithm. Thus to evaluate its performance, we have created a



**Fig. 6.** The 3D geometrical phantom (a), our (b) and Wilson-Noble’s (c) segmentation, and total errors per each phantom’s slice for both the approaches (d)

wooden phantom shown in Fig. 6(a) with topology similar to the blood vessels. Furthermore, the phantom mimics bifurcations, zero and high curvature that exist in any vasculature system, and it has a varying radius to simulate both large and small blood vessels. The phantom was scanned by CT and then manually segmented to obtain the ground truth. The blood vessel and non-vessel signals for the phantom are generated according to the distribution  $p(q|3)$  and  $p(q|1)$ ,  $p(q|2)$ , respectively, in Fig. 3(d) using the inverse mapping methods. The resulting phantom’s histogram was similar to that in Fig. 3(a).

Let the total segmentation error be a percentage of erroneous voxels with respect to the overall number of voxels in the manually segmented 3D phantom. Figure 6 shows our approach is 15 times more accurate than the Wilson-Noble’s one (the total errors 0.31% and 4.64%, respectively). The error constituents per each 2D slice for both the approaches are also plotted in Fig. 6.

## 5 Conclusions

We presented a new statistical approach to find blood vessels in multi-modal TOF-MRA images. The LCDG-model accurately approximates the empirical

marginal gray level distribution yielding the high quality segmentation. The accuracy of our approach is validated using a specially designed 3D geometrical phantom.

The proposed techniques include a modified EM for refining the model parameters and an accurate sequential EM-based initialization. The accurate initial LCDG-model ensures fast convergence of the model refinement with the modified EM algorithm. Our present implementation on C++ programming language using a single 2.4 GHZ Pentium 4 CPU with 512 MB RAM takes about 49 s for 93 TOF-MRA slices of size 512x512 pixels each.

The proposed LCDG-model is not limited only for TOF-MRA but also is suitable for segmenting PC-MRA and CTA medical images. The latter were not included in the paper because of the space limitations, but, the algorithm's code, sample data and segmentation results for the TOF-MRA, PC-MRA, and CTA images will be provided in our web site.

## References

1. Health Resources. Patient Resources, *Disorder of the Month. Cerebrovascular Disease. Neurosurgery: On-Call* [serial online], July 1999.
2. V. Caselles, R. Kimmel, and G. Sapiro., “Geodesic active contours,” *Int. J. Computer Vision*, vol. 22, pp. 61–79, 1997.
3. T. McInerney and D. Terzopoulos, “Medical image segmentation using topologically adaptable surface”, *Proc. CVRMED-MRCAS'97*, pp. 23–32, 1997
4. L. M. Lorigo, O. D. Faugeras, W. E. L. Grimson, and R. Keriven, “Curves: Curve evolution for vessel segmentation”, *Medical Image Analysis*, vol. 5, pp. 195–206, 2001.
5. O. Wink, W. J. Niessen, and M. A. Viergever, “Fast delineation and visualization of vessels in 3-D angiographic images,” *IEEE Trans. Med. Imaging*, vol. 19, pp. 337–346, 2000.
6. T. Deschamps and L. D. Cohen, “Fast extraction of tubular and tree 3D surfaces with front propagation methods”, *Proc. 16<sup>th</sup> ICPR*, pp. 731–734, 2002.
7. R. Manniesing and W. Niessen, “Local speed functions in level set based vessel segmentation”, *Proc. MICCAI'04*, pp. 475–482, 2004.
8. M. Descotesux, L. Collins, and K. Siddiqi, “Geometric flows for segmenting vasculature in MRI: Theory and validation”, *Proc. MICCAI'04*, pp. 500–507, 2004.
9. D. L. Wilson and J. A. Noble, “An adaptive segmentation algorithm for time-of-flight MRA data”, *IEEE Trans. Med. Imaging*, vol. 18, pp. 938–945, 1999.
10. D. Nain, A. Yezzi, and G. Turk, “Vessels segmentation using a shape driven flow”, *Proc. MICCAI'04*, pp. 51–59, 2004.
11. M. I. Schlesinger and V. Hlavac, *Ten Lectures on Statistical and Structural Pattern Recognition*, Kluwer Academic, Dordrecht, 2002.
12. J. W. Lamperti, *Probability*, J. Wiley & Sons, New York, 1996.
13. Mohamed Sabry, Charles B. Sites, Aly A. Farag, Stephen Hushek, and Thomas Moriarty, “A Fast Automatic Method for 3D Volume Segmentation of the Human Cerebrovascular,” Proc. of the 13th International Conf. on Computer Assisted Radiology and Surgery, (CARS'02), Paris, France, pp. 382-387, 26-29 June, 2002.

# MGRF Controlled Stochastic Deformable Model

Ayman El-Baz<sup>1</sup>, Aly Farag<sup>1</sup>, and Georgy Gimelfarb<sup>2</sup>

<sup>1</sup> Computer Vision and Image Processing Laboratory,  
University of Louisville, Louisville, KY 40292, USA  
[{farag, elbaz}@cvip.Louisville.edu](mailto:{farag, elbaz}@cvip.Louisville.edu)  
<http://www.cvip.louisville.edu>

<sup>2</sup> Department of Computer Science, Tamaki Campus,  
University of Auckland, Auckland, New Zealand

**Abstract.** Deformable or active contour, and surface models are powerful image segmentation techniques. We introduce a novel fast and robust bi-directional parametric deformable model which is able to segment regions of intricate shape in multi-modal greyscale images. The power of the algorithm in terms of computation time and robustness is owing to the use of joint probabilities of the signals and region labels in individual points as external forces guiding the model evolution. These joint probabilities are derived from a Markov–Gibbs random field (MGRF) image model considering an image as a sample of two interrelated spatial stochastic processes. The low level process with conditionally independent and arbitrarily distributed signals relates to the observed image whereas its hidden map of regions is represented with the high level MGRF of interdependent region labels. Marginal probability distributions of signals in each region are recovered from a mixed empirical signal distribution over the whole image. In so doing, each marginal is approximated with a linear combination of Gaussians (LCG) having both positive and negative components. The LCG parameters are estimated using our previously proposed modification of the EM algorithm, and the high-level Gibbs potentials are computed analytically. Comparative experiments show that the proposed model outlines complicated boundaries of different modal objects much more accurately than other known counterparts.

## 1 Introduction

Deformable or active model, is a curve in a 2D digital image or a surface in a 3D image that evolves to outline a desired object. The evolution is controlled by internal and external forces combined, together with user defined constraints, into internal and external energy terms, respectively. Introduced first by Kass et al. [1], the models gave rise to one of the most dynamic and successful research areas in edge detection, image segmentation, shape modeling, and visual tracking. By representation and implementation, deformable models are broadly categorized into parametric (e.g. [1, 2]) and geometric (e.g. [3, 4]) classes. In this

paper, we focus on parametric deformable models that form a parametric curve and move it toward an object boundary.

Performance of the deformable model depends on proper initialization, efficiency of the energy minimization process, and adequate selection of the force functions and energy functions. The original iterative minimization in [1] is based on a closed-form solution of Eulerian differential equations specifying the desired minimum. But it turns out to be unstable and usually gets trapped into local minima. Amini et al. [5] point to shortcomings of this minimization process in [1] and improve it by representing a contour as a linked chain of control points and minimizing its total energy by discrete dynamic programming. This approach allows for rigid constraints on the energy function that make the minimization more stable. But still its control parameters must be adjusted very carefully, and the process remains too time consuming. Time complexity of a more advanced greedy algorithm proposed by Williams and Shah [6] is linear with respect to both the number of control points and the neighbors of each point which are taken into account for energy minimization. It is much more stable and is simultaneously more than an order of magnitude faster than the previous techniques.

To more closely approach a complicated boundary with concavities, Grzeszczuk and Levin [7] control snake evolution by simulated annealing. In principle the latter eventually reaches the global minimum of energy and can escape local traps. But in practice, as emphasized in [8], simulated annealing typically stops very far from the global minimum. Alternative minimum-cut graph algorithms in [9] guarantee a close approximation of geodesic contours in 2D (or minimal surface in 3D) images having the minimum global energy under an arbitrary Riemannian metric of a set of boundary conditions. But both these minimization processes are extremely slow. A faster precise boundary approximation proposed by Xu and Prince [2] introduces a gradient vector flow (GVF) as a new external force. Due to its larger capture range, the GVF allows a contour to move into the boundary concavities. A bi-directional deformable model in [4] combines the geodesic contour and the GVF.

In spite of good segmentation results for objects of relatively simple shapes, the above conventional deformable models have serious drawbacks. Most of them are slow compared to other segmentation techniques, and the model evolution frequently stops well before approaching a complicated object boundary with concavities. Also, to initialize the model, typically a closed curve has to be interactively drawn near the desired boundary, and this manual step hinders their use in many applications.

In this paper we propose joint probabilities of signals and region labels in individual image points as a new class of external forces to guide the model evolution. This class overcomes the above problems in the case of multi-modal images where each object of interest relates to a separate mode of the empirical marginal signal distribution. We call a model with the probabilistic external forces a stochastic deformable model. Its advantages over more conventional models are in the automated initialization, insensitivity to the initialization, and ability to follow complex shapes with concavities.

## 2 Parametric Deformable 2D Contours

A conventional parametric deformable 2D model, or snake, is a curve  $\Phi = (\phi(\tau) = (u(\tau), v(\tau)); \tau \in T)$  in planar Cartesian co-ordinates  $(u, v)$  where  $\tau$  is the continuous or discrete index of a contour point and  $T$  is the index range. The deformable model moves through the spatial image domain to minimize the total energy

$$E = E_{\text{int}} + E_{\text{ext}} = \int_{\tau \in T} \xi_{\text{int}}(\phi(\tau)) + \xi_{\text{ext}}(\phi(\tau)) d\tau \quad (1)$$

where  $\xi_{\text{int}}(\phi(\tau))$  and  $\xi_{\text{ext}}(\phi(\tau))$  denote the internal and external forces, respectively, that control the point-wise model movements. The total energy is the sum of two terms, the internal energy keeping the deformable model as a single unit and the external one attracting the model to the region boundary. The internal force is typically defined as  $\xi_{\text{int}}(\phi(\tau)) = \alpha|\phi'(\tau)|^2 + \beta|\phi''(\tau)|^2$  where weights  $\alpha$  and  $\beta$  control the curve's tension and rigidity, respectively, and  $\phi'(\tau)$  and  $\phi''(\tau)$  are the first and second derivatives of  $\phi(\tau)$  with respect to  $\tau$ .

Typical external forces designed in [1] to lead an active contour toward step edges in a grayscale image  $\mathbf{Y}$  are:

$$\begin{aligned} \xi_{\text{ext}}(\phi(\tau)) = & -|\nabla \mathbf{Y}(\phi(\tau))|^2 \text{ or} \\ & -|\nabla[G(\phi(\tau)) * \mathbf{Y}(\phi(\tau))]|^2 \end{aligned} \quad (2)$$

where  $G(\dots)$  is a 2D Gaussian kernel and  $\nabla$  denotes the gradient operator. But both these and other traditional external forces (e.g. based on lines, edges, or the GVF) fail to make the contour to closely approach an intricate boundary with concavities. Moreover, due to high computational complexity the deformable models with most of such external energies are slow compared to the other segmentation techniques.

## 3 Stochastic Deformable 2D Contour

The above drawbacks are overcome to a large extent when joint probabilities of image signals and region labels in individual points along the deformable model are used as new external forces. The probabilities are easily derived from a simple MGRF model of multi-modal greyscale images. The model merges two interrelated spatial stochastic processes. The low level process is a conditionally independent random field of image signals (gray levels) with arbitrary probability distributions of signals. The distributions differ for different regions but are the same for each pixel in the region. By assumption, their mixed distribution for the whole image is multi-modal, each mode corresponding to one of the regions. This process relates to the observed image whereas a hidden map of regions is represented with the high level MGRF of interdependent region labels. The interdependence is restricted to only pairs of labels in the nearest 8-neighborhood of each

pixel. By symmetry considerations, Gibbs potentials are the same for all pairs and regions, depending only on whether the labels are equal or not in the pair, and thus have only two values:  $\gamma$  for the equal and  $-\gamma$  for unequal pairs of labels.

To compute the forces, the low-level model is identified for a given image  $\mathbf{Y}$  by the LCG-approximation of the conditional marginal signal distributions in each region with the modified EM-algorithm proposed in [10]. The estimated distributions allow us to get a region map  $\mathbf{X}$  for the image  $\mathbf{Y}$  by classifying the individual pixels. Then the high-level model is identified for the region map  $\mathbf{X}$  using the analytic potential estimate derived in accordance with [11]. In our case,  $\gamma = \frac{K^2}{(K-1)} (f_{eq}(\mathbf{X}) - \frac{1}{K})$  where  $K$  is the number of modes, or regions in the image  $\mathbf{Y}$  and  $f_{eq}(\mathbf{X})$  denotes the empirical frequency of the equal labels in the pairs of the nearest 8-neighboring pixels in the map  $\mathbf{X}$ . The total energy of the active contour is minimized by exploiting the greedy strategy [6]. The detailed description of these force computations has been given in our technical report [12].

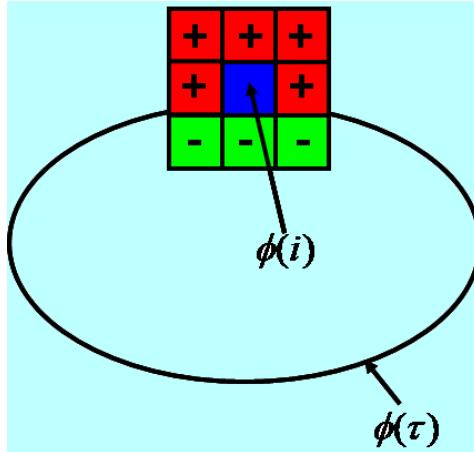
Let  $k$  and  $q$  denote a region label and a gray level, respectively:  $k = 1, \dots, K$ . The stochastic external force for each control point  $\phi(\tau)$  of a current deformable contour evolving in a region  $k^*$  is defined as follows:

$$\xi_{ext}(\phi(\tau)) = \begin{cases} -p(q|k)p(k) & \text{if } k = k^* \\ p(q|k)p(k) & \text{if } k \neq k^* \end{cases}$$

where  $q = \mathbf{Y}(\phi(\tau))$  and  $k = \mathbf{X}(\phi(\tau))$ . For each iteration of the greedy algorithm, the neighborhood of each control point  $\phi(\tau)$  is analyzed, and the neighboring pixel ensuring the smallest total energy becomes the new position for that control point as shown in Fig. 1. The iterations continue until the whole deformable model (that is, all its current control points) do not change anymore. The proposed algorithm of segmenting the region  $k^*$  is as follows:

1. Collect the empirical gray level distribution for a given image  $\mathbf{Y}$  and identify the low level MGRF model [10].
2. Use the Bayesian classifier to get the map  $\mathbf{X}$  and identify the high level MGRF model (i.e. find  $\gamma$ ).
3. Use the pixel with the maximum joint probability  $p(q, k^*)$  as an automatic seed to initialize the deformable contour.
4. For each control point  $\phi(\tau)$  on the current deformable contour, calculate sign distances indicating exterior (+) or interior (-) positions of each of the eight nearest neighbors w.r.t. the contour as shown in Fig. 1.
5. Check the label  $k = \mathbf{X}(\phi(\tau))$  for each control point:
  - (a) If the point is assigned to the region  $k = k^*$ , then
    - i. Estimate the region labels for its neighbors such that they have the (+) distance.
    - ii. If some of these sites are also assigned to the class  $k^*$ , then move the control point to the neighboring position ensuring the minimum total energy (i.e., expand the contour).
    - iii. Otherwise, do not move this point (the steady state).

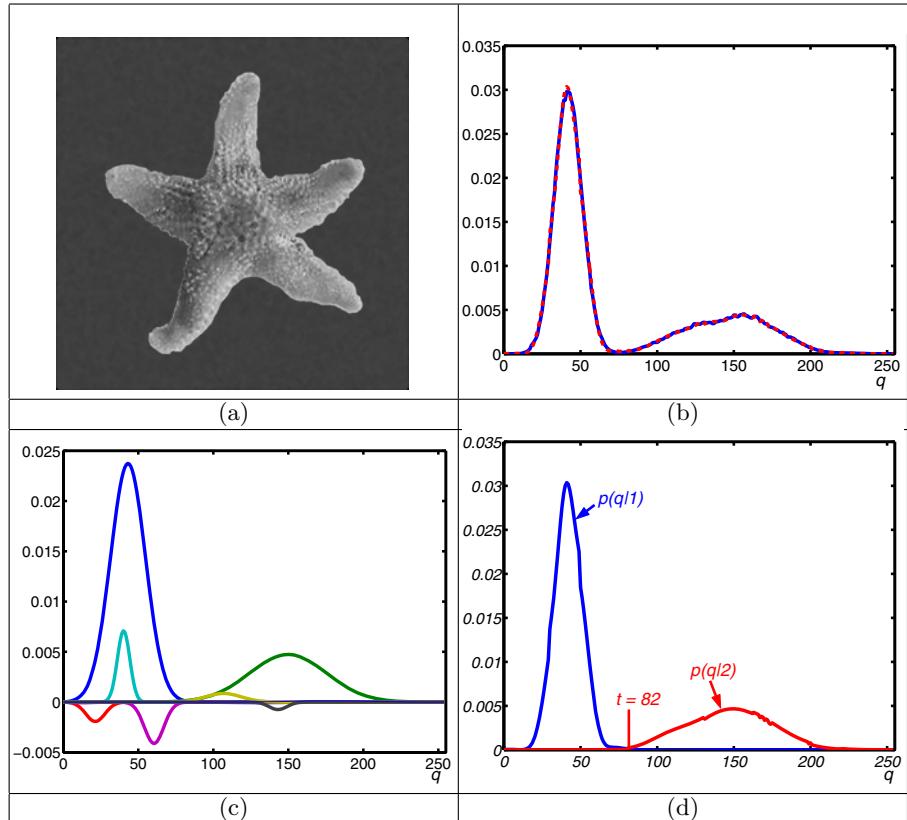
- (b) If the point is assigned to the region  $k \neq k^*$ , then
- Estimate the region labels for its neighbors such that they have the  $(-)$  distance.
  - Move the control point to the neighboring position ensuring the minimum total energy (i.e. contract the contour)
6. If the iteration adds new control points, use the cubic spline interpolation of the whole contour and then smooth all its control points with a low pass filter.
7. Repeat steps 4, 5, and 6 until no positional changes in the control points occur.



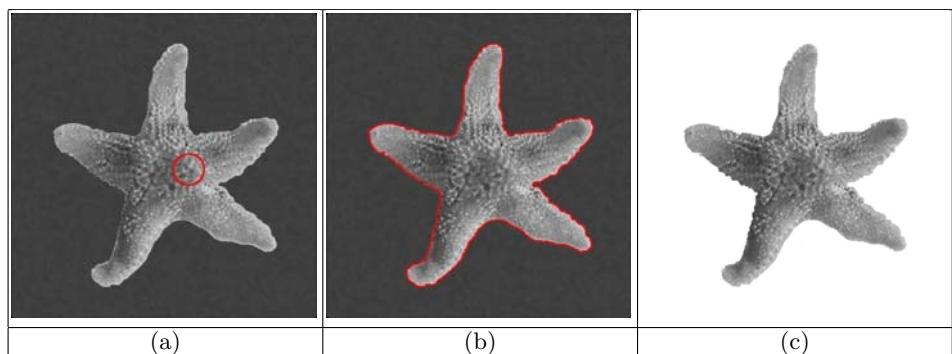
**Fig. 1.** Greedy propagation of the deformable model

## 4 Experiments and Conclusions

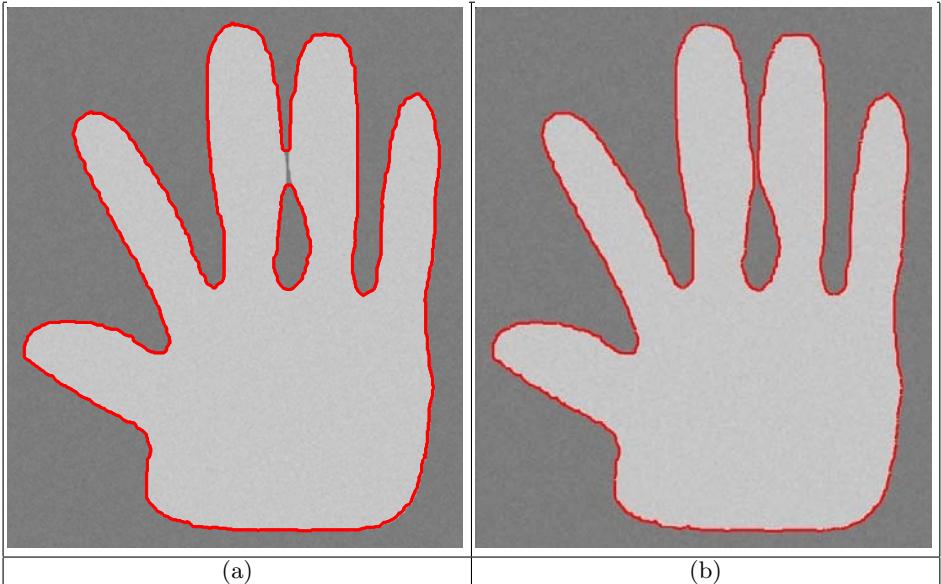
To assess robustness and computational performance, the proposed model has been tested on images of different objects with intricate shapes such as “Sea star” in Fig. 2(a). The image has only two dominant modes ( $K = 2$ ): the darker background and the brighter object. The low level model is identified by the modified EM algorithm [10]. Figures 2 (b)–(d) show, respectively, the LCG approximation of the mixed empirical bi-modal distribution of signals for the image, the individual components of the LCG, and the LCG-models for each region. Pixel-wise Bayesian classification based on this latter model produces the initial region map for the “Sea star” image. Then the Gibbs potentials are analytically estimated [11] (in this case  $\gamma = 2.17$ ), and the identified MGRF model is used to select the point with maximum joint signal/label probability to initialize the deformable contour. Figure 3(a) shows the initialization with a circle having the radius of 20 pixels from the maximum probable point. Figure 3(b) shows the



**Fig. 2.** “Sea star” (a), LCG approximation of the empirical distribution (b), individual Gaussians (c), and LCG region models (d)



**Fig. 3.** Initialization of the deformable contour (a), our segmentation with the error of 0.0036% (b), and the ground truth (c)



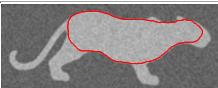
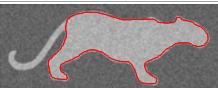
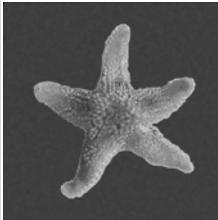
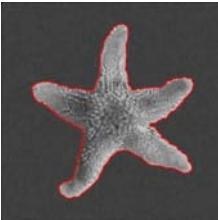
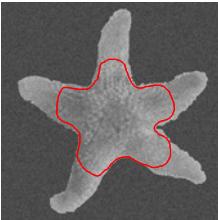
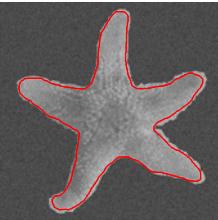
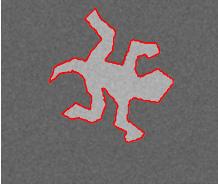
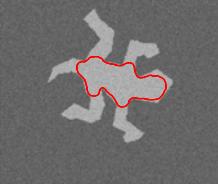
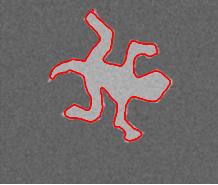
**Fig. 4.** GAC [13] (a) and our (b) segmentation of “Hand”

“Sea star” region segmented with the proposed deformable model. The segmentation error is 0.0036% with respect to the ground truth in Fig. 3(c).

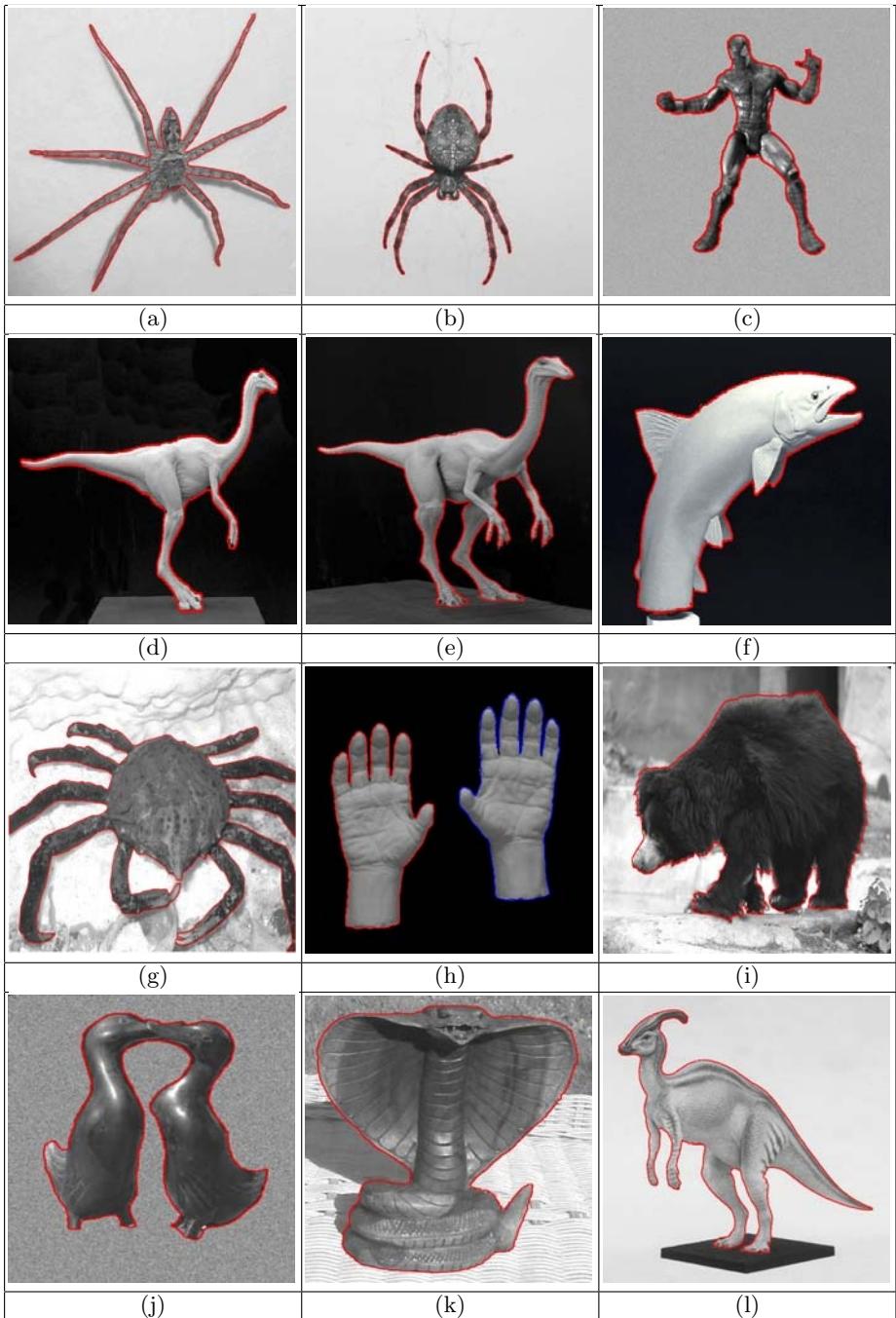
Figure 4 compares results of a popular geometric model, the geodesic active contour (GAC) [13], and our parametric model for a hand-shaped object. Our model preserves better the topology of the shape. Because two middle fingers are very close to each other, the initial curve splits into two separate curves so that the final GAC consists of a larger outer curve and a disjointed smaller inner curve shown in Fig. 4(a). Our segmentation in Fig. 4(b) keeps the separate boundary of each finger, and the final contour correctly reflects the shape of the hand.

Figure 5 highlights the advantages of our stochastic model over two conventional parametric deformable models by comparing segmentation results obtained by the proposed approach and with the greedy algorithm using the conventional deformable model proposed in [1] and with the like algorithm proposed in [2]. The two latter deformable models involve the image gradient (DMG) and the gradient vector flow (GVF), respectively, as an external force. Figure 6 adds more results obtained by our approach for objects of complicated shape.

These and other experimental results show that the proposed stochastic deformable model outperforms other known deformable models in terms of both the overall accuracy and processing time.

Image	Our model	DMG [1]	GVF [2]
	 $e = 1.21\%$ $t = 135 \text{ sec}$	 $e = 35.1\%$ $t = 420 \text{ sec}$	 $e = 15.2\%$ $t = 275 \text{ sec}$
	 $e = 0.0023\%$ $t = 121 \text{ sec}$	 $e = 32.1\%$ $t = 410 \text{ sec}$	 $e = 8.09\%$ $t = 260 \text{ sec}$
	 $e = 0.0009\%$ $t = 146 \text{ sec}$	 $e = 31.8\%$ $t = 487 \text{ sec}$	 $e = 8.6\%$ $t = 296 \text{ sec}$
	 $e = 0.0036\%$ $t = 152 \text{ sec}$	 $e = 41.1\%$ $t = 579 \text{ sec}$	 $e = 12.9\%$ $t = 290 \text{ sec}$
	 $e = 0.0027\%$ $t = 198 \text{ sec}$	 $e = 66.7\%$ $t = 610 \text{ sec}$	 $e = 13.4\%$ $t = 360 \text{ sec}$

**Fig. 5.** Comparative results on various shapes ( $e$  – the error;  $t$  – time for segmentation; final object contours are shown in red)



**Fig. 6.** More results obtained by our approach; final object contours are shown in red

## References

1. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Computer Vision*, Vol. 1, pp. 321–331, 1987.
2. C. Xu and J.L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Trans. Image Processing*, Vol. 7, pp. 359-369, 1998.
3. N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Trans. Pattern Analysis Machine Intelligence*, Vol. 22, pp. 1–15, 2000.
4. N. Paragios, O. Mellina-Gottardo, and V. Arnes, "Gradient vector flow fast geometric active contours," *IEEE Trans. Pattern Analysis Machine Intelligence*, Vol. 26, pp. 402–407, 2004.
5. A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 12, pp. 855–867, 1990.
6. D.J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *Proc. Int. Conf. Computer Vision, Graphics and Image Processing*, Vol. 55, pp. 14–26, 1992.
7. R.P. Grzeszczuk and D.N. Levin, "Brownian strings: Segmenting images with stochastically deformable contours," *IEEE Trans. Pattern Analysis Machine Intelligence*, Vol. 19, pp. 1100–1114, 1997.
8. Yu. Boykov and V. Kolmogorov, "An experimental comparison of min-cut / max-flow algorithms," in: *Proc. Third Int. Workshop Energy Minimization Methods in Computer Vision and Pattern Recogn.*, Sophia Antipolis, France, Sept. 2001 (*Lecture Notes in Comp. Science* **2134**, Springer: Berlin, pp. 359–374, 2001).
9. Yu. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in: *Proc. IEEE Int. Conf. Computer Vision*, Nice, France, Oct. 14–17, 2003, IEEE CS Press, pp. 26–33, 2003.
10. G. Gimel'farb, A. A. Farag, and A. El-Baz, "Expectation-Maximization for a linear combination of Gaussians," in *Proc. IAPR Int. Conf. Pattern Recognition*, Cambridge, UK, 23–26 Aug. 2004, IEEE CS Press, Vol. 3, 2004, pp. 422–425.
11. G. L. Gimel'farb, *Image Textures and Gibbs Random Fields*, Kluwer Academic, 1999.
12. A. A. Farag, A. El-Baz, and G. Gimelfarb, "Experimental Evaluation of Statistical and Deformable Model-Based Segmentation," TR-CVIP04-11, Computer Vision and Image Processing Laboratory, University of Louisville, KY, November 2004.
13. R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast Geodesic Active Contours," *IEEE Trans. Image Processing*, Vol. 10, no. 10, pp. 1467-1475, 2001.

# Dissolved Organic Matters Impact on Colour Reconstruction in Underwater Images

J. Åhlén<sup>1</sup>, D. Sundgren<sup>2</sup>, T. Lindell<sup>3</sup>, and E. Bengtsson<sup>3</sup>

<sup>1</sup> Centre for Image Analysis, Uppsala University  
Kungsbäcksv. 47, 801 76 Gävle, Sweden  
[jae@chig.se](mailto:jae@chig.se)

<sup>2</sup> Department of Computer and Systems Sciences,  
Royal Institute of Technology,  
Kungsbäcksv. 47, 801 76 Gävle, Sweden  
[dsn@chig.se](mailto:dsn@chig.se)

<sup>3</sup> Centre for Image Analysis, Uppsala University,  
Lägerhyddsvägen 3, 752 37 Uppsala, Sweden  
[{tommy, ewert}@cb.uu.se](mailto:{tommy, ewert}@cb.uu.se)  
<http://www.cb.uu.se>

**Abstract.** The natural properties of water column usually affect underwater imagery by suppressing high-energy light. In application such as color correction of underwater images estimation of water column parameters is crucial. Diffuse attenuation coefficients are estimated and used for further processing of underwater taken data. The coefficients will give information on how fast light of different wavelengths decreases with increasing depth. Based on the exact depth measurements and data from a spectrometer the calculation of downwelling irradiance will be done. Chlorophyll concentration and a yellow substance factor contribute to a great variety of values of attenuation coefficients at different depth. By taking advantage of variations in depth, a method is presented to estimate the influence of dissolved organic matters and chlorophyll on color correction. Attenuation coefficients that depends on concentration of dissolved organic matters in water gives an indication on how well any spectral band is suited for color correction algorithm.

## 1 Introduction

Colour correction of underwater images mainly focuses on balancing colours where blue colour can be prevailing [1]. The bluishness in images caused by apparent optical properties of water where light in the red and yellow region of the spectra will be more attenuated with increasing depth than in the blue spectral region [2]. Underwater photographers which are concerned with more natural color representation in the images use filters, but filters cannot fix color problems when shooting in ambient light at depths where red and yellow wavelengths of light are almost totally attenuated [3]. Underwater photographers usually set

up a pair of strobes on the camera as an extra light source. The drawbacks are though that the subject may be too large for conventionally mounted strobes, or the photographer working with strobes at depth may be unable to approach the subject close enough to accommodate the limited illumination range of strobe lights [4]. In either case, the result is that the subject is illuminated solely by ambient light in which red and yellow wavelengths may be deficient or totally absent.

The camera reproduces the wavelengths of light that enter the lens as faithfully as the design of the film emulsion or CCD sensor allows. Some cameras includes beautifying functions that adapt to the dominating light spectrum and filter the intensity values that have been registrated. Based on intensity values of the image we can estimate the downwelling irradiance and diffuse attenuation coefficients  $K_d$  of the diving site. Strictly speaking,  $K_d$  is not a property of the water itself but rather a descriptor of the underwater light field that varies with depth, solar altitude, and time. The reconstruction of colour in the image is based on the spectral model of the acquisition system and a gray reflectance target, which is present in each image [5]. By inverting the model, an estimation of the spectral reflectance of each pixel is possible. The reflectance of an object can be considered as a ratio of the intensity of reflected radiant energy to that reflected from a defined gray reference standard. In image collection and further processing the authors are using a gray Spectralon with 99% reflectance as such a standard [6].

Estimated values of  $K_d$  values gives a possibility to calculate the absorption coefficients of dissolved organic matters and chlorophyll for that particular water. We are arguing that the absorption coefficients can give an indication on what  $K_d$  values should be used in color reconstruction of the images.

Not only the pure water itself will affect the optical climate of the water mass and affect the colour in the photos taken at different water depths, but also all other inherent optical properties will have impact, sometimes substantial. So far our work has mainly been dealing with correction for the attenuation of the water itself. In a certain environment it could be recommendable to colour correct or at least be aware of possible effects of substances in the water.

### 1.1 Dissolved Matters and Chlorophyll

Two essentially different types of substances or substance behaviour need to be discussed. We separate between dissolved and suspended matter. The dissolved matter is commonly of organic origin and usually named yellow substance. Also the salt in sea water could have some influence of the optical behaviour. The suspended matter could be either inorganic or organic and the organic matter could be dead or living. It is common to call the inorganic matter just suspended matter and it is also common to treat the phytoplankton separately, due to its water quality importance.

The inorganic matter has a rather well defined behaviour, mainly as a matter of particle size distribution and the influence of dissolved matter is also rather easy to model. Phytoplankton influence, however, is very complex and the com-

plexity is much worse due to the fact that we always have a mixture of all three agents in natural waters.

In situ spectral absorption coefficient profiles can be measured with spectral radiometers [7]. This means that in order to calculate the spectral absorption of a particular water layer it is enough to measure a spectra with spectrometer. The instrument we use includes 1024 spectral channels and register pointwise the intensity counts. The concentration of dissolved organic matter is usually based on the amount of organic carbon present in the water. Experiments have shown that the spectral absorption of yellow substance varies approximately exponentially with wavelength and increases with decreasing wavelength [2]. Phytoplankton specific absorption coefficients have two absorption peaks at 440 and 675 nm. The peak at 440 nm is stronger than at 675 nm [8]. In clear near shore ocean waters the concentration of non-organic suspended matters is predominant, however we are not concerned with the calculation of those. Concentration of dissolved organic matters and chlorophyll may vary greatly for different diving sites thus should be measured in situ or estimated from measured irradiances.

Underwater photos are most commonly taken in a coral reef environment and here we often have very clear water with little material, suspended as well as dissolved. Typically there are very low concentrations of small fractions of inorganic matter. Only occasionally a plankton bloom may occur. Our colour correction model [5] is therefore often sufficient for reefs photos.

As we have taken photos in Florida, Portugal and Sweden we do need to, at least theoretically, discuss the colour effects of substances in the water as suspended matter, phytoplankton and yellow substances are very common in those waters, typically a mixture of all three. In this paper we will discuss the combined effects of dissolved matter and phytoplankton. The influence of suspended matter will be discussed in a separate paper.

## 2 Estimating the Absorption Coefficient for Dissolved Matters and Chlorophyll

One way to express the concentration of dissolved organic matters in water is to model the absorption property of the matter. In phytoplankton-dominated waters near the sea surface, the total absorption and backscattering coefficients can be written in terms of chlorophyll concentration  $C$  and yellow substance factor  $Y$  for each depth interval, see Eq. 1.

$$a(\lambda, z) = a_w(\lambda) + 0.06a_{ph}^*(\lambda)C(z)^{0.65} + Y(z)a_{ph}(440)\exp(-0.014(\lambda - 440)) \quad (1)$$

From this equation it can be seen that  $Y$  describes the relationship between the absorption by yellow substance at 440 nm and the absorption by phytoplankton at 440 nm.

The equations 2 and 3 for concentration of chlorophyll and yellow substances below are from [8]. Eq. 2

$$C = ((\delta_{ji}K_d - 0.686\delta_{st}K_d + 0.0753)/0.0746)^{1.54} \quad (2)$$

and  $Y$  Eq. 3

$$Y = -(\delta_{ji}K_d + 0.0049 - 0.0107C^{0.65})/0.0625C^{0.65}, \quad (3)$$

where  $\delta_{ji}K_d = K(\lambda_j, zz_1) - K(\lambda_i, zz_1)$ . An analysis of the commonly used wavelengths in ocean color measurements showed that the optimal wavelength pairs to solve these equations are (412, 443) and (443, 555) [8]. The  $a_w(\lambda)$  and  $a_{ph}^*$  coefficients are commonly used and found in [9, 10].  $K_d$  values are estimated from intensity count measurements with the spectrometer at different depths.

### 3 Correction of Underwater Images

The general method requires the presence of the gray standard in each photographed scene. However for more accurate estimation of the attenuation coefficients we are forced to further measurements with the spectrometer. Apart from that we apply the correct spectra on the images in order to remove all the beautifying effects built into the camera so that the image is as correct as possible before it is subjected to colour correction algorithm. Test imagery was collected off the coast of Lisbon at Sisimbra in April 2004. For each image we recorded the exact depth at which it was taken. To diminish the water absorption effects we use Beer's Law [11] to calculate the new intensity values for the image taken at depth  $z$  as if it was taken at depth  $z_1$ . The method consists of these steps:

1. We are using a spectrometer for this study and measure the intensity counts when pointing to the gray reflectance plate. We can then use the known reflectance of the gray plate to extrapolate the downwelling irradiance to the depth in question. Downwelling irradiance is proportional to intensity counts that we have from spectrometer, which means that the ratio between two downwelling irradiances for different wavelengths is equal to the ratio between two corresponding intensity counts.
2. We calculate  $K_d$  values from estimated downwelling irradiances for each depth interval by using the following equation:

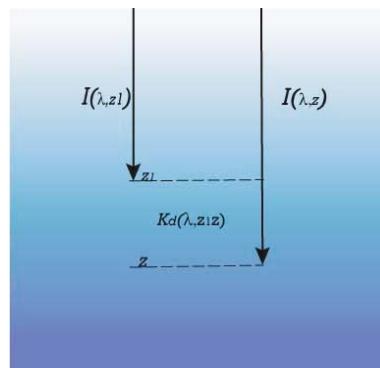
$$K_d(\lambda_i, zz_1) = \frac{I(\lambda_i) - I(\lambda_j)}{zz_1 I(\lambda_i)}, \quad (4)$$

as can be seen in Figure 1.

3. The variation of downwelling irradiance with depth can be approximated by

$$I(z_1) = I(z) e^{K_d(z) - K_d(z_1) z_1 - z}. \quad (5)$$

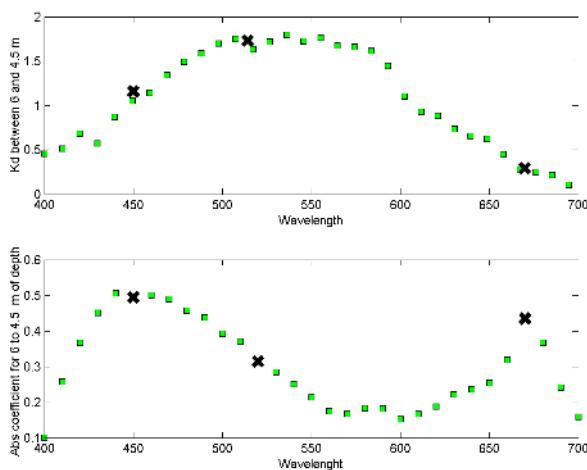
4. Each pixel for red, green and blue components of the image is subject to this operation and as a result we "lift up" the image to depth  $z_1$ . For this operation we choose three  $K_d$  values for the three spectral channels red, green and blue.



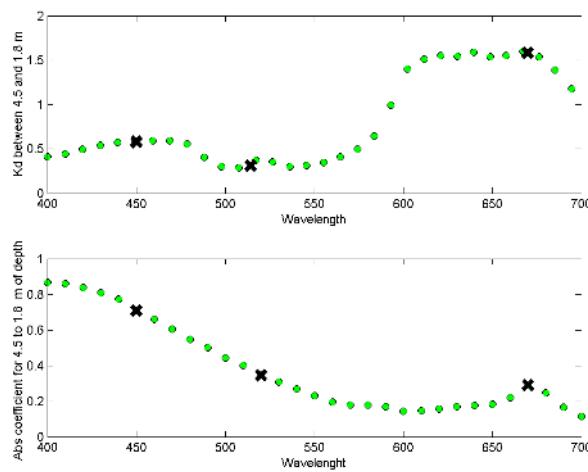
**Fig. 1.** Illustration of Beer's Law

## 4 Results

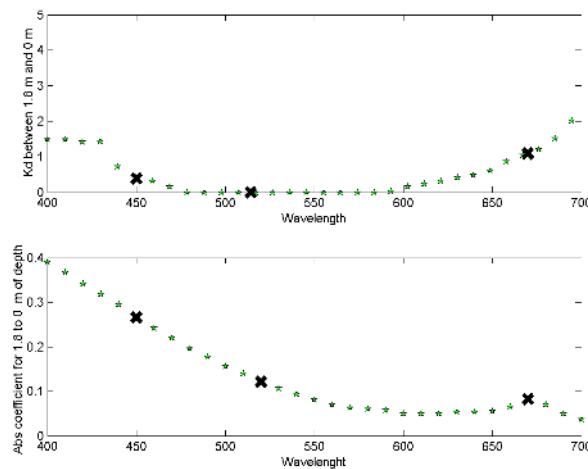
By modeling the absorption behaviour for each available wavelength we can clearly see which wavelengths are more sensitive to the absorption of dissolved organic matters in the water. In Figures 2, 3 and 4 we see the  $K_d$  values and absorption coefficients, respectively, plotted as functions of wavelength. As expected the shape of curves for the absorption coefficients shows the presence of phytoplankton in the water. The points marked with  $\times$  show which wavelength represents red, green and blue, respectively. We can see clearly that  $K_d$  values representing



**Fig. 2.**  $K_d$  used for correction of colors in images at depths between 6 m and 4.5 m compared to absorption coefficients



**Fig. 3.**  $K_d$  used for correction of colors in images at depths between 4.5 m and 1.8 m compared to absorption coefficients



**Fig. 4.**  $K_d$  used for correction of colors in images at depths between 1.8 m and 0 m compared to absorption coefficients

the red spectral interval correspond to the same wavelength as the lower peak value of absorption coefficient. This is illustrated especially clear in Figure 2.

In the graphs we can see that the value for the red channel corresponds to one of the absorption peaks. We set the marked values in the color reconstruction algorithm. The image that is taken at depth of 4.5 meters was corrected as to show what it would have looked like had it been taken at 1.8 meters depth, see Figures 5 and 6.



**Fig. 5.** Original image taken at 4.5 meters of depth



**Fig. 6.** Corrected image



**Fig. 7.** Corrected image with new  $K_d$  value for the red channel

To avoid the influence of phytoplankton when red channel is color corrected we choose another value of  $K_d$  that does not correspond to the peak of total absorption coefficient and also correspond to the least rate of change in the red channel. Experiments have shown that the best value to be set into Beer's law for red channel is 1.56 instead of 1.60. The result is shown in Figure 7.

## 5 Conclusions

The  $K_d$  values that are used in the color correction algorithm can be chosen based on the concentration of dissolved organic matters in the particular water layer. The red channel (the interval between 600 and 700 nm) is the most sensitive for underwater imaging therefore we should take into account every possible source of noise in this channel. In order to choose the least sensitive  $K_d$  value for colour correction we consider both of the following points:

- The representative wavelength for the red channel should not be in the neighborhood of the one that corresponds to the peak value of the total absorption coefficient.
- The rate of change for the  $K_d$  curve should be minimal in the red interval.

Since environmental differences will result in variations of optical water properties we need to find  $K_d$  values for each diving site. From those estimated values we have to choose three that represent the red, green and blue spectral intervals. We have found an indicator on what  $K_d$  values should be chosen for color correction. This will reduce the time needed to investigate and test the different  $K_d$  values.

For this study we would benefit from images and spectrometer data taken at depths with smaller interval. This would most likely give us less varying  $K_d$ .

We tested the method on the images taken in Mid Atlantic where corals are absent. In such waters the dissolved organic matters and chlorophyll are predominant, but scarce. Therefore the change of  $K_d$  values did not have a dramatic effect on colour correction. In the next stage of this project we would model the influence of color reconstruction by presence of suspended non-organic matters which usually dominate water column above coral reefs.

## Acknowledgments

We would like to thank KK-foundation for the financial support for this project.

## References

1. Fissenko, T., Y., Fissenko, V., T., Rogatchev, G., A. and Sushechev. G., A. An interpolation method for color image reconstruction in deep underwater observation. In: *Proc. of D. S. Rozhdestvensky Optical Society: The II International Conference, Current Problems in Optics of Natural Waters*, St.Petersburg, Russia, 2003, pp. 113-118.

2. R.E. Walker, Marine Light Field Statistics, John Wiley & Sons, Inc, 1994.
3. R. Delfs, Getting Rid of the Underwater Blues, *Wetpixel.com*, url: <http://www.wetpixel.com>, February 2005.
4. A. Kohler, D. Kohler, Underwater Photography Handbook. UK London: New Holland Publishers, 1998.
5. J. Åhlén and E. Bengtsson and T. Lindell, Color Correction of Underwater Images Based on Estimation of Diffuse Attenuation Coefficients, In *Proceedings of the PICS Conference An International Technical Conference on The Science and Systems of Digital Photography including the Fifth International Symposium on Multispectral Color Science*, Rochester, NY, 13-16 May 2003.
6. Spectralon, Reflectance Material for Component Fabrication, *Labsphere* url: <http://www.labshere.com/products/Products.asp>, January, 2005.
7. N.K. Hojerslev, A spectral light absorption meter for measurements in the sea. 1975 In: *Limnol. Oceanogr.*, 20: 1024-1034.
8. J. S. Bartlett, M. R. Abbott, R. M. Letelier and J. G. Richman, Chlorophyll Concentration Estimated From Irradiance Measurements At Fluctuating Depths, In: *Ocean Optics XIV*, Kailua-Kona, November 1998.
9. L. Prieur and Sathyendranath, An optical classification of coastal and oceanic waters based on the specific spectral absorption curves of phytoplankton pigments dissolved organic matter and other particulate materials for 400-700 nm, In *Limnol. Oceanogr.*, vol. 26(4), 1981, pp. 671-689.
10. R. C. Smith and K. S. Baker, Optical properties of the clearest natural waters (200-800) nm, In *Appl. Opt.*, vol. 20, 1981 pp.177-184.
11. Calculations Using Beer's Law, url: [http://www.oceansonline.com/beers\\_law.htm](http://www.oceansonline.com/beers_law.htm), (November 2002)

# Denoising of Time-Density Data in Digital Subtraction Angiography

Hrvoje Bogunović and Sven Lončarić

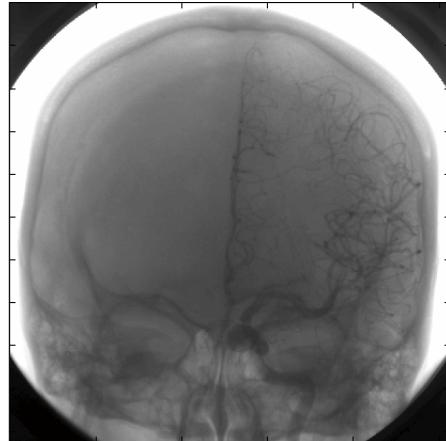
Faculty of Electrical Engineering and Computing,  
University of Zagreb, Unska 3, 10000 Zagreb, Croatia  
[{hrvoje.bogunovic, sven.loncaric}@fer.hr](mailto:{hrvoje.bogunovic, sven.loncaric}@fer.hr)  
<http://ipg.zesoi.fer.hr>

**Abstract.** In this paper we present methods for removing the noise from a time sequence of digitally subtracted x-ray angiographic images. By observing the contrast agent propagation profile in a region of the angiogram one can estimate the time of arrival of that agent. Due to the large level of noise, it is difficult to detect the moment of the contrast agent arrival accurately. Hence denoising is required. Two methods are presented. The first one is based on 1D Wiener filtering of the time data. Wiener filter was chosen because it presents the optimal linear filter in the least-squares sense. The other method is based on 3D wavelet denoising via wavelet shrinkage technique, which is a nonlinear method. Since it is based on 3D wavelet basis it can perform denoising simultaneously in the spatial as well as in the time dimension of the image sequence. Wavelet based denoising proved to be superior but computationally more demanding. The experiments were performed on a sequence of cerebral angiograms.

## 1 Introduction

Digital subtraction angiography (DSA) is often used to assess the structure of vessels. By injecting the contrast agent, which attenuates the x-rays, the vessel structure in the observed region of interest (ROI) becomes visible. Instead of viewing just the morphology, DSA can also be used to detect e.g. time of arrival of the injected agent to different regions of the tissue. This information can prove vital in detecting and locating the infarcted regions.

For acquiring the necessary data for later processing, the image acquisition process is defined. It consists of the following steps. First, contrast agent is injected into the main feeding artery. Since we are acquiring images of the brain, contrast agent in our case was injected into one of the carotids so only one brain hemisphere becomes visible. By using the high frame rate, we can get an image sequence showing propagation of the contrast agent. One image from a sequence where contrast agent is visible is shown in Fig. 1 In order to estimate the exact moment of agent arrival one needs to observe the bolus traversal time profile in the ROI. Such time-density signal is usually too noisy for the straightforward estimation of the arrival time, hence denoising of the time-density data is necessary.



**Fig. 1.** AP view showing contrast agent in the left hemisphere

We made the first attempt by using the Wiener filtering approach for smoothing the 1D signal which presents the time-density curve of a ROI. Later we used a different approach by using the 3D wavelet decomposition and performed the nonlinear denoising by wavelet shrinkage.

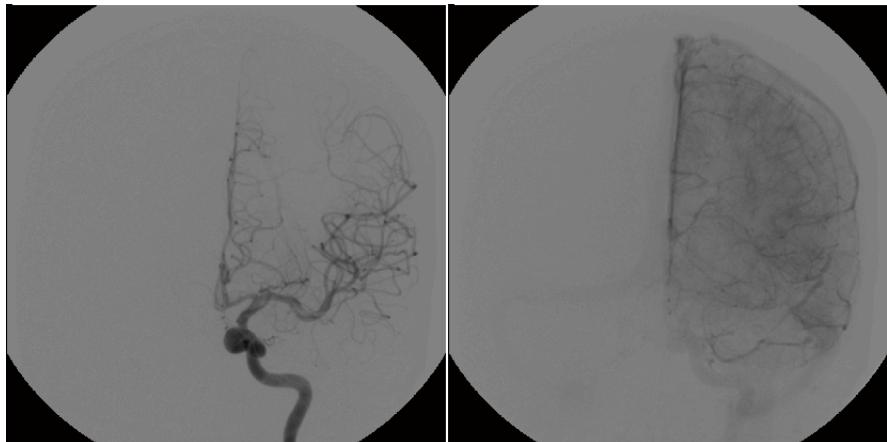
The outline of the paper is as follows. Background about the digital subtraction angiography is given in section 2. Time-density curves and time of arrival estimation are explained in section 3. Denoising techniques are discussed in section 4, which is the central part of the paper. Results and comparison of the used techniques are given in section 5 and section 6 concludes the paper.

## 2 Digital Subtraction Angiography

The product of the X-ray imaging device is an image showing both the enhanced vessel as well as the surrounding structure such as e.g. the bones. It is difficult to observe the vessels in such an image. Also the intensity value in the vessel is not proportional just to the amount of the contrast agent but also the structures behind the vessel contribute to the resulting intensity value due to X-ray device's projective nature.

To enhance the contrast agent visibility it is first necessary to eliminate background information from each image. This is performed using the digital subtraction from the mask image which is taken prior to the contrast injection and which shows only the non-vessel structure. This technique is called digital subtraction angiography. Before subtraction takes place both images are logarithmized to eliminate the thickness of the human body [1].

$$\begin{aligned} I_{mask} &= I_0 e^{-\mu_t x_t} \\ I_{contrast} &= I_0 e^{-(\mu_t x_t + \mu_I x_I)} \\ S &= \ln(I_{mask}) - \ln(I_{contrast}) = \mu_I x_I \end{aligned} \quad (1)$$



**Fig. 2.** Two DSA frames showing contrast propagation

Where  $x_t$  is the thickness of the body,  $x_I$  is the thickness of the vessel filled with contrast agent and  $\mu_t$ ,  $\mu_I$  are the corresponding attenuation coefficients. The intensity value of the subtracted image  $S$  is proportional only to the amount of contrast in a region represented by a pixel.

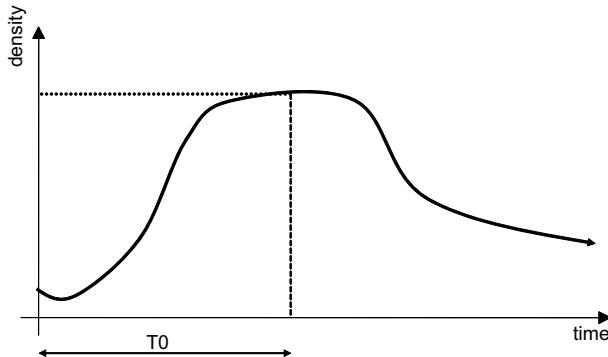
In order to remove the unnecessary anatomy the mask image has to align accurately with the image containing the agent. The alignment can be affected by the motion of the patient, thus motion compensation normally precedes the subtraction. In our test images, motion of the head was negligible so no compensation was performed.

The result of DSA operation is a sequence of images showing only the contrast propagation. Two frames at different time instant are shown in Fig. 2. This sequence is used as the input image in the later parts of our procedure.

### 3 Time-Density Signals

The pixel value of the input image in this stage of the procedure is proportional to the amount of the contrast agent in that region. For constant region volume this can be interpreted like the pixel value is proportional to the density of contrast agent in the volume. When we observe one region through time we can draw a time-density curve. Idealistic view of the time-density curve for the region, which is traversed by the contrast agent is shown in Fig. 3. Propagation of the contrast agent is clearly visible in the figure. Time of arrival (TA) is depicted as time from start of the sequence till the contrast reaches its maximum density value. This was taken to be analogous to time of arrival parameter commonly used in CT measurements [2].

At the end we want to produce as a result an image, representing the whole angiogram sequence, showing the time of arrival for each region where each region is commonly represented by a pixel.

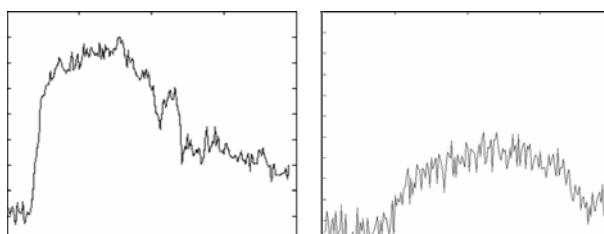


**Fig. 3.** Time-density signal

## 4 Denoising

In reality a time-density signal is not as smooth as it was previously shown in Fig. 3. Noise is present in the image sequence and it also appears in the curve. Additionally, due to the projective nature of X-ray imaging device, it is possible that the observed signal is a result of multiple signals that are behind each other and now are merging into one. To be able to find the main feature points of the signal and to estimate the time of arrival, denoising is required.

There are two types of signals. One appears in the artery and vein region and they have high Signal-to-Noise ratio (SNR) since both of these vessels consume large quantities of blood. The other type is the capillary region where SNR is much lower due to small amount of contrast that enters such region. Examples of such signals are depicted in Fig. 4. The biggest issue in designing the denoising filter is how to effectively model the noise. Normally all imaging devices observe speckle noise which has the Poisson distribution. We found reasonable the assumption that since our images were not fluoroscopic but rather high intensity ones that the number of emitted particles is large enough that we can approximate Poisson distribution with the Gaussian.



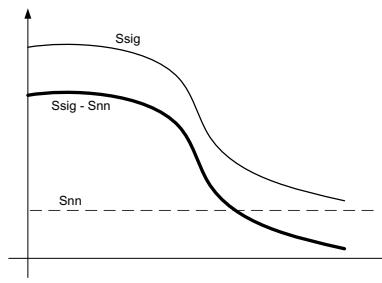
**Fig. 4.** Typical time density signal in the artery ROI (left) and in the capillary ROI (right)

#### 4.1 Wiener Filtering

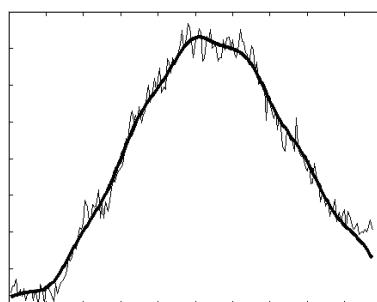
Each 1D time-density signal can be viewed as a realization of a stochastic process. We model noise as an additive white Gaussian noise, uncorrelated with the signal i.e. independent and identically distributed (i.i.d.). The stochastic process modeling the signal without noise has low-pass nature due to the fact that time-density signals are slowly changing functions. The power spectrum of the signal corrupted by noise is a wide band process where the values of higher frequency components are due to noise. So we can estimate noise from the higher frequency part of the signal power spectrum. Since power spectrum of the white noise is flat, the high frequency part is equal to the whole power spectrum. We estimate the noise power spectrum from the highest 50% of the power spectrum samples of the original signal.

By subtracting the power of noise from the original power spectrum, what remains is the power spectrum of a cleaned version of our original signal (Fig 5). So the transfer function which defines the Wiener filter is given by:

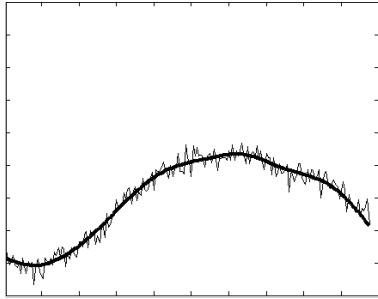
$$H_{wiener} = \frac{S_{sig} - S_{nn}}{S_{sig}} . \quad (2)$$



**Fig. 5.** Power spectrum subtraction



**Fig. 6.** Wiener denoising of artery ROI



**Fig. 7.** Wiener denoising of capillary ROI

Results of the filtering in two regions are shown in Fig. 6 and Fig. 7. The resulting denoised time-density curves turned out to be satisfying although improvements are possible.

Additionally, spatial smoothing can be performed by growing the regions of interest. Then, such regions are composed of multiple pixels of the original input image, and are average representatives of these pixels. Spatial smoothing is desirable because it is reasonable to expect that neighboring pixels will have the similar time of arrival. Side-effect of spatial smoothing is the blurred perception of the image since the edges between the vessels and the background are also smoothed out.

## 4.2 Wavelet Denoising

In order to evade the image blurring effect we turned towards the wavelet oriented denoising. Denoising is one of the most important applications of wavelets. Wavelet denoising should not be considered a smoothing method because smoothing always eliminates the high frequency part and keeps the low frequency one. By observing the signal at different scales wavelets are possible to remove the noise without destroying all the high frequency information.

Denoising via wavelet shrinkage method was first suggested by [3]. It involves three steps: a linear forward wavelet transform, nonlinear shrinkage of the wavelet coefficients in the wavelet domain and a linear inverse wavelet transform. Wavelet transform has a nice property of de-correlating the signal. Meaning that due to its compact base the signal will be presented by only few non-zero coefficients in the wavelet domain. On the other hand noise which is not correlated will be distributed throughout the wavelet domain in every scale and the corresponding wavelet coefficients will be small because orthogonal wavelet transform preserves the energy of the signal. This means that the wavelet coefficients of a noisy signal are also noisy and if we could eliminate the noise in the wavelet domain, we would also eliminate it in the time domain. Since wavelet coefficients corresponding to noise are small in value and coefficients corresponding to the signal are large, simple thresholding would perform denoising successfully.

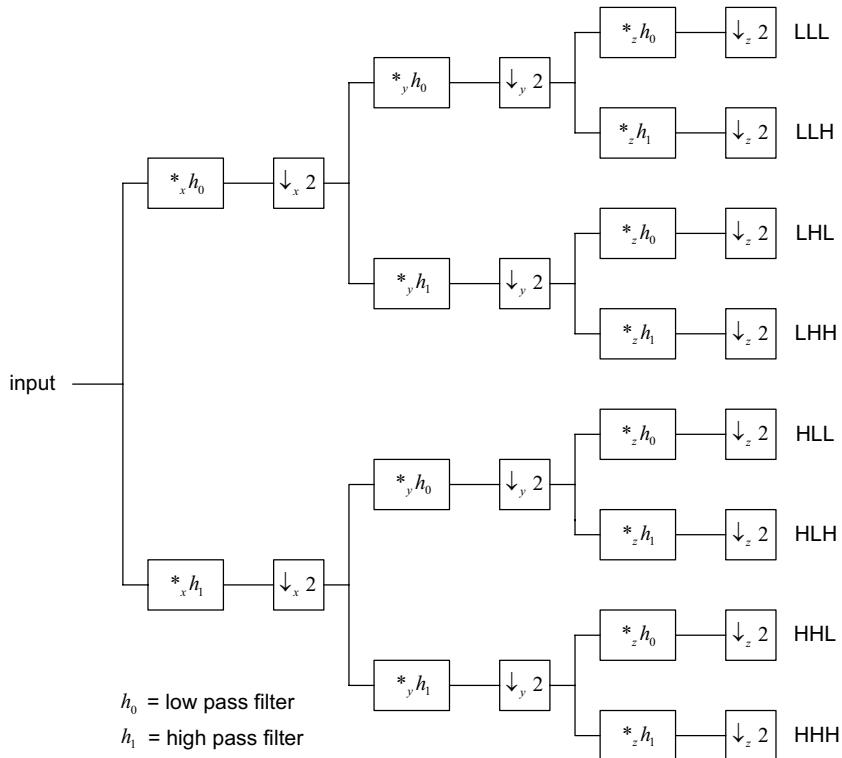


Fig. 8. 3D separable wavelet decomposition

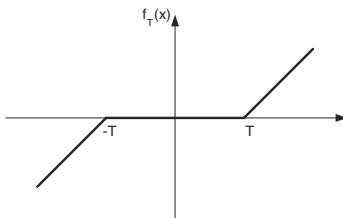


Fig. 9. Soft thresholding

Again we assume additive white Gaussian noise model which is i.i.d. To be able to perform denoising simultaneously in both the time and the spatial domain we opted for 3D orthogonal wavelet transform based on Daubechies db2 wavelet (corresponding filters have 4 coefficients). Such 3D wavelet transform is calculated in a separable manner (1D wavelet transform in each dimension). In general, each 3D wavelet decomposition produces 8 sub-volumes as shown in

the following Fig. 8. We used soft thresholding [4] technique and we chose universal threshold [3]. Function for performing the soft thresholding with particular threshold  $T$  is shown in Fig. 9.

The threshold value is not changed between the subbands, hence the name “universal”. The universal threshold is estimated from the lowest scale because it is usually the noisiest subband. In our case that is  $HHH_1$  subband. Although there are numerous other methods for obtaining the threshold parameter which

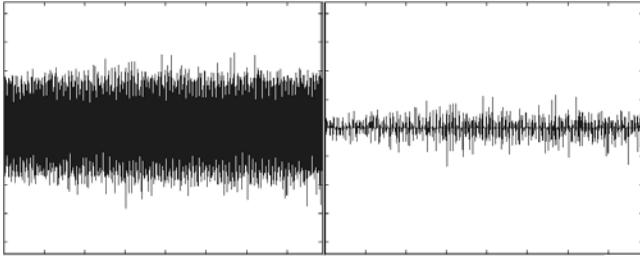
are subband adaptive we found that the universal thresholding performed good enough. Universal threshold  $\lambda_{univ}$  is found by the following formula:

$$\lambda_{univ} = \sigma \cdot \sqrt{2 \ln(N)} . \quad (3)$$

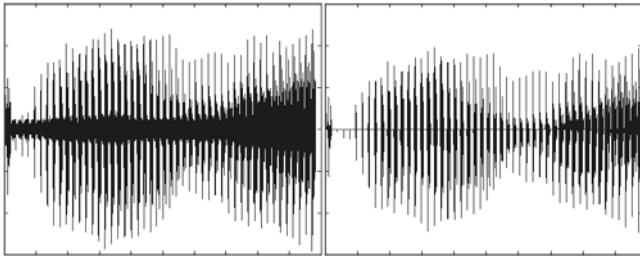
$N$  is the signal length and  $\sigma$  is the standard deviation of the noise.  $\sigma$  is estimated from the  $HHH_1$  subband coefficients by:

$$\sigma = median(|HHH_1|)/0.6745 . \quad (4)$$

The value 0.6745 comes from the median of white gaussian noise with distribution  $\mathcal{N}(0, 1)$ , so the estimator is calibrated for noise with normal distributions. The fact that the  $\sigma$  is estimated quite effectively is shown in Fig. 10 and Fig. 11.



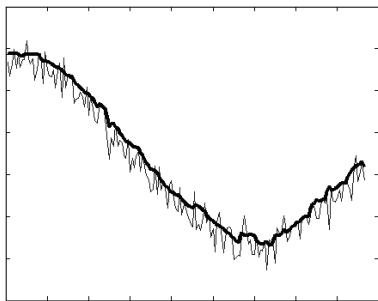
**Fig. 10.**  $HHH_1$  wavelet coefficients before (left) and after (right) thresholding



**Fig. 11.**  $HLL_2$  wavelet coefficients before (left) and after (right) thresholding

We performed four levels of decomposition and applied soft thresholding to all the detail coefficients. The example of a resulting 1D time signal is shown in Fig. 12. The denoised signal in Fig. 12 is not as smooth as the wiener de-noised one but it is more correlated to the neighboring time-density curve due to simultaneous 2D spatial denoising.

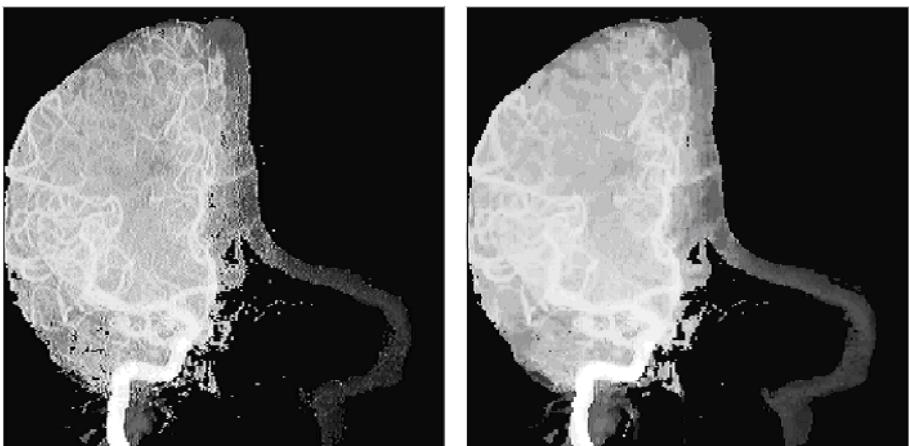
The contribution of 2D denoising is more evident when displaying the results of the time-of-arrival estimation which will be discussed in the next section.



**Fig. 12.** Signal in the capillary ROI after 3D denoising

## 5 Results

The resulting images showing the time of arrival can now be presented for different denoising schemes. The results will be compared to discover the preferred denoising technique. The biggest problem with the most of the medical image analysis methods is the lack of the ground truth. We have the same problem here and currently we can only visually assess the quality of the results. Results for 1D wiener filtering and 3D wavelet denoising are presented in Fig. 13. By observing the resulting images we can notice the following. Wiener filtering method gives results which still contain a lot of noise i.e. resulting neighboring pixels are not correlated enough. The results of wavelet denoising are smoother and the edges are preserved so the images are not blurred.



**Fig. 13.** Result for the Time of Arrival (brighter intensity means shorter time of arrival) after Wiener denoising (left) and Wavelet denoising (right)

## 6 Conclusion

Different techniques for the denoising of DSA image sequence for calculating the time of arrival of the injected contrast agent are presented. Time dimension contains slowly changing signals so smoothing may be sufficient, but in spatial dimension smoothing leads to the blurred perception. Wiener filtering of 1D time-density curves enables the estimation of time-of-arrival but the resulting 2D images are too noisy. On the other hand 3D wavelet non-linear based denoising performs both the time and the spatial denoising and thus gives better results but for the cost of larger computational complexity.

## Acknowledgments

The authors wish to thank Robert Homan for helpful discussions and Dr. Jacques Moret, from the Fondation Rothschild, Paris, for providing us with the image sequences of the brain.

## References

1. Hasegawa, B.: The Physics of Medical X-Ray Imaging, 2nd edition. Medical Physics Publishing Corporation. (1987)
2. Konig, M.: Brain perfusion CT in acute stroke: current status. European Journal of Radiology. **45** (2003) S11–S22.
3. Donoho, D., Johnstone, I.: Ideal adaptation via wavelet shrinkage. Biometrika. **81** (1994) 425–455.
4. Donoho, D.: Denoising by soft thresholding. IEEE Transactions on Information Theory. **41(3)** (1995) 613–627.

# The Use of Image Smoothness Estimates in Speeding Up Fractal Image Compression

Tomas Žumbakis<sup>1</sup> and Jonas Valantinas<sup>2</sup>

<sup>1</sup> Kaunas University of Technology, LT-51368 Kaunas, Lithuania  
t.zumbakis@vbt.lt

<sup>2</sup> Kaunas University of Technology, LT-51368 Kaunas, Lithuania  
jonas.valantinas@ktu.lt

**Abstract.** The paper presents a new attempt to speed up fractal image encoding. The range blocks and corresponding domain blocks are categorized depending on their smoothness parameter values (smoothness estimates), introduced, from the first, to characterize manifestation of high frequency components in the image. The searching of the best matched domain block is carried out between the neighbouring (or, within the same) smoothness classes. The computational complexity of the fractal image encoding process is reduced considerably. Theoretical and experimental investigations show that extremely high compression time savings are achieved for images of size 512x512.

## 1 Introduction

Those, who are closely bound up with digital images, acknowledge that the digital image processing problems, as well as image processing technologies, experience changes with the time flying by. There is no denying the fact that topics, such as detection of fractal nature of an image, image segmentation and texture analysis, synthesizing images, etc., came to the first place, [1-5]. The digital image analyzing tools have been expanding too – the new and promising mathematical means (discrete transforms, elements of fractal geometry, fundamental of genetics, etc.) were put at hand, [6-9].

In the field of digital images (computerized real-world image models) the fractal approach is of outmost importance, because it facilitates perception and understanding of the information content of an image. To say more, it provides us with a powerful means to catch sight of a fundamental real-world image property generally known as self-similarity. Due to this property, the research and development of algorithms (fractal techniques) to extract important fractal parameters from appropriate digital data has received significant attention in recent years. One of the most rapidly developing areas is the use of fractal geometry for performing image data compression.

Everybody, who is gone deep into the essence of the matter, comprehends that merely the extractability of self-similarity, found within images, made it possible to construct the fractal representation of an image. A. Jacquin was the first to propose a practical block based fractal image coding scheme (idea) in 1990, basis of most published fractal image coding schemes, [2, 5, 10, 11].

However, the encoding complexity of the fractal image coding is extremely high and has become the major obstacle for its workable practical applications. In sum, the most computationally intensive part of the fractal encoding (compression) process is the searching step. If a brute force approach (full search) to the detection of optimal pairings “range block – domain block” is used, the fractal encoding complexity is always dominated by this searching process. The design of efficient domain search strategies (searching algorithms) has consequently been one of the most active areas of research in fractal coding, resulting in a wide variety of solutions. A survey of some really significant advances is represented in [10, 11]. Unfortunately, we have to emphasize that, despite numerous and many-sided attempts to accelerate fractal image encoding times, the “speed problem” so far remains to be unsolved. Naturally, any original proposal, any new idea, leading to the improvement of the overall performance of fractal image compression technologies, is worthy of great praise.

This paper introduces a new idea (strategy) to speeding up image compression, i.e., to overcoming the “speed problem” in the block-based fractal image coding procedures. The proposed idea rests on the direct application of invariant image smoothness parameter values – image smoothness estimates. The latter estimates are used in stating the necessary image similarity condition, which is employed later on to achieve image compression speed gains in the search for optimal pairings “range block-domain block”. Theoretical and experimental analysis results show that the proposed strategy is comparatively simple in implementation and fast in encoding time, as compared with recently developed block based fractal compression schemes.

## 2 Image Smoothness Estimates and Their Properties

Consider a set of digital images  $S^2(n) = \{[X(m)] \mid m = (m_1, m_2) \in I^2\}$ , where:  $I = \{0, 1, \dots, N-1\}$ ,  $N = 2^n$ ,  $n \in \mathbb{N}$ ;  $X(m) \in \{0, 1, \dots, 2^p - 1\}$ , for all  $m \in I^2$ ;  $p$  ( $p \geq 1$ ) equals the number of bits per pixel in  $[X(m)]$ . The distance (mean squared error)  $\delta$  between any two elements of the set  $S^2(n)$  - images  $[X_1(m)]$  and  $[X_2(m)]$  - is specified by.

$$\delta = \delta(X_1, X_2) = \left( \frac{1}{N^2} \sum_{m \in I^2} (X_2(m) - X_1(m))^2 \right)^{1/2}. \quad (1)$$

Let us denote the two-dimensional discrete spectrum (Walsh-Hadamard (WHT), cosine (DCT), etc., [9]) of the image  $[X(m)] \in S^2(n)$  by  $[Y_X(k)]$ ,  $k = (k_1, k_2) \in I^2$ . It is well known that the spectral coefficients  $Y_X(k)$  decrease in absolute value, as their serial numbers  $k$  (indices  $k_1$  and  $k_2$ ) increase, provided the basis vectors of the discrete transform in use are presented in a frequency order. The latter circumstance implies that there exists a hyperbolic surface

$$z = z(x_1, x_2) = C / (x_1 \cdot x_2)^\alpha \quad (C \geq 0, \alpha \geq 0), \quad (2)$$

which approximates the ordered array of spectral coefficients  $\{|Y_X(k)| \mid k = (k_1, k_2) \in I^2, k_1^2 + k_2^2 \neq 0\}$  in the mean squared error sense, i.e.,

$$\delta = \delta(Y_X, z) = \left( \frac{1}{N^2 - 1} \sum_{\substack{k \in I^2 \\ (k_1^2 + k_2^2 \neq 0)}} \left( |Y_X(k)| - \frac{C}{(\bar{k}_1 \cdot \bar{k}_2)^\alpha} \right)^2 \right)^{1/2} \rightarrow \min; \quad (3)$$

here  $\bar{k}_i = \max\{k_i, 1\}$ ,  $i = 1, 2$ .

The quantity  $\alpha$  (expression (2)), characterizing the shape of the hyperbolic surface, i.e., the rate of decay of spectral coefficients (high frequency components of the image), as their serial numbers increase, is assumed, in what follows, to be the smoothness parameter (level, class) of the image  $[X(m)] \in S^2(n)$ . This assumption is intuitively understandable – the more intense manifestation of high frequency components in the discrete spectrum of the image, the more noticeable changes of pixel intensity values (sharp edges) are detected in the image.

Below, we present a means for finding the very first approximation of the image smoothness parameter value  $\alpha_0$ . Let us designate the set of indices of nonzero spectral coefficients in the discrete spectrum  $[Y_X(k)]$  of the image  $[X(m)] \in S^2(n)$  as  $H$ , i.e.,

$$H = \{k = (k_1, k_2) \in I^2 \mid Y_X(k) \neq 0, k_1^2 + k_2^2 \neq 0\}. \quad (4)$$

Then, application of the “linearization” procedure (logarithmization) both to the ordered array of nonzero spectral coefficients  $\{|Y_X(k)| \mid k = (k_1, k_2) \in H\}$  and to the hyperbolic surface  $z = C / (x_1 \cdot x_2)^\alpha$  leads to the following result (objective function)

$$\delta = \delta(\ln|Y_X|, \ln z) = \left( \frac{1}{|H|} \sum_{k \in H} (\ln|Y_X(k_1, k_2)| - \ln C + \alpha \ln(\bar{k}_1 \cdot \bar{k}_2))^2 \right)^{1/2}. \quad (5)$$

The minimum of the latter function is found using the least squares method (for the sake of simplicity,  $\delta$  is squared), namely:

$$\begin{cases} \frac{\partial \delta^2}{\partial \ln C} = -\frac{2}{|H|} \sum_{k \in H} (\ln|Y_X(k_1, k_2)| - \ln C + \alpha \ln(\bar{k}_1 \cdot \bar{k}_2)) = 0, \\ \frac{\partial \delta^2}{\partial \alpha} = -\frac{2}{|H|} \sum_{k \in H} (\ln|Y_X(k_1, k_2)| - \ln C + \alpha \ln(\bar{k}_1 \cdot \bar{k}_2)) \cdot \ln(\bar{k}_1 \cdot \bar{k}_2) = 0. \end{cases} \quad (6)$$

Now, solving this system of linear algebraic equations (expression (6)) for  $\alpha$ , we easily derive

$$\alpha = \alpha_0 = \frac{1}{A_N} \sum_{k \in H} (B_N - |H| \cdot P(k)) \cdot \log |Y_X(k)|, \quad (7)$$

where:  $A_N = |H| \cdot C_N - B_N^2$ ;  $B_N = \sum_{k \in H} P(k)$ ;  $C_N = \sum_{k \in H} P^2(k)$ ;  $P(k) = \log (\bar{k}_1 \cdot \bar{k}_2)$ ,

for all  $k \in H$ ; by the way,  $A_N = 0$  if and only if the set  $H$  is empty, i.e., the digital image  $[X(m)]$  is absolutely smooth. It is worth emphasizing that the above “rough” image smoothness estimates (expression (7)), sometimes, serve the purpose.

To make the estimate more precise, various approaches can be applied, namely: successive coordinate optimization procedures, special iterative techniques, etc., [12]. Experimental results show that the real world image smoothness estimates, obtained using DCT, fall into the interval  $(0; 3)$ .

We have proved that the image smoothness parameter values (smoothness estimates)  $\alpha$  for  $[X(m)] \in S^2(n)$ , found using DCT (or, WHT), possess the following exceptionally important properties, [12]:

1. Invariance of  $\alpha$  with respect to some transformations (rotation, reflection, inversion, luminance change), acting upon the image  $[X(m)]$ .
2. Continuity of  $\alpha: S^2(n) \rightarrow \mathbb{R}$ , taken in the way that small (discrete) changes in  $[X(m)]$  correspond to small (discrete) changes in  $\alpha$ .

Just the latter property makes it possible to introduce the necessary image similarity condition – two digital images  $[X_1(m)] \in S^2(n)$  and  $[X_2(m)] \in S^2(n)$  can not be similar if their smoothness parameter values (smoothness estimates  $\alpha_{X_1}$  and  $\alpha_{X_2}$ , respectively) differ (in the sense of  $\delta$ ) distinctly, i.e.,

$$(|\alpha_{X_1} - \alpha_{X_2}| > \varepsilon_0) \Rightarrow (\delta(X_1, X_2) > \delta_0); \quad (8)$$

here:  $\delta_0$  is a small positive number (the threshold criterion value). For task-oriented applications, the relationship  $\varepsilon_0 \leftrightarrow \delta_0$ , evidently, should be established experimentally.

### 3 Speeding Up Fractal Image Compression

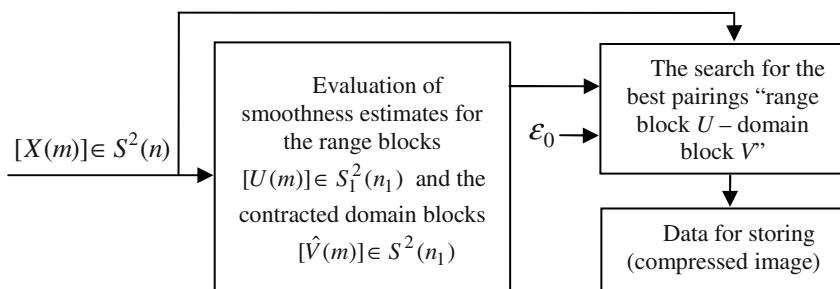
Since A. Jacquin described the first practical block based fractal image coding scheme, [2], fractal image coding technique has attracted a lot of attention as a new digital image processing technology. However, the encoding complexity of the fractal image coding is very high and has become the major obstacle for its workable applications. It is well known that the most computationally intensive part of the fractal encoding (compression) process is the searching step. This is required for each range block to find the best matched domain block within the searching region (pool). If the brute force strategy (full search; Jacquin's approach) is used, computation, needed to perform fractal encoding, is enormous (“speed problem”). Over the past

years, many fast searching algorithms have been developed to surmount the “speed problem”, [5, 10, 11].

We here present a new attempt (idea, strategy) to improve compression times in block based fractal image coding procedures (Figure 1). Let  $[X(m)] \in S^2(n)$  be an image to be processed. Fractal image compression speed gains are achieved, mainly, owing to the following factors:

1. Firstly, for the determination of smoothness level, each range block  $[U_i(m)] \in S_1^2(n_1) \subset S^2(n_1)$ , as well as each domain block  $[V_j(m)] \in S_1^2(n_2) \subset S^2(n_2)$ , is looked over only once (here:  $i = 1, 2, \dots, 4^{n-n_1}$ ;  $j = 1, 2, \dots, (2^n - 2^{n_2} + 1)^2$ ; usually,  $n_1 \in \{2, 3\}$  and  $n_2 = n_1 + 1$ ). As a result, two sequences of image smoothness estimates,  $\{\alpha_{U_i}\}$  and  $\{\alpha_{\hat{V}_j}\}$ , are formed ( $[\hat{V}_j(m)] \in S^2(n_1)$  is a shrunken copy of the domain block  $[V_j(m)]$ ). Thus, domain and range blocks are categorized into a finite number of classes, according to their invariant representations – smoothness parameter values (smoothness estimates).

Now, direct distance comparisons between the smoothness estimates make it possible to determine the best pairings “range block – domain block”. The search region (domain pool) for a particular range block  $[U(m)]$  is limited by the upper bound  $\varepsilon_0$  for the difference  $|\alpha_U - \alpha_{\hat{V}}|$  (the necessary image similarity condition; Section 2).



**Fig. 1.** Fractal image encoding scheme - implementation of the necessary image similarity condition

2. Secondly, a thorough analysis of algebraic expressions and actions used in evaluating the DCT spectrum of the range block (or, contracted domain block;  $n_1 = 3$ ), in finding image (block) smoothness estimates, as well as in establishing the fact of block similarity, shows that the total time expenditures ( $\tau$ ), associated with these steps, are equal to

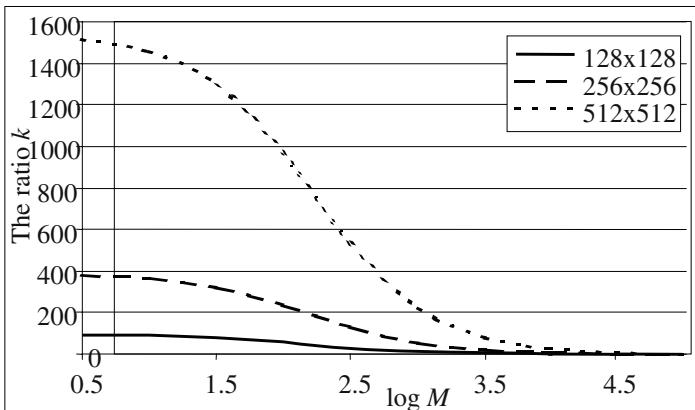
$$\tau = \left( \frac{9709}{64} N^2 + 10029(N-15)^2 + \frac{553}{8} N^2 M \right) \cdot \tau_a, \quad (9)$$

provided the time expenditures, required to perform a single addition and multiplication operations, are equal, i.e.,  $\tau_a = \tau_m$ ; here  $M$  indicates the averaged number of domain blocks  $[V(m)] \in S_1^2(4) \subset S^2(4)$  contained in the search region (pool) related to the range block  $[U(m)] \in S_1^2(3) \subset S^2(3)$ ;  $N \times N$  is the size of the image under processing.

Similarly, the total time expenditure ( $\tau^o$ ), associated with the brute force strategy (Jacquin's approach), equals

$$\tau^o = \left( \frac{65}{64} N^2 + 65(N-15)^2 + \frac{553}{8} N^2(N-15)^2 \right) \cdot \tau_a. \quad (10)$$

Now, fractal image compression (encoding) time savings can be expressed in terms of  $k = \tau^o / \tau$  (Figure 2). In particular, for  $N = 512$ , the ratio  $k$  exceeds 200, provided  $M \leq 10^3$ ; theoretically, for  $M \leq 10^4$ , "success" (compression time savings) is ensured unconditionally.

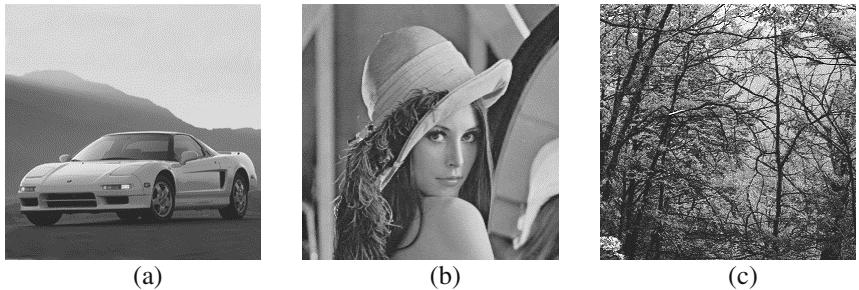


**Fig. 2.** Comparative analysis of fractal image encoding complexity ( $k = \tau^o / \tau$ )

But, what are the really acceptable values of  $M$ ? Experimental results show that those values, as well as the quality of restored images, depend on both the upper bound  $\varepsilon_0$  (expression (8); Section 2) and the smoothness level of range blocks.

## 4 Experimental Results

To corroborate the obtained theoretical results (Sections 2 and 3), some test images, characterized by different smoothness parameter values, were analysed (Figure 3). The range blocks were chosen to be of size 4x4, and the domain blocks – of size 8x8. Smoothness estimates were found using DCT.



**Fig. 3.** Test images: (a) image “Acura” 256x256,  $\alpha = 1.44$ ; (b) image “Lena” 256x256,  $\alpha = 0.69$ ; (c) image “Forest” 256x256,  $\alpha = 0.37$

Table 1 contains experimental results, associated with the averaged number  $M$  of domain blocks in a pool (expression (9); Section 3), for the test images “Acura”, “Lena” and “Forest” (Figure 3). It can be seen that the averaged pool size variations highly depend on the smoothness level of the image under processing.

Comparative analysis of two strategies – the full search for optimal pairings (brute force approach), characterized by the mean squared error  $\delta_{\text{opt}}$ , and the proposed search strategy, based on the use of image smoothness estimates and characterized by the mean squared error  $\delta_{\varepsilon_0}$ , is presented, also, in Table 1. Analysis results show, the higher smoothness of the image under processing, the better pairing results (in the sense of  $\delta_{\varepsilon_0}$ ) are obtained.

On the other hand, if the value  $\varepsilon_0$  (the upper bound for the difference  $|\alpha_U - \alpha_V|$ ) is chosen, say, to be not less than 0.1, then all the selected (best) pairings “range block – domain block” will appear to be sufficiently close to the optimal ones (Table 1).

The overall performance (compression speed gains) of the proposed fractal image encoding strategy is shown in Table 2 (Computer simulation was performed on a PC with CPU AMD1800+(@2800+), RAM 512MB, OS Windows XP).

**Table 1.** Dependence of the averaged pool size  $M$  and the averaged deviation  $\Delta\delta = \delta_{\varepsilon_0} - \delta_{\text{opt}}$  on  $\varepsilon_0$

The upper bound, $\varepsilon_0$	The averaged number of domain blocks in a pool, $M$			The averaged deviations, $\Delta\delta$		
	Image “Acura”	Image “Lena”	Image “Forest”	Image “Acura”	Image “Lena”	Image “Forest”
0.001	63	51	94	1.76021	2.54327	4.76107
0.005	340	294	817	0.96680	1.50931	3.14990
0.010	682	589	1721	0.71430	1.25823	2.55641
0.020	1385	1204	3527	0.57229	1.01700	2.03292
0.040	2787	2438	7101	0.42149	0.81976	1.57958
0.100	6966	6123	17340	0.25324	0.49752	0.89875
0.150	10343	9129	25088	0.18624	0.37377	0.60599

It can be seen, that quite tolerable (in the sense of  $\delta = \delta(X, \tilde{X})$ ; here  $[\tilde{X}(m)]$  is the restored image) processing results are obtained even for  $\varepsilon_0 \leq 0.005$  (images “Acura” and “Lena”; Table 2). In such cases, compression time savings are sufficiently good ( $k = \tau^0 / \tau > 18$ , for “Acura”, and  $k > 60$ , for “Lena”; Table 2).

**Table 2.** Fractal image compression speed gains and the quality of restored images, for different values of  $\varepsilon_0$

The upper bound, $\varepsilon_0$	The quality of restored images, $\delta = \delta(X, \tilde{X})$			The total time expenditure, $\tau$		
	Image “Acura”	Image “Lena”	Image “Forest”	Image “Acura”	Image “Lena”	Image “Forest”
Full search	2.41	3.80	17.88	547.06	550.38	546.98
0.200	2.87	4.27	19.42	330.95	213.65	164.13
0.100	3.03	4.44	20.22	242.50	121.59	84.19
0.050	3.27	4.64	21.12	164.81	66.34	43.05
0.013	3.68	5.09	23.00	67.53	19.48	12.39
0.005	4.02	5.79	24.38	30.39	8.61	5.65
0.001	5.20	7.23	27.57	8.69	3.11	2.31

More impressive fractal image compression speed gains are obtained for images of size 512x512 (Figure 4; image “Maroon”). For instance, the upper bound value  $\varepsilon_0 = 0.005$  ensures very high compression time savings ( $k > 145$ ) and, above all, the quality of the restored image  $[\tilde{X}(m)]$  remains to be sufficiently good (Figure 4, b, e).

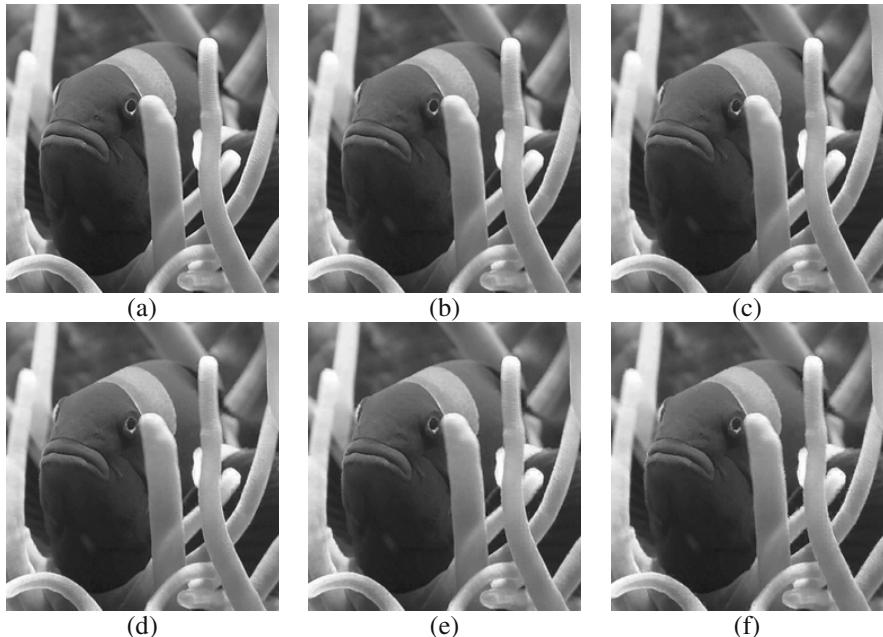
## 5 Conclusion

Theoretical and experimental analysis results confirm usefulness of the proposed image compression time accelerating approach, based on the use of image smoothness estimates. Compression time savings are achieved, mainly, owing to the following two factors: firstly, for the determination of the level of smoothness, each domain block, as well as each range block, is looked over only once; secondly, candidate domain blocks (forming a pool) and corresponding range block, roughly speaking, fall into the same class of smoothness. So, to ensure fast search for best pairings, it is quite enough to analyze only a small number of pairs “range block-domain block”. The quality of restored images satisfies the needs too.

In spite of the fact that fractal image coding techniques (the basic Jacquin’s idea and plenty of its modifications), based on the use of partitioned iterated function systems and distinguishing themselves by a clear disproportion of time expenditures, needed to perform image encoding and image decoding steps, cannot (till now) compete with the most widely used in practice the still image compression standard

JPEG (basically, at lower compression ratios), any attempt to promote (modify, improve) the very promising fractal image processing technologies, is worthy of great praise.

Future research we are to concentrate on two-dimensional binary (white-and-black) images, on the determination of their smoothness parameter values and on the development of appropriate fractal image coding schemes.



**Fig. 4.** Fractal image compression speed gains (range blocks 8x8, domain blocks 16x16): (a) image „Maroon“ 512x512; (b) full search ( $\tau = 6132.14$  sec,  $\delta = 4.31$ ); (c)  $\varepsilon_0 = 0.025$ ,  $\tau = 172.72$  sec,  $\delta = 4.90$ ; (d)  $\varepsilon_0 = 0.013$ ,  $\tau = 94.59$  sec,  $\delta = 5.09$ ; (e)  $\varepsilon_0 = 0.005$ ,  $\tau = 42.16$  sec,  $\delta = 5.39$ ; (f)  $\varepsilon_0 = 0.001$ ,  $\tau = 17.95$  sec,  $\delta = 6.24$

## References

1. Peitgen, H.-O., Jurgens, H., Saupe, D.: *Chaos and Fractals*. Springer-Verlag (1992)
2. Jacquin, A.: Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. *IEEE Transactions on Image Processing*, Vol. 1, no. 1 (1992) 18-30
3. Turner, M. J., Blackledge, J. M., Andrews, P. R.: *Fractal Geometry in Digital Imaging*. Academic Press, Cambridge (1998)
4. Valantinas, J., Zumbakis, T.: On the Use of Shift Dynamics in Synthesizing Fractal Images. *Intern. Journ. INFORMATICA*, Vol. 15, no. 3. Institute of Mathematics and Informatics, Vilnius (2004) 411-424
5. Fisher, Y.: *Fractal Image Compression – Theory and Application*. Springer-Verlag New York (1994)

6. Wallace, G.K. : The JPEG Still Picture Compression Standard. Communications of the ACM, Vol. 34, no. 4 (1991) 30-44
7. Culik II, K., Valenta, V.: Finite Automata Based Compression of Bi-Level and Simple Color Images. Department of Computer Science, University of South Carolina, Columbia , S.C. 29208, U.S.A (1998) 1-14
8. Valantinas, J., Valantinas, R.: Problem-Oriented Change of Image Dimensionality. Proceedings of the Third International Symposium on Image and Signal Processing and Analysis, Rome (Italy), Universita degli Studi ROMA TRE (2003) 228-232
9. Ahmed, N., Rao, K.R., Orthogonal Transforms for Digital Signal Processing. Springer-Verlag, Berlin Heidelberg New York (1975)
10. Saupe, D., Hamzaoui, R.: Complexity reduction methods for fractal image compression. Proceedings of the IMA Conference on Image Processing: Mathematical Methods and Applications, Oxford, England (1994) 211-229
11. Wohlberg, B., de Jager, G.: A review of the Fractal Image Coding Literature. IEEE Transactions on Image Processing, Vol. 8, no. 12 (1999) 1716-1729
12. Valantinas, J., Žumbakis, T.: Definition, evaluation and task-oriented application of image smoothness estimates. Information Technology and Control, no. 1(14), Technologija, Kaunas (2004) 15 – 24

# DCT Based High Quality Image Compression

Nikolay Ponomarenko<sup>1</sup>, Vladimir Lukin<sup>1</sup>,  
Karen Egiazarian<sup>2</sup>, and Jaakko Astola<sup>2</sup>

<sup>1</sup> Department 504, National Aerospace University  
(Kharkov Aviation Institute),  
17 Chkalova Street, 61070, Kharkov, Ukraine  
[lukin@xai.kharkov.ua](mailto:lukin@xai.kharkov.ua)

<sup>2</sup> Tampere International Center for Signal Processing,  
Tampere University of Technology,  
P.O.Box-553, FIN-33101, Tampere, Finland  
[{karen, jta}@cs.tut.fi](mailto:{karen, jta}@cs.tut.fi)

**Abstract.** DCT based image compression using blocks of size 32x32 is considered. An effective method of bit-plane coding of quantized DCT coefficients is proposed. Parameters of post-filtering for removing of blocking artifacts in decoded images are given. The efficiency of the proposed method for test images compression is analyzed. It is shown that the proposed method is able to provide the quality of decoding images higher than for JPEG2000 by up to 1.9 dB.

## 1 Introduction

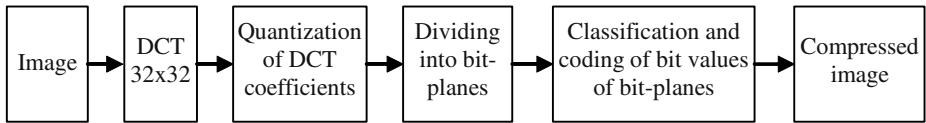
Discrete cosine transform (DCT) [1,2] is the basis of many image compression methods. For example, the standard JPEG [3,4], for which DCT is carried out in 8x8 image blocks existed as the main image compression standard for about 10 years. However, many investigations and achieved progress in this area have dealt with applications of discrete wavelet transform (DWT). For example, the compression standard JPEG2000 [5,6] accepted quite recently is based on DWT and it commonly provides considerably better quality of decoded images than JPEG.

Aforesaid allows supposing that DWT is more appropriate transform for applying in image compression than DCT. In this paper we try to show that this is not true. Due to rather simple improvements of the base method (used in JPEG) it is possible to obtain decoded images quality better than for JPEG2000.

There are three basic modifications introduced by us compared to JPEG. First, an image is divided into 32x32 pixel blocks instead of 8x8 for conventional JPEG. Second, the quantized DCT coefficients are divided into bit-planes; the bit values are coded according to complex probability models that take into account the presence of correlation between values of neighbor coefficients in blocks and between the values of the corresponding coefficients of neighbor blocks. Third, DCT based filtering [7] is used as post-processing for removal of blocking artifacts from decoded images and, thus, for increasing decoded image quality.

## 2 Coding and Decoding Schemes

The block-diagram of image coding for the proposed approach is depicted in Fig. 1.

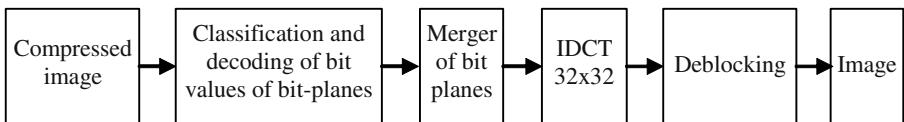


**Fig. 1.** The block-diagram of image coding

An image to be compressed is divided into 32x32 pixel blocks. Then, DCT for pixel values of each block is computed. After this, the quantization of DCT coefficients of image blocks is carried out. At this stage the basic losses are introduced into compressed image. Larger quantization step (QS) provides larger compression ratio (CR) and simultaneously it leads to larger losses. In this paper it is proposed to use uniform quantization that ensures the best results within the structure of the considered method.

Then, the division of quantized DCT coefficients into bit-planes is carried out. The obtained bit-planes are coded in the order starting from higher bits to lower ones. While coding each next plane, the values of bits of earlier coded planes are taken into account. A coded bit is referred to one or another group of bits according to the values of already coded bits. For each group of bits, individual probability model is used for dynamic arithmetic coding (see Section 3).

The block-diagram of image decoding is presented in Fig. 2 where IDCT denotes inverse DCT.



**Fig. 2.** The block-diagram of image decoding

As seen, at image decoding stage all steps are repeated in reverse order. Besides, at the final step the operation of decoded image filtering is added (see Section 4 for more details).

## 3 Bit-Plane Coding

Thus, after calculation of DCT in 32x32 blocks and quantization of obtained coefficients, we have an array of integer valued DCT coefficients. Divide the array of absolute values of DCT coefficients into  $n$  bit-planes, where  $n$  is the number of the highest bit-plane in which there are non-zero values. Coding begins with the bit-plane  $n$  and comes to an end by bit-plane 1.

The signs of non-zero DCT coefficients are practically random variables with approximately equal probabilities. Therefore, they are allocated into a separate array (one bit for each sign) and transferred to the output stream at once.

Let  $P_{l,m}^k(i,j)$  defines a bit value of a bit-plane  $k$  of a coefficient with the index  $i,j$  of the block of an image with the index  $l, m$ , where  $k=1..n, i,j=1..32, l=1..L, m=1..M$ ,  $L,M$  denotes the number of image blocks for vertical and horizontal directions. We introduce the following conditions which are used for classification of bits of bit-planes:

1)  $C1(k,l,m,i,j)=true$ , if  $1 \in \{P_{l,m}^{k+1}(i,j), \dots, P_{l,m}^n(i,j)\}$ . This condition is assigned *true* if, at least, one bit among earlier coded higher bit planes is equal to 1.

2)  $C2(k,l,m,i,j)=true$ , if  $1 \in \{P_{l,m}^{k+2}(i,j), \dots, P_{l,m}^n(i,j)\}$ . This condition is *true* if, without taking into account the previously coded higher bit-plane, the bit with these indices was equal to 1. If the condition  $C2$  is *true* then the current bit with approximately equal probability can be equal either to 0 or to 1. If the condition  $C1$  is *true* and the condition  $C2$  is *false* then the probability of zero for the current bit is considerably larger than the probability to be equal to 1.

3)  $C3(k,l,m,i,j)=true$ , if  $1 \in \{P_{l,m}^k(i,j), \dots, P_{l,m}^n(i,j)\}$ . This condition is *true* if in this or in, at least, one of earlier coded higher bit planes the bit with these indices was equal to 1. The condition  $C3$  can be checked for those bits neighboring the coded bit that till the current moment have been already coded. Here and below only the values of those bits can be checked that have been already coded. This is important for providing an opportunity of decoding. At decoding stage those bits that have been coded earlier are decoded earlier as well and they can be checked in conditions.

4)  $C4(k,l,m,i,j)=true$ , if  $P_{l,m}^{k+1}(i,j)=1$ . This condition is *true* if in the previously coded bit plane the bit with these indices was equal to 1.

5)  $C5(k,l,m,i,j)=true$ , if  $P_{l,m}^k(i,j)=1$ .

6)  $C6(k,l,m,i,j)=true$ , if  $true \in \{C1(k,l,m,i-1,j-1), C1(k,l,m,i-1,j), C1(k,l,m,i-1,j+1), C1(k,l,m,i,j-1), C1(k,l,m,i,j+1), C1(k,l,m,i+1,j-1), C1(k,l,m,i+1,j), C1(k,l,m,i+1,j+1)\}$ . This condition is *true* if for, at least, one of neighboring bits there is unity in higher bit planes.

7)  $C7(k,l,m,i,j)=true$ , if  $true \in \{C5(k,l,m,i-1,j-1), C5(k,l,m,i-1,j), C5(k,l,m,i-1,j+1), C5(k,l,m,i,j-1)\}$ . This condition is *true* if, at least, one among neighboring and already coded bits of this bit-plane was equal to 1.

8)  $C8(k,l,m,i,j)=true$ , if  $true \in \{C3(k,l,m,i-2,j-2), C3(k,l,m,i-2,j-1), C3(k,l,m,i-2,j), C3(k,l,m,i-2,j+1), C3(k,l,m,i-1,j-2), C3(k,l,m,i-1,j+2), C3(k,l,m,i,j-2), C3(k,l,m,i,j+2), C3(k,l,m,i+1,j-2), C3(k,l,m,i+1,j+2), C3(k,l,m,i+2,j-2), C3(k,l,m,i+2,j-1), C3(k,l,m,i+2,j), C3(k,l,m,i+2,j+1), C3(k,l,m,i+2,j+2)\}$ . This condition is *true* if there was, at least, one unity in this or higher bit planes for already coded bits displaced from the coded bit by 2 rows or 2 columns.

9)  $C9(k,l,m,i,j)=true$ , if  $true \in \{C3(k,l,m,i-3,j-3), C3(k,l,m,i-3,j-2), C3(k,l,m,i-3,j-1), C3(k,l,m,i-3,j), C3(k,l,m,i-3,j+1), C3(k,l,m,i-3,j+2), C3(k,l,m,i-3,j+3), C3(k,l,m,i-2,j-3), C3(k,l,m,i-2,j+3), C3(k,l,m,i-1,j-3), C3(k,l,m,i-1,j+3), C3(k,l,m,i,j-3), C3(k,l,m,i,j+3), C3(k,l,m,i+1,j-3), C3(k,l,m,i+1,j+3), C3(k,l,m,i+2,j-3), C3(k,l,m,i+2,j+3), C3(k,l,m,i+3,j-3), C3(k,l,m,i+3,j-2), C3(k,l,m,i+3,j-1), C3(k,l,m,i+3,j), C3(k,l,m,i+3,j+1), C3(k,l,m,i+3,j+2), C3(k,l,m,i+3,j+3)\}$ . This condition is *true* if there was unity in this or higher bit planes for already coded bits displaced from the coded bit by 3 rows or 3 columns.

10)  $C10(k,l,m,i,j)=\text{true}$ , if  $\text{true} \in \{C3(k,l-1,m-1,i,j), C3(k,l-1,m,i,j), C3(k,l-1,m+1,i,j), C3(k,l,m-1,i,j), C3(k,l,m+1,i,j), C3(k,l+1,m-1,i,j), C3(k,l+1,m,i,j), C3(k,l+1,m+1,i,j)\}$ . This condition is true if there was unity in this or in higher bit planes for bits in neighbor blocks. This condition allows taking into consideration correlation for bits having identical indices and belonging to image neighbor blocks.

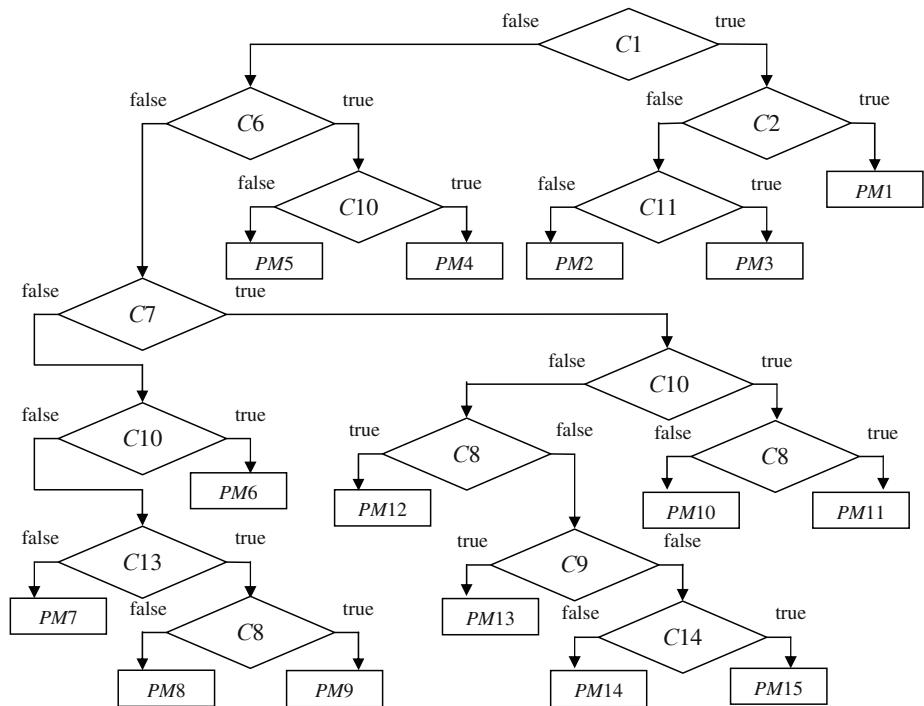
11)  $C11(k,l,m,i,j)=\text{true}$ , if  $(C2(k,l,m,i,j)=\text{false}) \text{and} (C6(k+1,l,m,i,j)=\text{false})$ . The checking of this condition allows classifying more reliably the bit for which in the previously coded bit plane there was unity.

$$12) C12(k,l,m,i,j) = \begin{cases} 1, & C5(k,l,m,i,j) = \text{true} \\ 0, & C5(k,l,m,i,j) = \text{false} \end{cases}.$$

13)  $C13(k,l,m,i,j)=\text{true}$ , if  $1 = C12(k,l,m,i-1,j-1) + C12(k,l,m,i-1,j) + C12(k,l,m,i-1,j+1) + C12(k,l,m,i,j-1)$ .

14)  $C14(k,l,m,i,j)=\text{true}$ , if  $k=1$ .

Fig. 3 presents the flowchart of bit value classification by checking the aforementioned conditions ( $PMX$  - probability model number X).



**Fig. 3.** Flow chart of classification of coded bits of bit-planes

Totally according to given classification a bit can be referred to one of fifteen probability models. For each model after coding the current bit the counters of 0 and 1 are corrected, and they are used for coding next bits referred to this model. For coding it

is proposed to use the dynamic version of arithmetic coding [8,9] that is the most effective for the considered case.

Let us consider the obtained classification more in detail. To *PM1* and *PM2* those bits are referred that for higher bit planes had unities and for which the probabilities of being equal to 0 and 1 are practically equal. To *PM3* those bits are referred for which there was unity in previously coded plane and the probability of unity was low. Because of this, for *PM3* the probability of 0 is larger than being equal to 1.

To the model *PM4* those bits are referred that have no unities in higher bit-planes but there are neighbor bits with unities in higher bit planes and there are unities in the corresponding bits of image neighbor blocks. For *PM4* the probabilities of 1 and 0 are rather close, and the bits referred to this model are compressed poorly. The difference between the models *PM4* and *PM5* consists in the following. To *PM5* those bits are referred that have no unities in the corresponding bits of image neighbor blocks. For the model *PM5* there are considerably more zeros than unities, and the corresponding bits are compressed considerably better.

The bits of the models *PM6-PM9* differ from the bits of the models *PM4* and *PM5*. For the former ones there are no neighbor bits with unities in higher bit planes, but there are unities for neighbor bits in the current coded plane. For the models *PM6-PM9* the probability of unities is considerably smaller than for the models *PM4* and *PM5*. Because of this, these data are compressed better. Those bits are referred to the model *PM6* that have unities in the corresponding bits of image neighbor blocks. The bits of this model are compressed in the worst way among the bits that belong to the group of the models *PM6-PM9*. For the models *PM7-PM9* there are no unities in the corresponding bits of image neighbor blocks. For the bits referred to the model *PM7* the number of unities in the neighbor bits is larger than 1. For the models *PM8* and *PM9* there is only one unity in the neighbor bits. Because of this, the probability of unities for them is even smaller than for the model *PM7*. Division of bits between the models *PM8* and *PM9* is accomplished using the condition *C8* that allows taking into account the presence of unities in the bits displaced from the coded one by 2 rows and 2 columns. Due to this, the bits of the model *PM8* for which *C8=false* are compressed best of all among the bits of the models *PM6-PM9*.

The bits of the models *PM10-PM15* differ from the bits of the models *PM4-PM9* by the following. For the former ones there are no unities either in higher bit planes or in the current coded plane. The bits of the models *PM10-PM15* are compressed very well, however, additional division of such bits into several models lead to considerable increasing of CR. Those bits are referred to the models *PM10*, *PM11* that have unity in the corresponding bits of image neighbor blocks. The bits of the model *PM10* are compressed slightly better since for them there are no unities in the bits displaced from the coded one by 2 rows and 2 columns. For the model *PM13*, there are unities only in bits displaced from the coded one by 3 rows and 3 columns. For the models *PM14* and *PM15* there are no unities in the checked area. Such bits are compressed in the best way (most efficiently). The difference between these models consists in the following. To the model *PM15* those bits are referred that belong to the lowest bit-plane ( $k=1$ ). We propose to avoid coding the bits of the model *PM15* (they all are

considered equal to 0). This is analogous to «dead zone» in quantization. But in our case, this occurs to be effective due to selectivity of its application.

Before starting coding each bit plane, the counters of unities and zeros for the models *PM1-PM14* are initialized as unities, i.e. the models of each coded plane are independent. Different copies of the models *PM1-PM14* are used for different regions of image blocks. For the bits of DCT coefficient with the indices  $i=1, j=1$  (this is the quantized value of the block mean) a separate copy of the models *PM1-PM14* is used. The statistical characteristics of this DCT coefficient considerably differ from statistics of other DCT coefficients. A separate copy of the models *PM1-PM14* is also used for the first (upper) row of block DCT coefficients. This is explained by the fact that for these coefficients there is only one earlier coded bit. This leads to considerable difference of bit distribution between the models. For all other DCT coefficients of a block (and they are the basic amount of data) the third copy of the models *PM1-PM14* is used.

The proposed classification is obtained by experimental studies of efficiency of various ways to divide bits into classes for different test images and QS. Probably, more effective variant of such classification can be found. In practice, simpler variants of classification can be used in order to increase coding speed. For example, the absence of checking the condition *C10* (in this case one does not take into account the correlation between neighbor blocks of an image) results in increasing the size of compressed image by 1-3 %.

If one does not check the condition *C9* (this condition deals with correlation of bits displaced from the coded one by 3 rows and 3 columns) the size of coded image increases by 1-1.5%. If one also does not check the condition *C8* (this condition deals with correlation of bits displaced from the coded one by 2 rows and 2 columns), this leads to the increasing of coded image size by 3-7%.

Let us mention one important point once again. For the used variant that includes the *PM15*, the losses of image quality occur not only at DCT coefficient quantization step, but also (though in much smaller degree), at the step of bit values coding for bit-planes. If one avoids using the model *PM15* this does not lead to any additional losses at this step.

## 4 Filtering for Removal of Blocking Artifact

For blocking effect reduction in decoded images, we employ an approach described in [7]. This approach presumes the use of DCT based filter for additive noise removal [10]. In the considered case, the noise to be removed is the quantization noise. The size of a sliding window of the DCT based filter is 8x8. For each position of the sliding window, DCT is carried out, then DCT coefficients having absolute values smaller than preset threshold are assigned zero values (hard thresholding). After this, inverse DCT is executed.

Spatially invariant denoising is employed. One problem is the setting of the threshold. For our application we recommend to set the threshold equal to  $QS/2$  (note that we know  $QS$  a priori).

The use of post-filtering in our case allows increasing quality of the decoded images by 0.5-1 dB. The decoding time increases by 30-40 %.

## 5 Numerical Simulations

The quality of compression for the proposed method was analyzed for 512x512 gray-scale images in comparison to JPEG2000 (Kakadu coder by D.Taubman [6] has been employed). The practical realization of our method in programming language Delphi (the coder has the name AGU) is accessible to downloading from the address <http://www.cs.tut.fi/~karen/agucoder.htm>. This version is intended for coding only 512x512 grayscale images in RAW format (without heading). The used set of test images is accessible to downloading from the same address.

The quality of decoded images was compared for CRs equal 8, 16, 32 and 64. As quality criterion, the peak signal to noise ratio was used:

$$PSNR = 10 \lg(255^2 / [\sum_{i=1}^I \sum_{j=1}^J (I_{ij} - I_{ij}^e)^2 / IJ]),$$

where  $I, J$  denote the image size,  $I_{ij}^e$  is

the value of the ij-th pixel of original image, and  $I_{ij}$  defines the ij-th pixel value for the analyzed (decompressed) image. Table 1 presents the obtained  $PSNR$ s for the considered methods.

**Table 1.** The quality of the test image compression for JPEG2000 and AGU,  $PSNR$ , dB

Image	CR=8		CR=16		CR=32		CR=64	
	JPEG2000	AGU	JPEG2000	AGU	JPEG2000	AGU	JPEG2000	AGU
Lenna	40.33	40.52	37.27	37.46	34.15	34.51	31.02	31.50
Barbara	38.07	39.26	32.87	34.65	28.89	30.77	25.87	27.55
Baboon	29.11	29.70	25.57	26.12	23.18	23.69	21.68	22.01
Goldhill	36.54	37.03	33.24	33.65	30.53	31.09	28.49	28.97
Peppers	38.17	38.33	35.80	35.55	33.54	33.32	30.79	30.90

As seen from data presented in Table 1, in overwhelming majority of the considered situations AGU outperforms JPEG2000 by quality of the decoded images. The only exceptions are CR=16 and CR=32 for the image Peppers for which JPEG2000 provides PSNRs that are better than for AGU by 0.2-0.25 dB. At the same time, for more complex images like Baboon and Goldhill the benefit of AGU for all CRs is 0.3-0.6 dB. And for the image Barbara that differs from other images by the presence of a large number of textural regions the advantage of AGU is 1.2-1.9 dB.

Image compression performance can be also compared for identical quality of decompressed images. For example, for  $PSNR=28.89$  dB JPEG2000 compresses the image Barbara by 32 times while AGU compresses this image by 46.7 times, that is by 1.46 times better. For  $PSNR=25.87$  dB AGU compresses this image by 101.5 times, that is 1.59 times better than JPEG2000 for which CR=64.

The presented data confirm that the proposed method outperforms JPEG2000 in image coding quality. The smoother is the image, the less difference of coding quality is observed for JPEG2000 and AGU. And the more complex and textural is the image, the difference of coding quality is larger.

The fragment of decoded image Barbara for JPEG2000 and AGU is shown in Fig. 4.



**Fig. 4.** A fragment of the decoded image Barbara, CR=32 a) JPEG2000,  $PSNR=28.89$  dB  
b) AGU,  $PSNR=30.77$  dB

## 6 Conclusions

The carried out studies show that the method proposed and described in this paper provides better quality of decoded images than JPEG2000 in most of practical situations. And its superiority for complex textured images in some cases can reach 1.9 dB.

The proposed method is obtained by rather simple modifications of JPEG, in which DCT serves as its core. This indicates that DCT is at least not worse transformation for use in image compression than DWT used as the basis of JPEG2000.

For software realizations of AGU (not optimized), the required computation time is by about 15-20 times larger than for standard JPEG. The ways to speed up AGU can be studied in future. In particular, algorithms of fast integer valued approximation of DCT in 32x32 blocks seem to lead to considerable decreasing of computation time.

In future it is possible to consider the use of partition schemes [11] that make image compression methods more adaptive. Besides, a perspective direction is the use of DCT based image compression methods directed on reduction of blocking effect such as lapped orthogonal transforms [12, 13].

## References

1. Ahmed, N., Natarajan T., Rao K. R.: Discrete cosine transform. In: IEEE Transactions on Computers, Vol. 23, (1974) 90-93
2. Rao, K., Yip P.: Discrete Cosine Transform, Algorithms, Advantages, Applications. In: Academic Press (1990)
3. Wallace, G. K.: The JPEG Still Picture Compression Standard. In: Comm. Of the ACM, Vol. 34, No.4 (1991)
4. Pennebaker, W. B., Mitchell, J. L.: JPEG Still Image Data Compression Standard. In: Van Nostrand Reinhold, New York (1993)
5. Christopoulos, C., Skodras, A., Ebrahimi, T.: The JPEG2000 still image coding system: an overview. In: IEEE Trans. on Consumer Electronics, Vol. 46, Issue: 4 (2000) 1103-1127
6. Taubman, D., Marcellin, M.: JPEG 2000: Image Compression Fundamentals, Standards and Practice. In: Boston: Kluwer (2002)
7. Egiazarian, K., Helsingius, M., Kuosmanen, P., Astola, J.: Removal of blocking and ringing artifacts using transform domain denoising. In: Proc. of ISCAS'99, Vol. 4, (1999) 139-142
8. Rissanen, J.: Generalized kraft inequality and arithmetic coding. In: IBM J. Res. Develop., Vol. 20, (1976) 198-203
9. Langdon, G.G., Rissanen, J.J.: A simple general binary source code. In: IEEE Trans. Inf. Theory, IT-28 (1982) 800-803
10. Yaroslavsky, L.: Local Adaptive Filtering in Transform Domain for Image Restoration, Enhancement and Target Location. In: 6th Int. Workshop on Digital Image Processing and Computer Graphics (DIP-97), SPIE volume 3346, (1997) 2-17
11. Ponomarenko N., Lukin V., Egiazarian K., Astola J.: Partition Schemes in DCT Based Image Compression. In: Technical Report 3-2002, ISBN 952-15-0811-6, Tampere University of Technology, Finland, (2002)
12. Malvar, H.S., Staelin, D.H.: The LOT: transform coding without blocking effects. In: IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, (1989) 553-559
13. Tran, T.D., de Queiroz, R., Nguyen, T.Q.: The generalized lapped biorthogonal transform. In: Proceedings of the ICASSP'98, Vol.3, (1998) 1441-1444

# Optimal Encoding of Vector Data with Polygonal Approximation and Vertex Quantization

Alexander Kolesnikov

Speech and Image Processing Unit  
Department of Computer Science, University of Joensuu  
P.O. Box. 111, 80101 Joensuu, Finland  
koles@cs.joensuu.fi

**Abstract.** Problem of lossy compression of vector data is considered. We attack the problem by jointly considering data reduction by polygonal approximation and quantization of the prediction errors for approximation nodes. Optimal algorithms proposed for vector data encoding with minimal distortion for given target bit-rate, and with minimal bit-rate for given maximum deviation.

## 1 Introduction

We consider the problem of vector data compression, which is important for vector map storage and transmission to end-user devices, such as desktop, notebook or PDA computers. The problem has been addressed in two complementary ways: vector data reduction by polygonal approximation [2], [7], [8], [9], [11], [13], [16], [17], [18] [20], and quantization of vector data [3], [4], [12], [14], [19]. We propose a rate-distortion optimal algorithm for vector data encoding, which combines polygonal approximation of vector data and quantization of approximation nodes.

Originally the problem of optimal polygonal approximation was formulated in two basic forms [6]:

- Min- $\epsilon$  problem:* Given an open  $N$ -vertex polygonal curve  $P$ , approximate it by another polygonal curve  $S$  with a given number of segments  $M$  so that the *approximation error*  $E(P)$  is minimized.
- Min-# problem:* Given an open  $N$ -vertex polygonal curve  $P$ , approximate it by another polygonal curve  $S$  with the *minimum number* of linear *segments*  $M$  that the approximation error  $E(P)$  does not exceed a given maximum tolerance  $\epsilon$ .

Many heuristic algorithms have been designed for these problems but they lack optimality. Optimal solution is based on dynamic programming algorithm for shortest path in graph for *min-# problem* [1], [5], [6], and  $k$ -link in weighted graph for *min- $\epsilon$  problem* [15], [10].

In [2], [7], [8], [9], [11], [13], [16], [17], [18], [20] polygonal approximation was applied to lossy compression of *digitized contours*. The algorithms were designed for the encoding of object contours for MPEG-4 standard where the vertices of object boundaries are defined on *uniform square grid* and can be represented with integer

undaries are defined on *uniform square grid* and can be represented with integer coordinates or by chain code. Relative coordinates of approximation nodes are encoded with variable length codes.

In the general case, vertex coordinates are given as *real numbers*. Lossless compression of such data type is not very efficient: some vertices of input polygonal curve are discarded, and the coordinates of the approximating nodes are stored with unnecessarily high accuracy. With vector quantization of the approximation nodes we achieve better compression in comparison to *lossless* encoding of the approximating data.

Straightforward quantization of the approximating nodes can destroy optimality of the polygonal approximation solution. For instance, in the case of *min-#* approximation the maximum deviation in a segment after quantization of approximating vertices may exceed the given threshold. On the other hand, applying only quantization of the vertices without reduction of the number of vertices we cannot achieve good rate-distortion performance, especially at a low bit-rate.

To get an optimal solution of the problem in question, we offer to take into joint consideration the effects caused by *Polygonal approximation* and *quantization of the approximating nodes*. Given a bit budget, we have to decide whether we should invest more bits for having more approximating nodes, or for better spatial resolution of the nodes.

## 2 Lossy Compression of Digital Curves

An *open N-vertex polygonal curve*  $P$  in 2-dimensional space is the ordered set of vertices  $P = \{p_1, p_2, \dots, p_N\}$ , and  $p \in \Re^2$ . The polygonal curve  $P$  is approximated by another polygonal curve  $S = \{s_1, \dots, s_{M+1}\}$ , where  $s \in \Re^2$ , and  $M < N$ . End points of  $S$  are end points of  $P$ :  $s_1 = p_1$ ,  $s_{M+1} = p_N$ ; other nodes of  $S$  are defined from vertices of  $P$  by polygonal approximation and quantization procedures.

### 2.1 Vertex Prediction

To use spatial correlation of successive approximating nodes for the data compression, at first a predictor  $p^*(j)$  for vertex  $p(j)$  is calculated, then corresponding *prediction error* (or *residual*) is encoded. Some of preceding vertices can be used as predictor for vertex  $p(j)$ :  $p^*(j) = p(i)$ ,  $i < j$ . Prediction error  $\Delta p(j)$  is defined as difference between the current vertex  $p(j)$  and its predictor  $p^*(j)$ :

$$\Delta p(j) = p(j) - p^*(j). \quad (1)$$

In case of vertices given on uniform square grid, the prediction errors are integer numbers that can be encoded with lossless encoder. In our case, coordinates of vector map vertices are real numbers,  $\Delta p(j) \in \Re^2$ , and quantization of the prediction error induces quantization error. According to the scheme, prediction error  $\Delta p(j)$  for vertex

$p(j)$  is defined as difference between the node  $p(j)$  and *decoded* value  $p_q(i)$  of the node  $p(i)$ :

$$\Delta p(j) = p(j) - p_q(i). \quad (2)$$

In its turn, the decoded value  $p_q(j)$  of the point  $p(j)$  is defined by quantized value  $Z[\Delta p(j)]$  of the prediction error  $\Delta p(j)$  and by the decoded value  $p_q(i)$  of the vertex  $p_q(i)$  as follows:

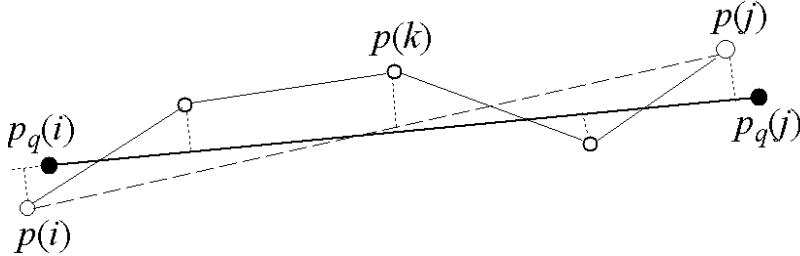
$$p_q(j) = p_q(i) + Z[\Delta p(j) - p_q(i)] \quad (3)$$

With the scheme we avoid accumulation of quantization errors, because every time we quantize difference between target vertex  $p(j)$  and *decoded* point  $p_q(i)$ .

## 2.2 Polygonal Approximation Error

Quantization of prediction errors induces displacement of decoded approximating nodes  $s(m) = p_q(i_m)$  relatively its original location at point  $p(i_m)$ . This distortion affects on error of polygonal approximation with the nodes and should be taken into account. Approximation error for a curve segment  $(p(i), p(i+1), \dots, p(j))$  with measure  $L_2$  is defined as sum of squared distances from vertices  $p(k)$  of the curve segment to the approximating line defined by end points  $p_q(i)$  and  $p_q(j)$  of the segment (see Fig. 1):

$$e_2(p_q(i), p_q(j)) = \sum_{k=i}^j d^2(p(k); (p_q(i), p_q(j))). \quad (4)$$



**Fig. 1.** Scheme of polygonal approximation of curve segment  $(p(i), \dots, p(j))$  (circles) by line segment (solid line) with end points  $p_q(i)$  and  $p_q(j)$  (black dots).

The distance from point  $p(k) = (x_k, y_k)$  to the line  $(p_q(i), p_q(j))$  is given with following expression:

$$d(p(k); (p_q(i), p_q(j))) = \frac{|ax_k + by_k + c|}{a^2 + b^2} \quad (5)$$

where the coefficients  $a$ ,  $b$  and  $c$  of the line can be computed from the values  $p_q(i)$  and  $p_q(j)$  as end points of the line segment. The approximation error  $e_2(p_q(i), p_q(j))$  can be calculated in  $O(1)$  time using pre-computed cumulants for coordinates  $x^3$ ,  $x$ ,

$xy$ ,  $y$ , and  $x^2$  [15]. The total distortion (approximation error)  $E_2(P)$  for the input curve  $P$  is defined as the sum of approximation errors for all approximating segments:

$$E_2(P) = \sum_{m=1}^M e_2(p_q(i_m), p_q(i_{m+1})). \quad (6)$$

The total approximation error with additive error measure  $L_2$  for  $K$  curves  $\{P_1, P_2, \dots, P_K\}$  is defined as sum of the distortions for the curves.

Approximation error for a curve segment  $(p(i), p(i+1), \dots, p(j))$  with measure  $L_\infty$  is defined as maximum distance from vertices  $p(k)$  of the curve segment to the approximating line with end points  $p_q(i)$  and  $p_q(j)$ :

$$e_\infty(p_q(i), p_q(j)) = \max_{i \leq k \leq j} \{d(p(k); (p_q(i), p_q(j)))\} \quad (7)$$

Complexity of algorithm for computation of the maximal deviation for one segment is  $O(N)$ . The total distortion  $E_\infty(P)$  for the input curve  $P$  is defined as maximum of the deviations for all segments:

$$E_\infty(P) = \max_{\{m\}} \{e_\infty(p_q(i_m), p_q(i_{m+1}))\} \quad (8)$$

Distortion with error measure  $L_\infty$  for  $K$  curves  $\{P_1, P_2, \dots, P_K\}$  is defined as maximal distortions for the curves.

### 2.3 Encoding of Prediction Errors

To compress the data, prediction errors have to be quantized with given codebook, and indices of the codecells are to be encoded with entropy encoder; the coordinates of the start vertex  $p(1)$  can be stored without compression. Vector quantizer is defined by set of codecells  $\{C_1, \dots, C_L\}$  and corresponding codevectors  $\{c_1, \dots, c_L\}$  [3], [4]. If prediction error  $\Delta p(j)$  falls into a codecell  $C_l$ , decoded value  $Z(\Delta p(j))$  of the prediction error  $\Delta p(j)$  is defined by codevector of the codecell:

$$\Delta p(j) \in C_l \Rightarrow Z(\Delta p(j)) = c_l. \quad (9)$$

The integer index  $l$  of the codecell can be compressed with variable length codes. To encode a prediction error  $\Delta p(j)$  with given vector quantizer, we have to test all code vectors from codebook to find the nearest one.

*Uniform product quantizer* is defined by quantization step  $q$ . For the quantizer the pair of indices  $(l_x, l_y)$  and quantized values  $Z(\Delta x), Z(\Delta y)$ , where  $\Delta x$  and  $\Delta y$  are Cartesian coordinates of  $\Delta p(j)$ , are defined as follows:

$$\{l_u = [\Delta u / q]\}, \quad Z(\Delta u) = q \cdot l_u, \quad (10)$$

Bit-rate for the encoding the current approximating node  $p_q(i_{m+1})$  is defined by number bits  $r(p_q(i_m), p_q(i_{m+1}))$  to encode the quantized prediction error  $Z(\Delta p(i_{m+1}))$ . The total bit-rate  $R(P)$  for the approximating polygon  $P$  is sum of bit-rates for all the approximating nodes of  $S$ :

$$R(P) = \sum_{m=1}^M r(p_q(i_m), p_q(i_{m+1})), \quad (11)$$

here bit-rate for encoding the start vertex is omitted for sake of simplicity, because it does not affect on optimization procedure.

For uniform product quantizer with step  $q$  the bit-rate  $r(p_q(i_m), p_q(i_{m+1}))$  can be estimated by entropy  $H$  of the prediction error:

$$H = \lceil \log_2(2|l_x|) \rceil + \lceil \log_2(2|l_y|) \rceil, \quad (12)$$

where  $\Delta x$  and  $\Delta y$  are Cartesian coordinates of prediction error  $\Delta p(i_{m+1})$ .

## 2.4 Optimal Encoding of Approximating Nodes with $L_2$ Error Measure

The problem of optimal polygonal approximation and quantization of prediction errors for approximating nodes can be stated as follows: approximate the polygonal curve  $P$  by the polygonal curve  $S$  so that total approximation error  $E_2(P)$  is minimized, and bit-rate  $R(P)$  does not exceed the given threshold  $R_{\max}$ :

$$\begin{aligned} E_2(P) &= \min_{\{i_m\}} \left\{ \sum_{m=1}^M e_2(p_q(i_m), p_q(i_{m+1})) \right\} \\ \text{subject to } &\sum_{m=1}^M r(p_q(i_m), p_q(i_{m+1})) < R_{\max}. \end{aligned} \quad (13)$$

For sake of simplicity we omit term connected with start point encoding. The found set of vertices  $\{p_q(i_m)\}$  give us vertices of approximating curve  $S$ :  $s(m)=p_q(i_m)$ .

The *constrained optimization problem* can be converted into *unconstrained problem* with Lagrange multiplier  $\lambda$  introducing modified cost function  $D$ :

$$D = E_2 + \lambda \cdot R = \min_{\{i_m\}} \left\{ \sum_{m=1}^M (e_2(p_q(i_m), p_q(i_{m+1})) + \lambda \cdot r(p_q(i_m), p_q(i_{m+1}))) \right\}. \quad (14)$$

To find solution of the *unconstrained* problem, a directed acyclic graph  $G$  is constructed. Nodes of the graph correspond to vertices  $(p(1), p(2), \dots, p(N))$  of  $P$ . Edge  $v(i, j)$  of  $G$  corresponds to approximation of curve segment  $(p(i), p(i+1), \dots, p(j))$  by line segment  $(p_q(i), p_q(j))$ . Weight  $W(i, j)$  of the edge  $v(i, j)$  is defined as value of modified cost function  $D(i, j) = e(i, j) + \lambda r(i, j)$ . Solution of the problem under is given by shortest path on the weighted graph  $G$  with given cost function. To find solution which satisfies the constraint on bit-rate  $R < R_{\max}$  we have to repeat construction of the shortest path in  $G$  for different values of  $\lambda$ . The optimal value of  $\lambda$  can be found by bisection, for example.

To calculate the shortest path in directed acyclic graph  $G$  we need two 1-dimensional arrays: one array  $D[j]$  for cost function  $D$ , another one is array  $A[j]$  of pointers to parent nodes which provides optimal solution for the current node. To find the best parent node  $n(i_{\text{opt}})$  for a node  $n(j)$  we have to calculate cost functions for all

parent nodes  $n(i)$ , where  $i < j$ , and select that one which provides minimum of the cost function  $D$  for the node  $n(j)$ . In the case under consideration, weight  $W(i, j)$  of edge  $v(i, j)$  is defined by *decoded* values of vertex  $p_q(j)$  and parent vertex  $p_q(i)$ . The decoded value  $p_q(j)$  is defined by  $p_q(i)$ , see (3), and this value is different for different parent vertices considered as candidates to approximating node. To reduce processing time, for every node  $n(j)$  we keep decoded value  $p_q(i)$  of approximating node along with pointer to the best parent node  $A[j]=n(i_{\text{opt}})$ .

Weights of edges cannot be calculated in advance for the whole graph as in the case without quantization, but the weights can be calculated sequentially during the vertices processing. current state. However, weight of graph is completely defined by (decoded) previous approximation vertex, so principle of optimality is not violated, and dynamic programming algorithm can be applied for the shortest path construction in the case under consideration.

Complexity of the algorithm is defined by complexity of algorithm for the shortest path in graph, which is  $O(N^2)$  and the number of iterations to calculate  $\lambda$ . The time complexity of approximation error calculation is  $O(1)$ ; complexity of vector quantization is  $O(L)$ , where  $L$  is size of codebook (the number of codewords).

The algorithm for compression of one curve can be extended on the case of  $K$  curves. All the curves are processed sequentially, the total distortion is defined as sum of distortions for all the objects, and the total bit-rate is sum of bit-rates for the objects, too. Modified cost function  $D=E_2+R$  for  $K$  objects is sum of cost functions for all the objects, and Lagrange multiplier  $\lambda$  is one for all shapes.

## 2.5 Optimal Encoding of Approximating Nodes with $L_\infty$ Error Measure

Now let us extend the approach to the algorithm for vector data compression with  $L_\infty$  error measure. In such a case, the problem of optimal compression with polygonal approximation and prediction errors encoding can be stated as follows: approximate the polygonal curve  $P$  by the polygonal curve  $S$  so that total bit-rate  $R(P)$  is minimized and approximation error  $E_\infty(P)$  does not exceed the given threshold  $D_{\max}$ :

$$R(P) = \min_{\{i_m\}} \left\{ \sum_{m=1}^M r(p_q(i_m), p_q(i_{m+1})) \right\} \quad \text{subject to: } E_\infty(P) < D_{\max}. \quad (15)$$

Solution of the optimization problem can be found as shortest path in weighted acyclic graph  $G$  taking into consideration quantization of prediction errors for approximating nodes (vertices of  $S$ ). Nodes of the *feasibility* graph  $G$  correspond to vertices of  $P$ . Edge  $v(i, j)$  connecting two nodes  $n(i)$  and  $n(j)$  corresponds to approximation of segment  $(p(i), p(i+1), \dots, p(j))$  by line segment  $(p_q(i), p_q(j))$ , where  $p_q(i)$  and  $p_q(j)$  are *decoded* values of end points  $p(i)$  and  $p(j)$ . Weight  $W(i, j)$  of the edge  $v(i, j)$  is defined as follows:

$$W(i, j) = \begin{cases} r(p_q(i_m), p_q(i_{m+1})), & \text{if } \max_{i \leq k \leq j} \{d(p(k); (p_q(i_m), p_q(i_{m+1})))\} < D_{\max}, \\ \infty, & \text{otherwise.} \end{cases} \quad (16)$$

Again, in addition to two arrays to keep the current cost function (the bit-rate) values and parent nodes, we need one array to store *decoded* values of approximation nodes we need to calculate *real* distortions and bit-rates.

Complexity of the algorithm is defined by complexity of the algorithm for the shortest path construction, which is  $O(N^2)$ , by complexity of algorithm for maximum deviation calculation,  $O(N)$ , and by complexity of vector quantization. That gives us the overall time complexity  $O(LN^3)$ ; the spatial complexity is  $O(N)$ . The algorithm for compression of one curve can be extended on the case of  $K$  curves in natural way: all the curves are processed individually, the total bit-rate is defined as sum of bit-rates obtained for the curves.

## 2.6 Construction of Vector Quantizer

Generally speaking, the distortion  $E_2(P)$  or bit rate  $R$  can be minimized also by proper choice of vector quantizer. To achieve the minimum of the cost function for the given bit-rate, the optimal quantizer can be constructed basing on statistics of prediction error[3], [4], [12]. On the other hand, in the case under consideration, the statistics (probability distribution) of prediction error is defined by 1) polygonal approximation and 2) quantization of the residuals. To construct quantizer, we need statistics, but the statistics depends on the quantizer.

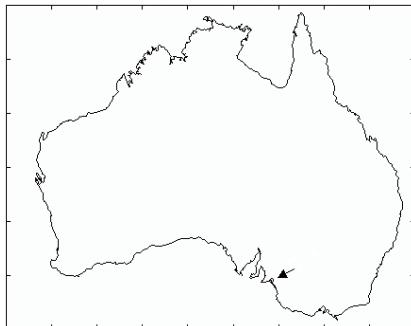
The problem can be solved with iterative algorithm. We can start with some vector quantizer (for instance, with uniform product scalar quantizer) to find optimal polygonal approximation for the quantizer and collect statistics of residuals. Then we design rate-distortion optimal vector quantizer using the collected statistics, and use the codebook to calculate the rate-distortion optimal polygonal approximation. The process is iterated until no improvement or some fixed number of runs.

In the case of uniform scalar product quantizer, the quantization steps  $q_x$  and  $q_y$  has to be found to minimize the cost function.

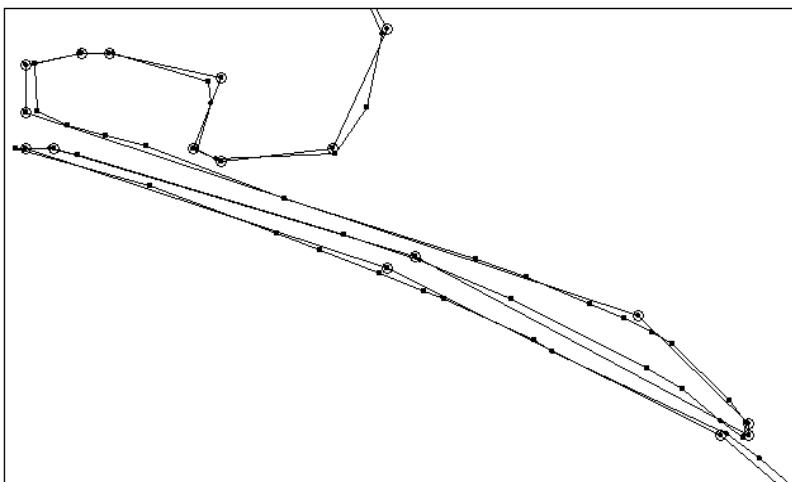
## 3 Results and Discussion

To illustrate proposed algorithm for *min-ε problem* and estimate performance of the algorithm we use 2903-vertex shape "Australia" (see Fig. 2). Prediction errors were quantized with uniform scalar product quantizer. Bit-rate was estimated as entropy of quantized prediction error for one point. For uniform product quantizer for relative coordinates, approximating nodes are located on regular square lattice with step  $q$ .

Result of lossy compression of the test shape is represented on Figs. 3. Original data are stored with 8 bytes per point. For target bit-rate  $R=2$  bit/point, the optimal quantization step  $q_{\text{opt}}=1/2^{4/2}$ , the number of segments  $M=1079$ , and distortion  $E=0.281$ . Reduction of the vertex number provides 2.7:1 compression ratio, the further improvement of the ratio is result of quantization of relative coordinates.



**Fig. 2.** Test shape "Australia",  $N=2903$ . The labeled by arrow fragment depicted on Fig. 3.



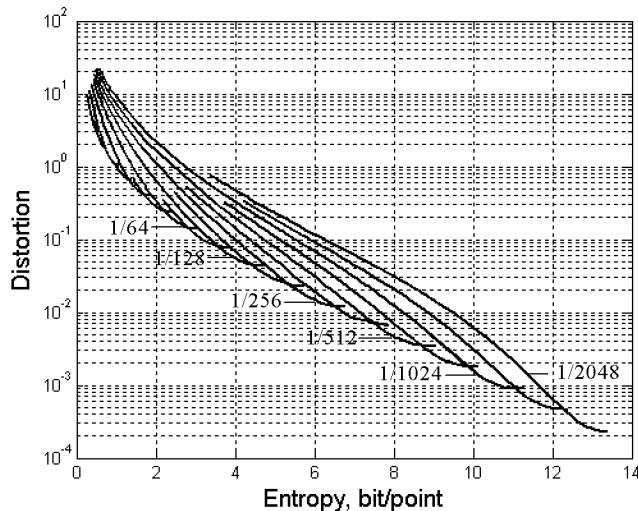
**Fig. 3.** Result of decoding for the test shape (fragment); bit-rate  $R=2$  bit/point. Vertices of original curve  $P$  are labeled with dots, approximation nodes of  $S$  are labeled with circles. The fragment is labeled by arrow on Fig. 2.

The rate-distortion functions were calculated for quantization steps  $q=1/2^k$ , where  $k=3, 3\frac{1}{2}, 4, \dots, 11$  with step  $\frac{1}{2}$  (see Fig. 4). As it follows from the Fig. 4, for the given bit-rate, bigger quantization step provides smaller distortion, because with bigger quantization step  $q$  we can afford the bigger number of approximating nodes; the bigger number of approximating nodes means smaller polygonal approximation error. On the other hand, the bigger quantization step means bigger quantization error, with big quantization error we cannot reduce the total distortion below some limit, even we still have got some reserve of bits.

With smaller quantization step we can further reduce quantization error by cost of bigger polygonal approximation component of the distortion, caused by smaller number of approximating nodes. With bigger target bit-rate we can afford smaller quantization step to invest more bits into spatial resolution of approximating nodes. So, for

given target bit-rate we have to find the optimal quantization step which provides minimum of the total distortion.

The optimal solutions of *min-ε problem* for different bit-rates are given by points on *lower envelope* of the curves. In the case under consideration, the dependence of logarithm of distortion  $\log(D)$  on bit-rate  $R$  is almost linear in the interval 1-14 bit/point.



**Fig. 4.** Rate-distortion functions for quantization steps  $q_k=1/2^k$ , where  $k=3, 3\frac{1}{2}, \dots, 11$ . The corresponding rate-distortion curves are followed from left to right.

The entropy of relative coordinates that has been used for bit-rate estimation gives the theoretical low limit for the bit-rate. For real encoder the distortion should be bigger to satisfy the given constraint on bit-rate. On the other hand, with better vector quantizer we can reduce distortion and improve rate-distortion performance in comparison with the uniform product scalar quantizer in use.

## 4 Conclusions

The problem of optimal of vector data (vector maps) is considered. We formulate the problem by taking into joint consideration data reduction by polygonal approximation, and quantization of the prediction errors for approximation nodes. Optimal algorithms for vector data compression with minimal distortion for given target bit-rate, and with minimal bit-rate for given maximum deviation are suggested. The proposed approach can be generalized on the case of vector data approximation with non-linear functions (polynomials, splines, and wavelets).

## References

1. Dunham, J.: Optimum uniform piecewise linear approximation of planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8 (1986) 67-75
2. Chung, J.-W., Lee, J.-H., Moon, J.-H., Kim, J.-K.: A new vertex-based binary shape coder for high coding efficiency. *Signal Processing: Image Communication*, 15 (2000) 665-684
3. Gersho, A., Gray, M.: *Vector Quantization and Signal Compression*. Kluwer Int. Series in Engineering and Computer Science, Vol. 152 (1992)
4. Gray, R.M., Neuhoff, D. L.: Quantization, *IEEE Trans. Information Theory*, 44 (1998) 2325-2383
5. Imai, H., Iri, M.: Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics and Image Process.* 36 (1986) 31-41
6. Imai, H., Iri, M.,: Polygonal approximations of a curve (formulations and algorithms). In: G.T.Toussaint, (Ed), *Computational Morphology*, North-Holland, Amsterdam, (1988) 71-86
7. Hu, M., Worrall, S., Sadka, A., Kondoz, A.M.: A scalable vertex-based shape intra-coding scheme for video objects. In: Proc. Int. Conf. Acoustics, Speech, and Signal Process.- ICASSP'04. Vol. 3 (2004) 273-276
8. Katsaggelos, A.K., Kondi, L.P., Meier, F.W., Ostermann, J., Schuster, G.M.: MPEG-4 and rate-distortion-based shape-coding techniques. *Proc. IEEE*, 86 (1998) 1126-1154
9. Kim, J.I., Bovik, A.C., Evans, B.L.: Generalized predictive binary shape coding using polygon approximations. *Signal Processing: Image Communication*, 15 (2000) 643-663
10. Kolesnikov, A., Fränti, P.: Reduced-search dynamic programming for approximation of polygonal curves. *Pattern Recognition Letters*, 24 (2003) 2243-2254
11. Kolesnikov, A., Fränti, P.: Data reduction of large vector graphics. *Pattern Recognition*, vol. 38 (2005) 381-394
12. Akimov, A., Kolesnikov, A., Fränti, P.: Coordinate quantization in vector map compression. In: Proc. IASTED Int. Conf. Visualization, Imaging and Image Process.-VIIP'04, (2004) 748-753
13. Le Buhan, C., Ebrahimi, T.: Progressive polygon. encoding of shape contours. In: Proc. Int. Conf. Image Processing and its Applications, (1997) 1721
14. Li, Z., Openshaw, S.: Algorithms for objective generalization of line features based on the natural principle. *Int. J. Geographical Information Systems*, 6 (1992) 373-389
15. Perez, J.C., Vidal, E.: Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15 (1994) 743-750
16. Servais, M., Vlachos, T.: Progressive polygon encoding of segmentation maps. In: Proc. Int. Conf. Image Process.-ICIP'04, Singapore (2004) 1121-1124
17. Schuster, G.M., Melnikov, G., Katsaggelos, A.K.: Operationally optimal vertex-based shape coding. *IEEE Signal Processing Magazine*, 15 (1998) 91-108
18. Schuster, G.M., Katsaggelos, A.K.: An optimal polygonal boundary encoding scheme in the rate-distortion sense. *IEEE Trans. Image Proc.* 7 (1998) 13-26
19. Shekhar, S., Huang, Y., Djugash, J., Zhou, C.: Vector map compression: a clustering approach. In: Proc. 10th ACM Int. Symp. Advances in Geographic Inform. Syst.-GIS'02 (2002) 74-80
20. Zaletelj, J., Tasic, J.: Optimization and tracking of polygon vertices for shape coding. In: *Lecture Notes in Computer Science*, Vol. 2756. Springer-Verlag, Berlin Heidelberg New York (2003) 418-425

# Image Compression Using Adaptive Variable Degree Variable Segment Length Chebyshev Polynomials

I.A. Al-Jarwan and M.J. Zemerly

Department of Computer Engineering  
Etisalat College of Engineering  
PO Box: 980, Sharjah, United Arab Emirates  
[jamal@ece.ac.ae](mailto:jamal@ece.ac.ae)  
<http://www.ece.ac.ae>

**Abstract.** In this paper, a new lossy image compression technique based on adaptive variable degree variable segment length Chebyshev polynomials is proposed. The main advantage of this method over JPEG is that it has a direct individual error control where the maximum error in gray level difference between the original and the reconstructed images can be specified by the user. This is a requirement for medical applications where near lossless quality is needed. The compression is achieved by representing the gray level variations across any determined section of a row or column of an image by the coefficients of a Chebyshev polynomial. The performance of the method was evaluated on a number of test images and using some quantitative measures compared to the well known JPEG compression techniques.

## 1 Introduction

Medical applications cannot tolerate visual distortion produced by compression techniques. There is a requirement to obtain near-lossless quality from decompressed images in order to preserve the information required to make a correct diagnosis. Usually, there is a compromise between the distortion that can be tolerated and the amount of compression obtained. Sometimes also like in JPEG [1] there is no direct control over the individual error obtained. Rather, a percentage quality measure is given that may not be sufficient to medical applications as it may affect the validity of a diagnosis, for example. Increasing the percentage quality reduces the individual errors but it may require a number of attempts to control the individual pixel error to an acceptable level. In this paper, it was found that individual errors of 10 may be acceptable in most applications and to a certain extent visual distortion starts to appear if the error exceeds this level.

A new lossy image compression technique based on representing the gray level variations of an image by a series of variable degree Chebyshev polynomials is presented. Image compression is achieved by storing the polynomial coefficients obtained from fitting a variable size segment (a number of pixels) size, which are generally fewer than the original number of pixels. The coefficients are rounded-off to

be stored in a byte each to reduce their size. The decompression is obtained using the stored coefficients for each segment using a fast and efficient technique.

Previous work using polynomials concentrated on applying surface or region based fitting using implicit polynomials, which suffer from being ill formed and cannot reach high degrees [2-6]. Results of these techniques showed visual distortion that cannot be tolerated in many applications, let alone medical ones. Previous work done by the authors using Chebyshev polynomials produced 2 new lossy methods termed FDVSL and VDFSL [7, 8]. These 2 methods are also described later in this paper and will be compared with the new proposed method. The method discussed here is concerned with compressing images with no or very little visible distortion (near-lossless quality). The amount of distortion is measured also quantitatively by a number of performance measures commonly used in assessing image compression techniques such as Peak Signal-to-Noise ratio (*PSNR*) and Normalized Cross Correlation (*NCC*).

## 2 Problem Formulation

Chebyshev polynomials are among the most popular orthogonal polynomials that are used to approximate a set of data and are useful in such contexts as numerical analysis and circuit design [9-10]. They form an orthogonal set. A (type I) Chebyshev polynomials  $T_n(x)$  are generated via the equation:

$$T_n(x) = \cos(n \arccos x) \quad (1)$$

Equation 1 can be combined with trigonometric identities to produce explicit expressions for  $T_n(x)$ .

So, for Chebyshev polynomials of the first order:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1 \\ T_{n+1}(x) &= 2x T_n(x) - T_{n-1}(x) \quad \text{for } n > 1 \end{aligned} \quad (2)$$

To calculate the coefficients of the Chebyshev polynomial, the following method is used:

The residual sum of squares is given by:

$$\partial_i^2 = \sum_{r=0}^m \{Y_i(x_r) - y_r\}^2 \quad (3)$$

where  $y_r$  ( $r = 0, 1, 2 \dots m$ ) is the observed or computed values of dependent variable  $y$  at given values  $x_r$  of the independent variable  $x$ .  $Y_i(x)$  is the polynomial of degree  $i$  which minimises the residual sum of squares.

The polynomial  $Y_i(x)$  may be obtained by truncating the series after  $(i+1)$ :

$$c_0 p_0(x) + c_1 p_1(x) + c_2 p_2(x) + \dots \quad (4)$$

where  $p_i(x)$  is a polynomial of degree  $i$  satisfying the orthogonality condition:

$$\sum_r p_i(x_r) p_j(x_r) = 0 \quad (i \neq j) \quad (5)$$

The coefficients  $c_j$  in (4) are therefore given by [10]:

$$c_j = \frac{\sum_r y_r p_j(x_r)}{\sum_r \{p_j(x_r)\}^2} \quad (6)$$

Approximating a curve with Chebyshev polynomials has some important advantages. For one, if the curve is well approximated by a converging power series, one can obtain an equally accurate estimate using fewer terms of the corresponding Chebyshev series. More importantly is the property over the interval [-1, 1], each polynomial has a domain of [-1, 1]; thus, the series is nicely bounded. And because of this bounded property, approximations calculated from a Chebyshev series are less susceptible to machine rounding errors than the equivalent power series.

Variable  $x$  is obtained from the following equation where  $[a]$  and  $[b]$  are the minimum and maximum coordinates of  $X$  respectively. The usage of this variable will affect the range of approximation to be between -1 to 1 instead of two arbitrary limits a and b. [9-11]

$$x = \frac{2X - (b + a)}{(b - a)} \quad (7)$$

Choosing Chebyshev polynomials as a compression technique is due to a number of factors: a) they are orthogonal and numerically stable. b) they offer rapid convergence for possible truncation. c) they are normalised polynomials. d) they have the smallest maximal deviation from zero. Under this aspect they are unique:

$$\max_{0 \leq x \leq 1} \left| \frac{T_n(x)}{2^n} \right| = \frac{1}{2^n} \quad [10].$$

Clenshaw recurrence formula is used to evaluate (reconstruct) Chebyshev polynomials efficiently using the following equations rather than the normal CPU hungry power series used in each Chebyshev polynomial [10]:

$$\begin{aligned} d_{m+1} &\equiv d_m \equiv 0 \\ dj &= 2x d_{j+1} - d_{j+2} + c_j \quad \text{for } j=m-1, m-2, \dots, 1 \\ f(x) &\equiv d_0 = x d_1 - d_2 + c_0 \end{aligned} \quad (8)$$

There are five compression performance evaluation measures (termed statistics) that are calculated to evaluate the quality of the reconstructed compressed image. These are the Deviation per Pixel ( $DPP$ ), the Normalised Cross Correlation ( $NCC$ ), the Mean Square Error ( $E_{ms}^2$ ), the ( $PSNR$ ) and the Compression Ratio ( $CR$ ).

The (*DPP*) is given by:

$$DPP = \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} |f(x, y) - f_r(x, y)|}{X \times Y} \quad (9)$$

$f(x, y)$  and  $f_r(x, y)$  are the gray levels of the corresponding  $x$  (column) and  $y$  (row) in the original and reconstructed images respectively. In the above equation,  $X$  is the number of columns (width of image) and  $Y$  is the number of rows (height of image).

The (*NCC*) is given by:

$$NCC = \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} [f(x, y) \times f_r(x, y)]}{\left[ \left( \sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} f^2(x, y) \right) \times \left( \sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} f_r^2(x, y) \right) \right]^{\frac{1}{2}}} \quad (10)$$

A value of *NCC* close to 1 means that the reconstructed image is close to the original, thus a value of 1 indicates that the image is compared to exact copy of itself [12].

The Mean Square Error  $E_{ms}^2$  is given by:

$$E_{ms}^2 = \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} [f(x, y) - f_r(x, y)]^2}{X \times Y} \quad (11)$$

The (*PSNR*) is given by:

$$PSNR = 10 \times \log \left( \frac{(255)^2}{E_{ms}^2} \right) \text{ [dB]} \quad (12)$$

The higher the *PSNR*, the lower the noise is in the reconstructed image [12].

The *CR* is given by:

$$CR = \frac{B}{C} \quad (13)$$

B is the number of bytes in the original image and C is the number of bytes needed to store the compressed image [12].

### 3 Adaptive Chebyshev Fitting

Three methods of compression that utilize Chebyshev polynomials are described in this paper. Here, the VDVSL method (see later) was implemented and tested after implementing and testing the first two methods (VDFSL & FDVSL) in previous

publications [7,8]. These methods are called: *Variable Degree Fixed Segment Length* (VDFSL), *Fixed Degree Variable Segment Length* (FDVSL) and *Variable Degree Variable Segment Length* (VDVSL). All the three methods share the same concept of representing a group of pixels (segment) of the image as a Chebyshev polynomial; however, they are different in the fashion in which the compression of the image is achieved. First, the implemented method, VDVSL, will be described and some of the testing results obtained will be highlighted in the next section. The way in which VDFSL and FDVSL operate will be outlined later.

### **3.1 Variable Degree Variable Segment Length (VDVSL)**

This method starts by fitting the whole row/column (depending on option) as a segment starting from zero degree and increases the degree if error is exceeded. If the segment is not compressed (reached the maximum degree allowed i.e. 7), then two new segments will be taken instead. Each new segment has half the number of pixels in the previous segment. The method will then try to compress each segment individually. If any segment is not compressed, then the process is repeated for any non-compressed segment in which the segment will be divided into two segments that have the same number of pixels. The process is repeated until reaching the minimum allowed number of pixels in a segment. The minimum allowed number of pixels in a segment could be fixed to 4 or could be made an input value entered by the user along with the maximum individual error. The compression is finished when all the rows/columns in the image are fitted. Each segment is represented by a byte followed by the coefficients. The first byte is used for the degree (in 4 bits; maximum degree i.e. 7 needs only 3 bits but negative degrees used for special cases) and the number of pixels as power of 2 (also 4 bits, i.e. if the number of pixels is 128 then 7 is stored).

As an optional input that the user can specify, there is the maximum allowed individual error between the original and the reconstructed pixel values. The default value is taken to be 10, as this amount of difference in gray levels can hardly be noticed by the human eye [12].

#### **3.1.1 Option 1: Row-Segment**

Each row is compressed separately into a number of polynomials that represent segments in the same way that is described previously.

A fitted segment is represented by storing the degree of the fitting polynomial with the number of pixels in one byte followed by the coefficients of the polynomial rounded to bytes. Noted that the number of coefficients = degree +1.

#### **3.1.2 Option 2: Column-Segment**

This option is identical to option 1. The only difference is that a column in an image is compressed instead of a row. Otherwise, the processing steps of the two options are identical.

### **3.2 Variable Degree Fixed Segment Length (VDFSL)**

In this method a row/column is divided equally into a number of segments each with an equal number of pixels (or the image is divide into  $4 \times 4$  portions). Starting with a degree of 0, the pixels in the segment are fitted and all reconstructed pixels are

checked against the maximum error allowed. If the maximum error is exceeded, then the degree is incremented and the fitting is repeated. The same fitting procedure is repeated until either the errors are acceptable or the degree exceeds the maximum degree allowed, which is the maximum of number of pixels in a segment divided by two or 8. In the latter case, a special value of -1 is stored (as the degree of the polynomial) and then each pixel will be represented by a value equals to the original data of the pixels minus 128 (in order to fit within a byte).

A fitted segment is represented by storing the degree of the fitting polynomial followed by the coefficients of the polynomial rounded to bytes. Note that the number of coefficients = degree +1.

### 3.3 Fixed Degree Variable Segment Length (FDVSL)

In this method the number of pixels in each segment is specified dynamically. That is the algorithm starts from a fixed number of pixels (in this case it is 4 for degree 2) and the pixels are fitted using the Chebyshev polynomials and if the error is acceptable the number of points is increased by one (the number of increased pixels is an input by the user to achieve better execution time). The fitting is repeated every time a new point is added and the error is checked if it is greater than the maximum individual accepted error. If no it continues adding new points until it reaches 127 pixels (because the byte can only handle numbers from -128 to 127) and start again or the error exceeds the acceptable range.

Another critical case is that when we left with less than 4 points at the end of each row we can not handle it so to solve it a special value of -1 is stored and then each pixel will be represented by a value = original gray levels -128.

A fitted segment is represented by storing the number pixels in the segment of the fitting polynomial followed by the coefficients of the polynomial. It is noted that the number of coefficients = degree +1. The special degree of -1 indicates that individual values for each pixel in the segment are stored instead of the coefficients. When the compressed file is read, this means that either the segment of 4 points was not fitted with the degree specified or the remaining points left at the end of a row is less than 4.

There are three options for FDVSL. All share the idea of representing a group of pixels (termed segment) by the coefficients of a Chebyshev polynomial. These options are: *row-segment*, *column-segment* and *all-rows-together* [8].

As an optional input that the user can specify, there is the maximum allowed individual error between the original and the reconstructed pixel values. The default value is taken to be 5, as this amount of difference in gray levels can hardly be noticed by the human eye [12]. The second optional input is the degree of the fitting polynomial. The default value is set to 2. A third parameter is also introduced to speed up processing which is the number of pixels to add to the segment in case the error is acceptable. The default is 1 pixel.

### 3.4 Compressed Image Format

The compressed image is represented and stored in a file having the extension “.ch”. As in the case of PGM, characters that locate from "#" to the next end-of-line are ignored (comments). The compressed image is represented as the following:

- The width and height of the image (2 bytes for each)
- The segments are represented sequentially starting from the first row or first column in two ways:
  1. Compressed segment: consists of one byte for the degree (4 bits) and the number of pixels as power of 2 (4 bits) and (Degree + 1) of coefficients.
  2. Non-compressed segment (in the case of 4 points that are not fitted at the end of segmenting): A special value of -1 and a value, of the actual greyscale value – 128, is stored for each pixel in the segment to convert grey levels (0-255) into a byte in the compressed format.

## 4 Results and Discussion

Many images were tested using the 3 methods but only results for 3 representative images are shown here as similar results were obtained for most of the tested images.

A test Image of “Lena” of size 256×256 (not shown here) was used to compare FDVSL, VDFSL, VDVSL and the JPEG [Quality factor 80%] compression technique (see results in Table 1). The second option of VDVSL was selected for comparison with input of 10 maximum allowed individual errors. The second option of FDVSL was selected for comparison with inputs of degree 2 and 10 maximum allowed individual errors. The second option of VDFSL was selected for comparison with inputs of 16 pixels in a segment and 10 maximum allowed individual errors.

**Table 1.** Shows the statistics previously described obtained from the comparison for Lena

Statistics	VDFSL	FDVSL	VDVSL	JPEG
DPP	2.1166	3.15964	2.82904	2.65884
NCC	0.9996	0.99934	0.999453	0.999425
$E_{ms}^2$	10.128	16.5071	13.6509	14.327423
PSNR	38.07	35.9541	36.7792	36.56912
CR	2.2531	3.62930	3.13488	4.939775
Max Error	10	10	10	28

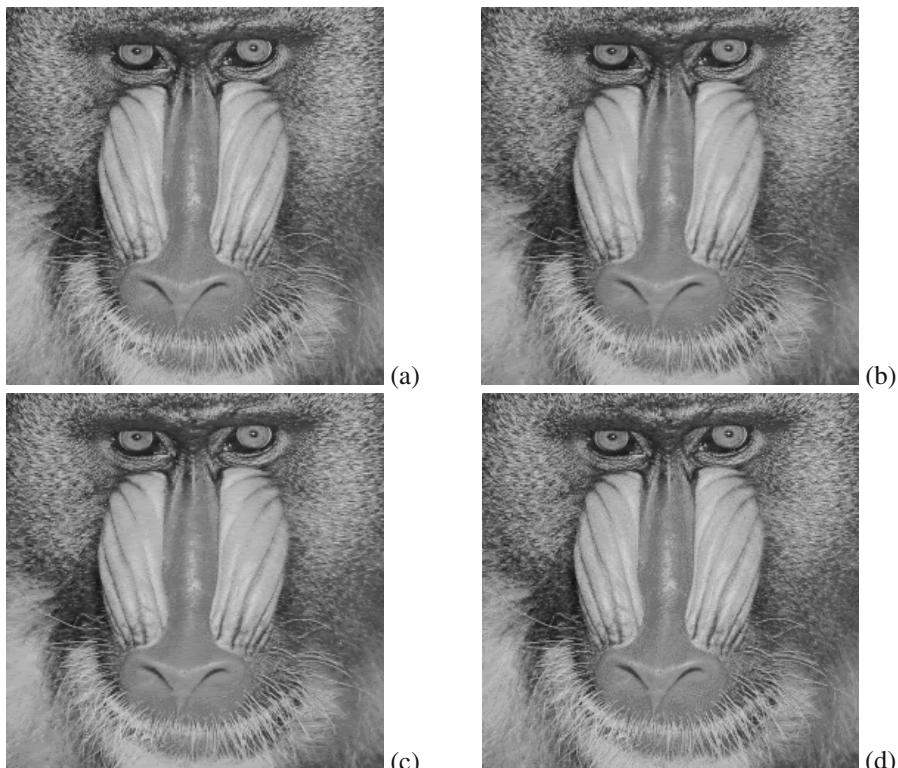
A texture image (baboon) of size 256×256 was used to compare FDVSL, VDFSL, VDVSL and the JPEG [Quality factor 89%] compression technique. The first option of VDVSL, VDFSL and FDVSL was selected for comparison with input of 10 maximum allowed individual errors, degree 2 [For FDVSL] and 16 pixels in a segment [For VDFSL]. Table 2 summarizes the results obtained for the three methods.

For this image the CR of the JPEG compression technique is better than the other two with comparable compression ratios. Increasing Maximum Error to 14 for VDFSL will result in PSNR of 35.9015 (i.e. approximately equal to the others) and CR of 1.26591. Figure 1 shows the original image of the baboon image and the

reconstructed images obtained from FDVSL, VDFSL and VDVSL as well as that of JPEG. In these two cases no visible distortion could be seen in any of the 4 methods.

**Table 2.** Shows the statistics previously described obtained from the comparison

Statistics	VDFSL	FDVSL	VDVSL	JPEG
DPP	0.755325	3.41394	2.99699	3.2038
NCC	0.999871	0.999438	0.99953	0.99953
$E_{ms}^2$	4.62044	20.0552	16.6964	16.65769
PSNR	41.484	35.1085	35.9046	35.91465
CR	1.05903	1.61523	1.53978	2.316414
Max Error	10	10	10	20



**Fig. 1.** An example of the results: (a) Original PGM image, (b) Reconstructed image using VDVSL, (c) Reconstructed image using FDVSL, and (d) Reconstructed image using JPEG

The proposed image compression technique was tested on a number of test images. The two options of VDVSL generally produced acceptable compression ratios when the default optional input value is used.

On the other hand, increasing the individual error led to an increase in the compression ratio, but the quality of the image deteriorated. The quality of the reconstructed compressed image became worse when reaching high individual error and differences were clearly seen in the case of maximum individual error of 15.

The first option of VDVSL was compared with the JPEG compression technique. The optional input is selected to have a maximum individual error of 10. The choice was done based on acquiring an acceptable compression ratio (1.53978), while keeping the quality of the reconstructed compressed image acceptable for use. VDVSL gave better results in the terms of calculated statistics; better *NCC*, *DPP* and maximum error (see tables 1, 2).

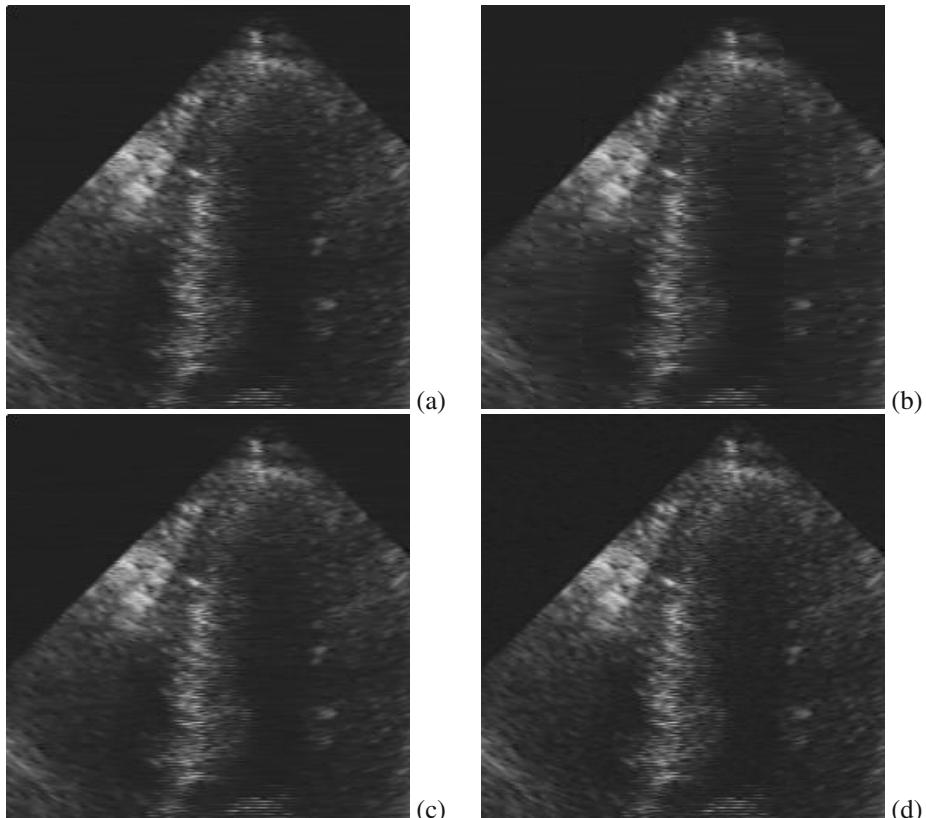
One major advantage of VDVSL (and other methods that depend on Chebyshev Polynomials) over JPEG is the ability to control the maximum individual error. This is clearly evident from Table 2 as the maximum individual error of VDVSL was 10 (specified maximum individual error), while that of JPEG was 20 (using quality of 89). To reduce the error in JPEG to 14 quality 93 was used and then compression ratio of only 1.051 was obtained [with *DPP* = 2.099, *NCC* = 0.999796, *PSNR* = 39.5216].

This system can be applied in fields where quality of the reconstructed image is the main issue with a reasonable compression ratio such as medical images. An ultrasound scan image of size 256×256 was used to compare FDVSL, VDFSL, VDVSL and the JPEG [Quality factor 75%] technique with the same specifications of the previous test. Table 3 summarizes the results obtained for the three methods.

**Table 3.** shows the statistics previously described obtained from the comparison

Statistics	VDFSL	FDVSL	VDVSL	JPEG
DPP	2.77173	3.11668	2.91797	2.1523
NCC	0.99748	0.99692	0.99729	0.99836
$E_{ms}^2$	13.2404	16.2639	14.2847	5.96339
PSNR	36.9118	36.0186	36.5821	38.7588
CR	4.70365	6.211543	6.341977	7.287445
Max Error	10	10	10	21

For this image the quality of the JPEG compression technique is better (in PSNR) than the other three with comparable compression ratios. Also, VDVSL performs better than FDVSL and VDFSL in texture images but slower than the two other methods. Figure 2 shows the original image of the ultrasound scan image and the reconstructed images obtained from FDVSL, VDFSL and VDVSL as well as that of JPEG. Visual distortion in the first three reconstructed images can hardly be seen to the naked eye as the maximum individual error for any pixel is 10. No visible distortion could be seen in any of the 4 methods despite the fact that the individual error in the JPEG image is 21.



**Fig. 2.** An example of the results: (a) Original PGM image, (b) Reconstructed image using VDVSL, (c) Reconstructed image using FDVSL, and (d) Reconstructed image using JPEG

The Chebyshev methods described in this paper can be used in applications where the errors of the reconstructed compressed image needed to be directly controlled and the quality of textured areas preserved. In medical imaging, for example, an error reaching a value of 28 (as JPEG for Lena) may result in wrong diagnosis obtained from the image.

Another advantage is that VDFSL, FDVSL and VDVSL preserve the quality of the most difficult parts of the image that cannot be compressed, as they are stored without any change (only some pre-processing and post-processing operations to store the original values in a byte format).

It was observed that compression ratios obtained from the two options (row/column) on the same input image were not the same (for fixed optional inputs); one of the two compression ratios is better than the other one with little difference in the other statistics. As a different image was tested, it was found that the best compression was not generated by the same option that generated the previous best compression. The possibility of one method producing a better compression than the other one is related directly to the structure or shape of the input image. For example,

for the Lena image the column option gave slightly better results than the others because of vertical uniform areas. This can be considered as an advantage for these 3 methods that depend on Chebyshev Polynomials.

Also, Huffman coding is applied to the compressed data (Coefficients) for further improvement in terms of CR in both FDVSL and VDVSL.

Another observation is that in texture images VDVSL behaves slightly better than FDVSL since it will automatically tries to fit the segments by increasing the degree. In addition, in most cases VDVSL achieves slightly better in term of PSNR than FDVSL.

Note that VDVSL is not as fast as JPEG or FDVSL, but could be improved by using parallel processing. This is possible as the method lends itself readily to parallel processing since it is row or column-based. However, time to reconstruct the image is very fast.

Another improvement to these methods (i.e. VDFSL, FDVSL and VDVSL) can be introduced by fitting in the frequency domain using for example the Fast Fourier Transform (*FFT*).

## 5 Conclusions

A new compression technique based on fitting adaptive variable-degree Chebyshev polynomials to variable length segments of data of the rows or columns of an image is presented. The method offers the user direct control of individual gray level variations in the reconstructed image. The technique was tested on a number of images and the results showed slightly better quality reconstructed images but with lower compression ratios than JPEG. For medical images where the amount of visible distortion cannot be tolerated the new suggested methods performed better because of their ability to control the individual error.

## References

- [1] G. Wallace, "The JPEG Still Picture Compression Standard", *CACM*, vol. 34, pp 33-40, Apr 1990.
- [2] D.M. Bethel, D.M. Monro, "Polynomial image coding with vector quantised compensation", *Proc. ICASSP-95*, vol. 4, 2499-2502.
- [3] Y. Chee, K. Park, "Medical image compression using the characteristics of human visual system", *Proc. 16th Int. Conf. of the IEEE*, Eng. Advances: New Opportunities for Biomedical Eng., vol. 1, 618-619, 1994.
- [4] F. De Natale, *et. al.*, "Polynomial Approximation and Vector Quantization: A Region-Based Approach", *IEEE Trans. On Communications*, vol. 43, no. 2-4, Feb-Apr 1995, pp 198-206.
- [5] A. Helzer, *et. al.*, "Using implicit polynomials for image compression", *Proc. 21st IEEE Conv. of the EEEI*, pp 384-388, 2000.
- [6] I. Sadeh, "Polynomial Approximation of Images", *Computers Math. Applications*, vol. 32, no. 5, pp 99-115, 1995.
- [7] N. Al - Mutawwa, M.J. Zemerly, "Image Compression using Adaptive Chebyshev Polynomials", *WSEAS Trans. On Mathematics*, Vol. 3, issue 2, pp 417-422, April 2004.

- [8] I. Al - Jarwan, M.J. Zemerly, et. al, "Image Compression using Adaptive Fixed-Degree Chebyshev Polynomials", *proc. IEEE-GCC*, Bahrain, Nov 2004.
- [9] M.G. Cox, J.G. Hayes, "Curve Fitting: A guide and suite of Algorithms for the Non-Specialist user", *NPL Report NAC 26*, Dec 1973.
- [10] C.W. Clenshaw, "Mathematical Tables", Volume 5, National Physical Laboratory, London 1962.
- [11] "Numerical Recipes in c: The Art of Scientific Computing", *Cambridge University Press*, 1988-1992.
- [12] M.J. Zemerly, "A rule-based system for processing retinal images", Ph.D. thesis, University of Birmingham, U.K, August 1989.

# Linear Hashtable Method Predicted Hexagonal Search Algorithm with Spatial Related Criterion

Yunsong Wu<sup>1</sup>, Graham Megson<sup>1</sup>, Zhengang Nie<sup>2</sup>, and F.N. Alavi<sup>3</sup>

<sup>1</sup> Computer Science, Reading University,

Reading, UK, RG6 6AA

{sir02yw, G.m.Megson}@rdg.ac.uk

<sup>2</sup> Beihang University,

zhengang.nie@ee.buaa.edu.cn

<sup>3</sup> Computer Science, Queen Mary, University of London,

fna@dcs.qmul.ac.uk

**Abstract.** The paper presents a novel Linear Hashtable Method Predicted Hexagonal Search (LHMPHS) method for block base motion compensation. It bases on the edge motion estimation algorithm called hexagonal search (HEXBS). Most current variances of hexagonal search are investigated. On the basis of research of previous algorithms, we proposed a Linear Hashtable Motion Estimation Algorithm (LHMEA). The proposed algorithm introduces hashtable into motion estimation. It uses information from the current frame. The criterion uses spatially correlated macroblock (MB)'s information. Except for coarse search, the spatially correlated information is also used in inner search. The performance of the algorithm is evaluated by using standard video sequences and the results are compared to current algorithms such as Full Search, Logarithmic Search etc. The evaluation considers the three important metrics: time, compression rate and PSNR.

## 1 Introduction

In this paper, we propose a Linear Hashtable Motion Estimation Algorithm (LHMEA) to predict motion vectors for intra-coding. The objective of our motion estimation scheme is to achieve good quality video with very low computational complexity. Our method attempts to predict the motion vectors using linear algorithm. It uses hashtable method into video compression. After investigating of most traditional and on the edge motion estimation methods, we use latest optimization criterion and prediction search method. Spatially MBs' information is used to generate the best motion vectors. We also combine the LHMEA with HEXBS by motion predictor method. HEXBS is one of the best motion estimation methods currently. The new method improved by us achieves the best results so far; related statistics has been listed in this paper. The main contributions of this paper are (1) Megson introduced hashtable concept into video compression which uses several variables to represent whole MB.<sup>1</sup> This shows a direction for future research. (2) Linear Algorithm is used in video

---

<sup>1</sup> Graham Megson & F.N.Alavi Patent 0111627.6 -- for SALGEN Systems Ltd.

compression. This will improve speed; also leave space for parallel implementation. (3) LHMEA is combined with HEXBS. A new LHMEA predicted hexagonal method is proposed, which makes up for drawback of coarse search of HEXBS. It can also be used for Diamond Search etc. nearly all kinds of similar fast algorithms. (4) Spatially related MB's information is used not only in coarse search but also inner fine search.

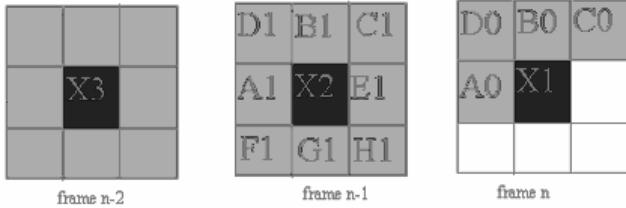
There are a large number of motion prediction algorithms. We only focus on one class of such algorithms, so called the Block Matching Algorithms, which is widely used in MPEG2, MPEG4, and H.263. By partitioning a current frame into non-overlapping macroblocks with equal size, block-matching method attempts to find a block from a reference frame (past or future frame) that best matches a predefined block in the current frame. Matching is performed by moving and comparing with a criterion, which is called the MAE mean absolute error. The MB (macroblock) in the reference frame moves inside a search window centered on the position of the current block in the current frame. The best matched block producing the minimum distortion is found within the search window in the reference frame. However, the motion estimation is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search is used. It is highly desired to significantly speed up the process without sacrificing the distortion seriously. Many computationally efficient variants were developed, among which are typically Two Level Search(TS), Two Dimensional Logarithmic Search(DLS), Subsample Search(SS)[1], the Three-Step search (TSS), Four-Step Search (4SS) [2], Block-Based Gradient Descent Search (BBGDS) [3], and Diamond Search (DS) [4][5] algorithms. A very interesting method called HEXBS has been proposed by Ce Zhu, Xiao Lin, and Lap-Pui Chau [6] in 2002 on IEEE. There are some variant HEXBSs, such as Enhanced Hexagonal method [7], Hexagonal method with Fast Inner Search [8].

## 2 Hexagonal Search Algorithm

HEXBS is an improved method based on DS (Diamond Search). It has shown the significant improvement over other fast algorithms for example DS. In contrast with the DS that uses a diamond search pattern, the HEXBS adopts a hexagonal search pattern to achieve faster processing due to fewer search points being evaluated. The motion estimation process normally comprises two steps. The low-resolution coarse search to identify a small area where the best motion vector is expected to lie, and then followed by fine-resolution inner search to select the best motion vector in the located small region. Most fast algorithms focus on speeding up the coarse search by taking various smart ways to reduce the number of search points in identifying a small area for inner search. There are two main directions to improve the coarse search, first is usage of predictors [8] [9], second is early termination [9]. In [8] a new algorithm was introduced on HEXBS, which is similar as Motion Vector Field Adaptive Search Technique (MVFAS) [10] based on DS. The algorithm has significantly improved the preexisting HEXBS algorithm both in image quality and speed up by initially considering a small set of predictors, namely the (0,0) motion vector and the motion vectors of the three spatially adjacent blocks(left, top, top-right) as possible motion vector predictor candidates.

## 2.1 Enhanced Hexagonal Algorithm with Variant Spatial Related Predictors

Modified Hexagonal pattern used the best motion vector predictor candidate as the center of search. In [9, 10] it was proposed a prediction set. In general, we can state that the blocks correlated with the current one, which are likely to undergo the same motion, can be divided into three categories as in Fig.1.



**Fig. 1.** Blocks correlated with the current one

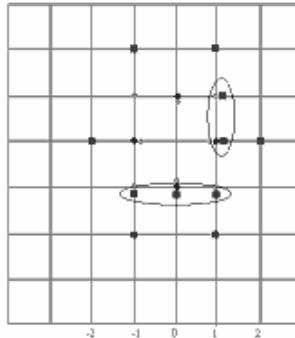
- (1) Spatially correlated blocks (A0, B0, C0, D0),
- (2) Neighboring blocks in the previous frame (A1, B1, C1, D1, E1, F1, G1, H1)
- (3) Co-located blocks in the previous two frames (X2 and X3), which provide the acceleration motion vector (MV).

This last one can enhance temporal prediction in sequence with fast and non-uniform motion. So the MV set composed of the (0,0) vector; the median of the MVs of the left- up- and upright blocks (respectively named A0, B0 and C0 in Fig. 1; 4 neighboring blocks in the current frame; the ones of the co-located block (X2) and of the four vertically (B1, G1) and horizontally (A1, E1) adjacent blocks in the previous frame, and the acceleration motion vector  $MV_{acc} = MV_{X2} + (MV_{X2} - MV_{X1})$

## 2.2 Enhanced Hexagonal Algorithm with Variant Spatially Related Pixels Inner Search

In the original HEXBS algorithm [6], it uses the large hexagon search pattern consisting of six endpoints. After coarse search procedure first locates a region where the optimal motion vector is most expected. The coarse search continues based on a gradient scheme until the center point of the hexagon has the current smallest distortion. After a hexagonal area is located in the coarse search, then the following fine-resolution search looks into the small area enclosed by the large hexagon for focused inner search. The original HEXBS inner search [6] used 4 points. Normal hexagonal inner search now uses 8 points inner search. If full search is required for the inner search, eight search points inside the large hexagon will be evaluated, which is computationally inefficient. Based on the monotonic distortion characteristic in the localized area around the global minimum, it is proposed to check only a portion of the inner search points that are nearer to the checked points with smaller distortions, which can save more than half of the eight search points inside. [8] This is based on knowledge that spatial coherence. Spatially related pixels have similar information and tend to be same in not only Sum of Absolute Difference (SAD) in motion

estimation but also similar motion vectors. It considers forming the 6 endpoints into 6 groups. For each group, a group distortion is defined by summing the distortions of all the points within the group. The area near to the group with the smallest group distortion is considered where the minimum distortion is most likely to be found, as explained in the figure 2.



**Fig. 2.** 8 points hexagonal inner search

In figure 2, (a) Three inner points  $(1,-1)$   $(0,-1)$   $(-1,-1)$  nearest to bottom two endpoints  $(-1,-2)$   $(1,-2)$  of hexagon with the smallest group distortion are to be checked. If the smallest distortion group is bottom two points, three checking points nearest to them will be used in the focused inner search. The top 3 inner points are the same. (b) Two inner points  $(1,0)$   $(1,1)$  nearest to two endpoints  $(2,0)$   $(1,2)$  of hexagon with the smallest group distortion are to be checked. If the smallest distortion group is  $(2,0)$   $(1,2)$ , two inner points nearest to the smallest distortion group will be evaluated in the focused inner search.

The following table is for the statistics to show that inner group search is faster than the other inner search methods when PSNR does not change. In this figure inner group is 6-side-based fast inner search; inner normal is 8 points check; inner square is 4 points check. Normal inner search is the slowest of all.

**Table 1.** Comparison of inner search methods (based on 100 frames of Table Tennis)

Hex predictor	pred_median	pred_median	pred_median
Inner search	Inner group	Inner normal	inner square
Early termination	hex_near	hex_near	hex_near
compression time(P)	1	1	1
Frame per second(fps)	17.5325	15.9763	16.7702
compression Rate(P)	42	42	42
average P frame PSNR	21.2	21.2	21.2

### 2.3 Enhanced Hexagonal Algorithm with Early Termination Using SAD from Spatially Related MBs

Sequences with low or global motion usually have more predictors close to the optimum and provide an acceptable distortion. To take advantage of this situation, early termination criteria can be applied to minimize the number of matches. The chosen threshold takes into account the minimum SAD found for the adjacent blocks and for the current block in the last frame:

$$T = \min(MSAD_{A0}, MSAD_{B0}, MSAD_{C0}, MSAD_{X1}) + npel.$$

Where  $MSAD_i$  is the minimum SAD found for block  $i$ ;  $A0, B0$  and  $C0$  refer to the left, up and upright block respectively,  $X1$  is the current block in the previous frame and  $npel$  is the number of pixels in the block. As in the figure below, the early termination can increase about 10% of compression speed. In the Table 2, hex max means without early termination, hex near is with early termination.

**Table 2.** Comparison of normal and early termination method (100 frames of Table Tennis)

Hex predictor	pred_hashtable	pred_hashtable
Inner search	inner group	inner group
Early termination	hex_max	hex_near
compression time(P)	2	2
Frame per second(fps)	10.9312	12.1076
Compression rate(P)	14	14
average P frame PSNR	27.5	27.5

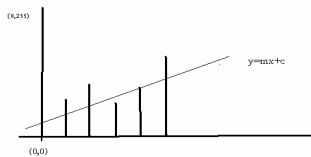
### 3 Linear Hashtable Method Predicted Hexagonal Search (LHMPS)

Most of current hexagonal search algorithms predictive methods focus on relations between current frame and previous frames. What we want to do is to find a fast method which discovers the predictors from current frame information. It uses spatially related MB or pixels' information. It is fast, accurate and independent on finding right predictors. So we designed a vector hashtable lookup matching algorithm which is more efficient method to perform an exhaustive search: it considers every macroblock in the search window. This block-matching algorithm calculates each block to set up a hashtable. It is a dictionary in which keys are mapped to array positions by a hash function. We try to find as few as possible variables to represent the whole macroblock. Through some preprocessing steps, “integral projections” are calculated for each macroblock. These projections are different according to different algorithm. The aim of these algorithms is to find best projection function. The

algorithms we present here has 2 projections, one of them is the massive projection, which is a scalar denoting the sum of all pixels in the macroblock. It is also DC coefficient of macroblock. Another is A of Y=Ax+B (y is luminance, x is location.) Each of these projections is mathematically related to the error metric. Under certain conditions, the value of the projection indicates whether the candidate macroblock will do better than best-so-far match. The major algorithm we discuss here is linear algorithm

### 3.1 Linear Hashtable Motion Estimation Algorithm (LHMEA)

Linear Algorithm is most beautiful, easy and fast to calculate on computer because the construction of computer calculator bases on additions. So if most of calculations of video compression are done by linear algorithm, we can save lots of time on compression. It is also very easy to put on parallel machines in the future, which will benefit real time encoding. In the program, we try to use polynomial approximation to get such result  $y=mx+c$ ; y is luminance value of all pixels, x is the location of pixel in macroblocks. The way of scan y is from left to right, from top to bottom. Coefficients m and c are what we are looking for. As in the figure 3



**Fig. 3.** Linear algorithm for discrete algorithm

In this function  $y=f(x)$ , x will be from 0 to 255 in a macroblock,  
 $y=f(x)=mx+c$

$$m = \frac{N * \sum_{i=0}^N (x_i * y_i) - \sum_{i=0}^N x_i * \sum_{i=0}^N y_i}{N * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i}$$

$$c = \frac{\sum_{i=0}^N y_i * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i * y_i}{N * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i}$$

In this way, we initially realize the way to calculate the hashtable. In previous research methods, when people try to find a block that best matches a predefined block in the current frame, matching was performed by SAD (calculating difference

between current block and reference block). In Linear Hashtable Motion Estimation Algorithm (LHMEA), we only need compare two coefficients of two blocks. In current existing methods, the MB moves inside a search window centered on the position of the current block in the current frame. In LHMEA, the coefficients move inside hashtable to find matched blocks. If coefficients are powerful enough to hold enough information of MB, motion estimators should be accurate. So LHMEA increases lots of speed, accuracy and will make a new era of video encoding.

### 3.2 Linear Hashtable Method Predicted Hexagonal Method

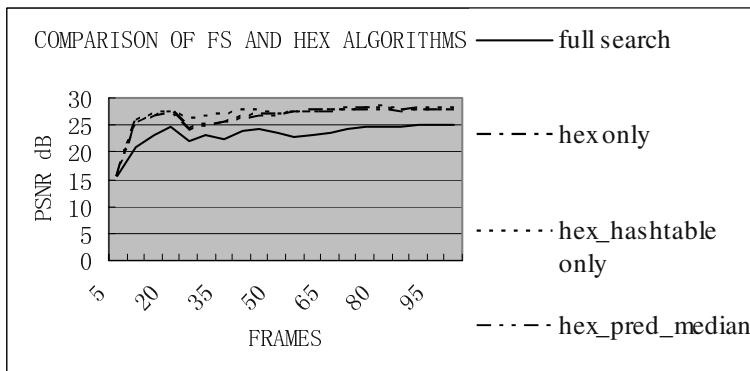
After motion estimators are generated by LHMEA, they will be used as predictors for HEXBS for coarsely search. These predictors are different from all previous predictors. They are based on full search and current frame only. Because LHMEA is linear algorithm, it is fast. Because the predictors generated are accurate, it will improve HEXBS without too much delay in speed.

In the Figure below, we compared Full Search (FS), Linear Hashtable Motion Estimation Algorithm (LHMEA), Subsample Search(SS), Two Level Search(TLS), Logarithmic Search(LS) and three kinds of HEXBS Algorithms. Three HEXBS Algorithms are Hexagonal Search without predictor(HEXBS), LHMPHS and Hexagonal with median predictors of spatially adjacent blocks ( left, up and upright blocks what are respectively named A0, B0 and C0 in Fig. 1) (HSM)[10]. All HEXBS algorithms used 6-side-based fast inner search [9] and early termination criteria [10] mentioned in our paper. All the data here refer to P frames only. HEXBS can achieve nearly the same PSNR as FS and only takes 10% time of FS. The LHMPHS is better than HEXBS without predictor on compression rate when time and PSNR are the same. HSM is the best algorithm of three. But if we can find better coefficients in the hashtable to represent MB, the hashtable will have a wonderful future.

**Table 3.** Comparison of compression rate, time and PSNR between FS, LS, SS, TLS, HEXBS, LHMPHS, HSM (based on 100 frames of Table Tennis)

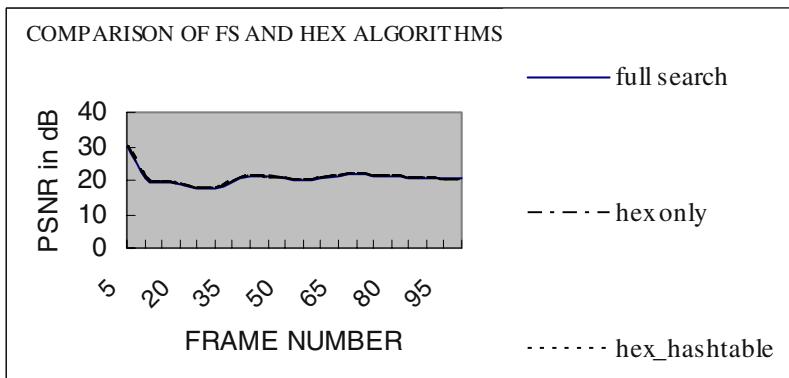
Search Method	EXHAUSTIVE	LOGARITHMIC	SUBSAMPLE	TWOLEVEL	HexNo Predictor	Pred _Hashtable	Pred _Median
Inner Search Early Termination Compression Time(s) (fps)	11 2.3726	1 24.7706	4 6.4286	3 7.3171	Inner Group Hex_near 1 19.4245	Inner Group Hex_near 1 14.2857	Inner Group Hex_near 1 17.5325
Compression Rate	48	42	48	48	37	39	42
PSNR	21.3	21.1	21.3	21.3	21.2	21.2	21.2

The figure below shows the PSNR from 5 to 100 frames of flow garden data stream among FS, HEXBS, LHMPHS, HSM. It shows the LHMPHS has better PSNR than the other algorithms.



**Fig. 4.** Comparison of PSNR between FS, HEXBS, LHMPHS, HSM(based on 5-100 frames of Flower Garden)

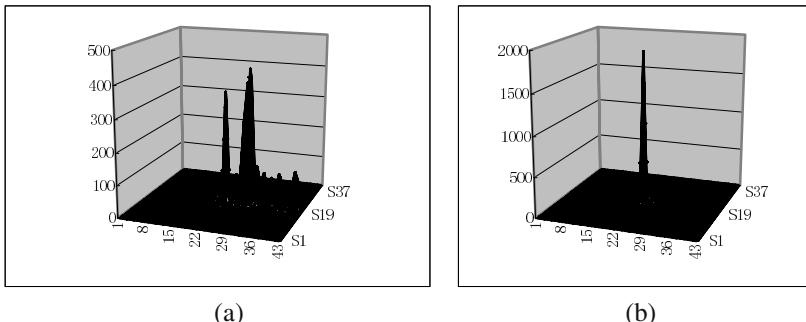
But in the following figure about PSNR from 5 to 100 frames of table tennis data stream, all algorithms' PSNR are the same. It means LHMEA works better on large motion vector video stream.



**Fig. 5.** Comparison of PSNR between FS, HEXBS, LHMPHS, HSM (based on 5-100 frames of Table Tennis)

The FS, HEXBS, LHMPHS are certain center biased algorithms. This is also basis of several other algorithms. It was based on the fact that for most sequences motion vectors were concentrated in a small area around the center of the search. This can also be seen in the figures below. Unfortunately for some sequences this is not always true as can be seen in the flower garden figure 6 (a), which implies that these algo-

rithms will have reduced performance in such cases. From additional simulations we can observe that predictor seems to have much higher correlation with the current motion vector than (0,0) even for non-center biased sequences such as the flower garden mentioned previously. This suggests that, instead of initially examining the (0,0) position, we can achieve better results if the linear hashtable predictor is examined first and given higher priority with the usage of an early termination threshold.



**Fig. 6.** Motion vector distribution in (a) Flower Garden and (b) Table Tennis by LHMEA

## 4 Summary

In the paper we proposed a new algorithm called Linear Hashtable Motion Estimation Algorithm (LHMEA) in video compression. It uses linear algorithm to set up hashtable. The algorithm searches in hashtable to find motion estimator instead of by FS. Then the motion estimator it generates will be sent to HEXBS, which is best motion estimation algorithm, as predictor. No matter in coarse search or fine inner search, new method used lots of spatial related MB or pixels' information. In this way, it improves both quality and speed of motion estimation. The key point in the method is to find suitable coefficients to represent whole MB. The more information the coefficients in hashtable hold about pictures, the better result LHMPHS will get. This also leaves space for future development.

## References

1. Ze-Nian li Lecture Note of Computer Vision on personal website (2000)
2. L. M. Po and W. C. Ma: A novel four-step search algorithm for fast block motion estimation," IEEE and systems for video technology, vol. 6, pp. 313–317, (June 1996.)
3. L. K. Liu and E. Feig: A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 419–423, (Aug. 1996.)
4. S. Zhu and K.-K. Ma: A new diamond search algorithm for fast blockmatching motion estimation: IEEE Trans. Image Processing, vol. 9, pp. 287–290, (Feb. 2000.)

5. J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim: A novel unrestricted center-biased diamond search algorithm for block motion estimation: IEEE Trans. Circuits and systems for video technology, vol. 8, pp. 369–377, (Aug. 1998)
6. Ce Zhu, Xiao Lin, and Lap-Pui Chau: Hexagon-Based Search Pattern for Fast Block Motion Estimation: IEEE Trans on circuits and syst. for video technology, Vol. 12, No5, (May 2002)
7. C. Zhu, X. Lin and L.P. Chau: An Enhanced Hexagonal Search Algorithm for Block Motion Estimation: IEEE International Symposium on Circuits and Systems, ISCAS2003, Bangkok, Thailand, (May 2003)
8. Ce Zhu, Senior Member, IEEE, Xiao Lin, Lappui Chau, and Lai-Man Po: Enhanced Hexagonal Search for Fast Block Motion Estimation: IEEE Trans on circuits and systems for video technology, Vol. 14, No. 10, (Oct 2004)
9. Paolo De Pascalis, Luca Pezzoni, Gian Antonio Mian and Daniele Bagni: Fast Motion Estimation With Size-Based Predictors Selection Hexagon Search In H.264/AVC encoding: EUSIPCO (2004)
10. Alexis M. Tourapis, Oscar C. Au, Ming L. Liou: Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) Enhancing Block Based Motion Estimation: proceedings of Visual Communications and Image Processing, San Jose, CA, January (2001)
11. A. M. Tourapis, O. C. Au and M. L. Liou: Highly Efficient Predictive Zonal Algorithms for Fast Block Matching Motion Estimation: IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, No.10, pp 934-947, (October 2002)

# Fractal Dimension Analysis and Statistical Processing of Paper Surface Images Towards Surface Roughness Measurement

Toni Kuparinen<sup>1</sup>, Oleg Rodionov<sup>1</sup>, Pekka Toivanen<sup>1</sup>, Jarno Mielikainen<sup>1</sup>,  
Vladimir Bochko<sup>1</sup>, Ate Korkalainen<sup>2</sup>, Juha Parviaisen<sup>2</sup>, and Erik Vartiainen<sup>2</sup>

<sup>1</sup> Department of Information Technology

<sup>2</sup> Laboratory of Physics,

Lappeenranta University of Technology,

P.O. Box 20, FIN-53851 Lappeenranta, Finland

[tkuparin@lut.fi](mailto:tkuparin@lut.fi), [olerodi@hotmail.com](mailto:olerodi@hotmail.com),

{ptoivane, mielikai, botchko, akorkala, jparviai, emvartia}@lut.fi

**Abstract.** In this paper we present a method for optical paper surface roughness measurement, which overcomes the disadvantages of the traditional methods. Airflow-based roughness measurement methods and profilometer require expensive special equipment, essential laboratory conditions, are contact-based and slow and unsuitable for on-line control purposes methods. We employed an optical microscope with a built-in CCD-camera to take images of paper surface. The obtained image is considered as a texture. We applied statistical brightness measures and fractal dimension analysis for texture analysis. We have found a strong correlation between the roughness and a fractal dimension. Our method is non-contact-based, fast and is suitable for on-line control measurements in the paper industry.

## 1 Introduction

The roughness of paper surface is an important parameter in paper manufacturing and its measurement is one of central measurement problems in paper industry. The surface roughness affects the major paper property — the printing resolution, i.e. the capability to transfer without breaks and distortions the most thin printed lines, dots and their combinations. Eventually it defines the quality of a final printed product.

At present time standardized and employed in the paper industry roughness rating methods are based on measurements with the help of airflows aimed at tested surfaces. By means of specialized pneumatic devices under laboratory conditions the airflow rate between measured paper surface and a specified flat land is recorded. Such a measurement closely corresponds to the roughness of a surface, the less required the time — the rougher the surface is. The measured roughness is given in micrometers or milliliters per second according to Parker Print-Surf (PPS) and Bendtsen methods respectively [1].

Airflow-based roughness measurement methods have certain disadvantages; central ones are a poor accuracy and incompatibility to perform measurements of fine and smooth papers. Contrary to airflow-based methods, there is a possibility to inspect a surface by means of very accurate electronic profilometer devices [2]. Both described methods require expensive special equipment, essential laboratory conditions, are contact-based and slow and unsuitable for on-line control purposes methods.

Moreover, the abovementioned methods provide information on the physical roughness of a measured surface, i.e. on the microrelief and microgeometry regularity and uniformity. On the other hand, we can think of an optical roughness, that in turn refers to how paper scatters the light off the surface and appears to the observer, i.e. it characterizes optical paper properties instead. Further, we can imagine an extra roughness rating method that is intended to produce measurements of the optical paper roughness by utilization of machine vision techniques combined together with image processing.

Light transmission image analysis has been previously applied to paper formation and quality measurements [3, 4, 5]. Studied machine vision techniques were co-occurrence matrix, Fourier method and wavelets. In this research scattering light is detected and statistical image processing and fractal image analysis have been used. Fractals were first introduced in 1977 by Mandelbrot [6] and developed further in the end of 1980's by Barnsley [7]. The fractal dimension is a measure of the complexity of a fractal and a review of methods can be found in [8] Kent [9] and Johansson [10] measured paper surface roughness using profilometry and studied the fractal nature of the measured profile. According to Kent [9] paper surface topography exhibits fractal characteristics. Later on, Johansson [10] also made the same conclusion using different paper grades. Contrary to the previous methods, our approach is based on an acquired microscope image and a linear dependency between the measured fractal dimension and conventional roughness has been found. Gopalakrishnan [11] applied similar fractal dimension estimation computation as we did for online monitoring of surface roughness. The difference is that they estimated the roughness of metallic surface, to which a different kind of physical reflection model has to be applied than to paper samples [12].

The article is organized as follows: in Chapter 2 we propose the method and in Chapter 3 the image processing is presented. In Chapter 4 are the experiments and in Chapter 5 the conclusions are drawn.

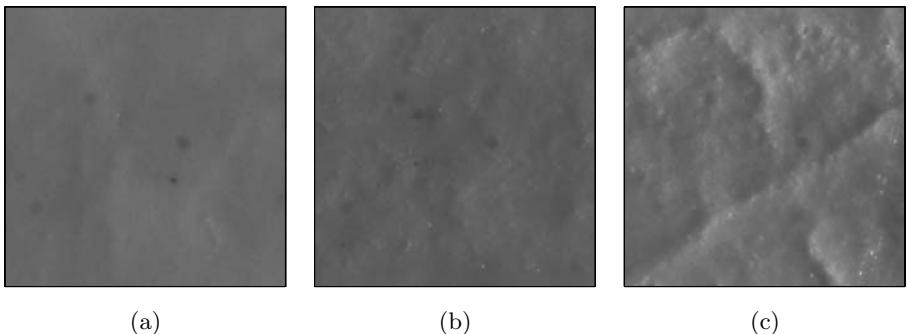
## 2 Proposed Method

We have developed a method based on an idea that the roughness of paper surface can be analyzed using an image formed from the light which is reflected off paper surface. Smooth papers appear to be more monotonic in terms of the reflected light spatial intensity, than rough ones that have a strong spatial variation of the reflected light intensity.

In our method the measured surface is illuminated by grazing light. The scattering light from the sample is captured by a microscope with a built-in CCD-camera and digitized by a frame grabber. The produced digital gray-scale image is processed by a computer image processing, which provides an image measure characterizing the roughness of original surface. As in airflow methods several images of the one paper surface has to be taken and processed in order to average image measures and estimate the roughness more accurately. We have proposed statistical image processing and a fractal dimension image analysis for image processing.

### 3 Image Processing

According to the classification given in [13], we merely consider microscope paper surface images as gray-scale stochastic textures. Fig. 1 (a–c) depicts typical different roughness paper surface images.



**Fig. 1.** Paper surface images taken by a microscope of (a) low, (b) medium and (c) high roughness

Consequently, a suitable image processing approach is statistical image processing that gives image statistical measures. Another approach is to perform a fractal dimension image analysis that estimates the complexity of image.

#### 3.1 Statistical Processing

We compute a set of statistical brightness measures of a gray-scale texture image representing an inspected paper surface. The following statistical measures from the brightness histogram are computed: mean, variance, skewness, kurtosis, entropy of the brightness.

The skewness factor  $s$  is a degree of symmetry, or more precisely, lack of symmetry. The kurtosis factor  $k$  is a measure of whether a distribution is peaked or flat. The entropy [14] characterizes the compressibility of image and is measured

in bits per pixel. The lower the entropy — the less the number of bits needed for coding.

The main disadvantage of previous mentioned statistical measures is that they do not take into consideration the spatial brightness distribution in the image. Therefore measures like 2D kurtosis and fractal dimension are utilized.

The 2D kurtosis is an extension of kurtosis to multidimensional data and it was introduced by Mardia [15]. Johansson [10] applied 2D kurtosis  $b_{2,2}^*$  to gray-scale images as a measure of homogeneity of a paper surface.

### 3.2 Fractal Dimension Analysis

Microscope taken paper surface images appear as gray-scale stochastic textures that have complex spatial brightness distributions. With the help of fractal dimension analysis we have tried to overcome the abovementioned disadvantage of statistical image processing and take into account the brightness spatial distribution.

In the 1980's a discussion of fractals and chaos introduced an idea to consider texture images as fractals, [6, 7]. It leads towards a consideration of a digitized image  $\mathbf{G}$  as a discrete set of points  $\mathbf{S}$ , each representing a particular pixel in the original image:

$$\mathbf{S} = \{s_i\}, \quad i = 1, 2, \dots, N. \quad (1)$$

The fractal dimension  $d$  can be computed for the set  $\mathbf{S}$  that measures the complexity of a set. It is motivated in general by a power law relationship, if this exists, between two measures.

Let us consider a measure  $N$  that is estimated by taking measurements at a scale  $\epsilon$ . If we change the scale  $\epsilon$ , most probably we get another value of the measure  $N$ , thus  $N = N(\epsilon)$ . However, it is possible to find a constant  $d$  that yields an equation:

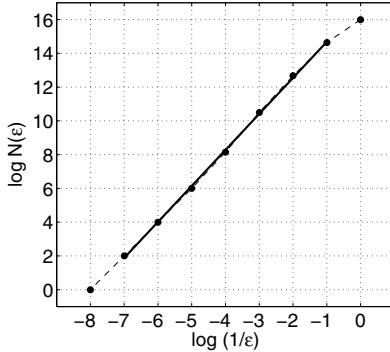
$$N(\epsilon) \sim \frac{1}{\epsilon^d}. \quad (2)$$

The constant  $d$  is the fractal dimension and not usually integer. The fractal dimension  $d$  shows how rapidly the measure  $N$  grows, while the scale  $\epsilon$  decreases. Thereby, the fundamental idea is that the measure  $N$  and scale  $\epsilon$  do not vary arbitrary, but are related by a power law relationship according to Eq. 2.

To find the fractal dimension  $d$ , we solve Eq. 2 for  $d$  taking a limit as  $\epsilon$  approaches to zero:

$$d = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}. \quad (3)$$

We can try to find the fractal dimension  $d$  for a given finite set  $\mathbf{S}$  representing an image with the help of the above described approach and solving Eq. 3. However, we will run against a problem as the limit will always yield zero, because eventually the scale  $\epsilon$  will be so small that the measure  $N(\epsilon)$  gets the maximum value, while  $\log(1/\epsilon)$  grows without a bound as the scale  $\epsilon$  tends zero.



**Fig. 2.** Linear regression for fractal dimension estimation

In order to estimate the fractal dimension  $d$  of a finite set  $S$ , we must calculate values of the measure  $N(\epsilon)$  for some range of the scale  $\epsilon$  and take as our estimate for the fractal dimension  $d$  the slope of the straight line minimizing the mean-square deviation of  $\log N(\epsilon)$  vs.  $\log(1/\epsilon)$ , see Fig. 2. The problem of minimization of the mean-square deviation has a solution that is given as linear regression.

The most popular and efficient way to compute the fractal dimension is through the capacity dimension by box-counting [16]. The box-counting approach to compute the fractal dimension gives a systematic measurement procedure that applies to any structure in a texture plane and can be adapted for structures in multi-dimensional spaces [17]. The idea of the approach is to put an image onto a regular rectangular grid of the size  $\epsilon$  and simply count a number of grid cells – ‘boxes’ – containing some of image texture. This gives the measure  $N(\epsilon)$ . Then, we change the grid size  $\epsilon$  to progressively smaller sizes and produce a series of measures  $N(\epsilon)$ . Finally, we try to fit a straight line as it is described above, the fractal dimension estimate comes as the slope of the line.

For practical purposes it is often apt to consider a sequence of grids, where the size reduces by a factor of 2 consequently [18]. In such a case each cell is subdivided into four cells each of half size in the next grid. When box-counting using such grids, we arrive at a sequence of counts  $N(2^i)$ ,  $i \in Z^+$ .

In order to estimate the fractal dimension  $d$  of the whole texture image a linear regression should be employed as mentioned above.

**Fractal Dimension of a Gray-Level Image.** We have used an extension of the box-counting approach suggested in [19]. The idea of such an approach is to treat a gray-scale image  $G$  as a three-dimensional discrete set of points  $S$ , where first two components  $s_0$  and  $s_1$  are the  $x$  and  $y$  coordinates of pixels, and the third component  $s_2$  is the brightness  $b$  of pixels. As a result, we deal with a  $(x, y, b)$ -space.

Let us consider a gray-scale image  $G$  of the size of  $256 \times 256$  pixels and with 256 gray levels. Then, we can produce a three-dimensional array  $S$  of elements

of the size of  $256 \times 256 \times 256$  representing the image  $\mathbf{G}$ : we zero all elements of  $\mathbf{S}$  except those, whose indexes equal to the coordinates of gray-scale image pixels.

Now let us consider a three-dimensional square grid of the size  $\epsilon$  of one element, see Fig 3 (a) and (b). We find a number  $N(\epsilon)$  of non-zeros elements. Then, we sum the array at a time across each of three dimensions, that is equivalent of increasing the grid size  $\epsilon$  up to the size of two elements, and find a new  $N(\epsilon)$ , see Fig 3 (c) and (d). We proceed with such a procedure, until the grid size  $\epsilon$  increases up to the size of the whole array. We arrive at a sequence of grids of the sizes  $\epsilon_i$  and corresponding measures  $N(\epsilon_i)$ , whereupon we apply the linear regression in order to compute the fractal dimension we refer to as  $d$ . We have to mention that ending points should be dropped due to a phenomenon described in [18]. See Fig. 2 for the illustration. The fractal dimension  $d$  characterizes the complexity of the texture of a gray-scale image.

## 4 Experimental Setup and Results

We have implemented all necessary computational algorithms for statistical image processing and image fractal dimension analysis in Matlab and run all experiments under the Matlab environment.

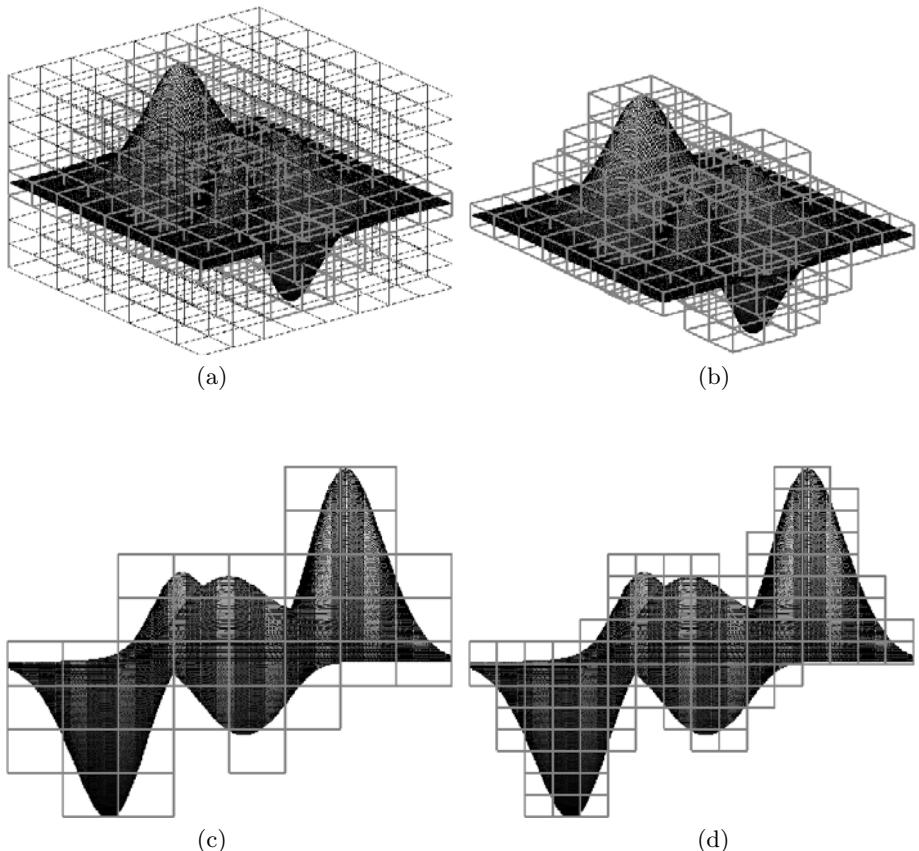
In our set of experiments we have used three test sets of paper samples. In every set we have produced 10 gray-scale microscope images of each paper specimen. Each image is averaged from 5 images from the same area to reduce white noise. The size of images was  $512 \times 512$  pixels with 8 bits per pixel and a image shows an area of size  $0.3\text{mm} \times 0.3\text{mm}$  of the paper surface. For each test set illumination was adjusted according to paper brightness and roughness and then held constant during the measurement of a set.

The results from the first test set are reported in more detail to illustrate the method. The reported values for paper specimens are the average values of a measurement from 10 images. From all the test sets the correlation of roughness and each measure are reported.

In the first set, denoted as UPM A, there are 6 coated, calendered paper specimens of the different roughness measured by PPS. The roughness of paper specimens appear to possess values of 0.63, 0.66, 0.67, 0.71, 0.91 and  $0.95\mu\text{m}$ .

In the second set, denoted as UPM B, there are 6 coated, calendered samples of two different paper grades measured by PPS. The first three samples are for gravure printing and the roughness of paper specimens are  $0.62$ ,  $0.67$ ,  $0.70\mu\text{m}$ . The last three samples are for offset printing with the roughness values  $0.99$ ,  $1.08$ ,  $1.08\mu\text{m}$ .

In the third set, denoted as Stora Enso, there are 2 samples of base board, a sample of coated board, a sample of uncoated paper and a sample of coated paper measured by Bendtsen. The roughness of base boards are  $303$  and  $447\text{ ml/min}$ , coated board  $11\text{ ml/min}$ , uncoated paper  $228\text{ ml/min}$  and coated paper  $8\text{ ml/min}$ .



**Fig. 3.** 3D Box-counting algorithm: (a) regular rectangular grid over the image, (b) grid cells containing some image texture (denoted as gray boxes), (c-d) 2D presentation of different grid sizes  $\epsilon$ .

#### 4.1 Results from Statistical Processing

We have computed statistical measures of our test set of images and the correlation with the physical roughness. See Table 1 for image statistics averaged for each paper specimen of UPM A test set.

We have found out that some of statistical measures show the strong correlation with the roughness for some test sets, see Table 2. For instance the brightness mean has a high correlation 0.98 and 0.91 in test sets UPM A and UPM B, respectively. Unfortunately, in test set Stora Enso the correlation is only 0.73. The average correlation of the three test sets for the brightness mean, variance and 2D kurtosis were 0.87, 0.87 and 0.88, respectively. These three were the best statistical measures.

**Table 1.** Statistical measures: mean  $\mu$ , variance  $\sigma^2$ , skewness  $s$ , kurtosis  $k$ , entropy  $\eta$  and 2D kurtosis  $b_{2,2}^*$  and fractal dimension  $d$ 

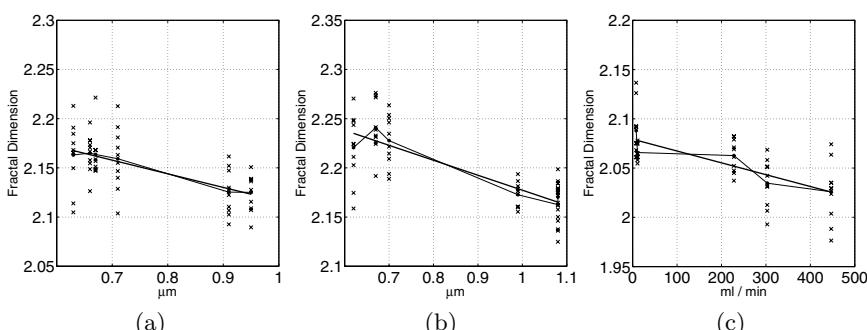
$r, \mu\text{m}$	$\mu$	$\sigma^2$	$s$	$k$	$\eta$	$b_{2,2}^*$	$d$
0.63	98.22	14.910	0.901	14.33	3.929	347.3	2.163
0.66	98.61	16.170	0.619	9.880	3.998	348.1	2.165
0.67	98.59	18.080	0.915	12.29	4.051	345.9	2.164
0.71	98.09	16.850	0.847	9.289	4.016	347.3	2.159
0.91	99.79	25.410	1.892	28.15	4.253	354.9	2.125
0.95	101.9	29.080	1.674	16.19	4.316	359.2	2.125

**Table 2.** Correlation between different image measures and the roughness

Measure	UPM A	UPM B	Stora Enso
$\mu$	0.98	0.91	0.73
$\sigma$	0.94	0.84	0.83
$s$	0.69	0.77	0.81
$k$	0.98	0.49	0.66
$\eta$	0.98	0.56	0.81
$b_{2,2}^*$	0.96	0.90	0.77
$d$	-0.99	-0.96	-0.91

## 4.2 Results from Fractal Dimension Analysis

We have computed fractal dimension  $d$  explained above for test sets. See the last column of Table 1 for numerical results of UPM A test set. We have revealed that microscope paper surface images are fractal: a logarithmic scale plot of box-counting perfectly fits a straight line, see Fig. 2 for an example fit of a paper surface image from UPM A test set.

**Fig. 4.** (a–c) Fractal dimension  $d$  vs. the roughness (curved lines) and least-square lines (straight lines) of test sets UPM A, UPM B and Stora Enso, respectively

The fractal dimension computation has turned out that it can be efficiently used in order to estimate the surface roughness and leads to the high correlation with the latter. See Table 2 for the obtained correlation values and plots in Fig. 4 (a–c) of fractal dimension  $d$ . In a case of fractal dimension  $d$  we have received in average the highest correlation,  $-0.95$ , in our set of experiments. The average correlation is significantly higher for fractal dimension than for statistical measures.

In Fig. 4 (a–c) the variance of fractal dimension measurements seems relatively high due to scaling of the vertical axis. However, the variance of 10 fractal dimension and PPS measurements are similar; the variance divided by the mean of measurements is 0.02 and 0.04, respectively. Therefore, fractal dimension may be successfully used in the paper surface roughness estimation. However, the current method requires calibration of measurement device for each paper type.

## 5 Conclusions

We have studied the possibility to utilize machine vision techniques in order to perform measurements of the roughness of paper surface. We have used images taken by a microscope with a built-in CCD-camera and employed statistical image processing and image fractal dimension analysis.

We have found that the proposed method has a great potential and performs well compared to standardized and industrially employed airflow-based methods. The proposed method overcomes the disadvantages of existing methods, namely the need for expensive special pneumatic equipment, essential laboratory conditions and time consumption to carry out measurements. Our method is also non-contact-based, fast and is suitable for on-line control measurements in the paper industry.

The obtained results are promising and indicate that the employed image taking and processing can successfully be used in order to measure the paper surface roughness. We have found a set of image measures, which show strong correlation with the roughness of original surface.

We have studied the correlation between the roughness and image statistics. We found out that the image brightness variance, mean and 2D kurtosis performs reasonably well for some test sets, but unfortunately not for all.

The best results and correlation have been achieved by applying fractal dimension image analysis. We have investigated that microscope paper surface images are fractal and image fractal dimension depends on the roughness of the original surface and show high correlation values for all applied test sets.

In our experiments we have used and worked with uncoated and coated, calendered paper samples and cardboards and the suitability of our method for even more diverse types of papers and cardboards is under research. In the future the method will be developed so that, there will be no need for a calibration of measurement device for each paper type. The ultimate goal is to perform the measurements online on a paper machine.

## Acknowledgements

The authors gratefully appreciate the provided funding from European Regional Development Fund (ERDF) and National Technology Agency of Finland (TEKES). Special thanks come to the Laboratory of Physics of Lappeenranta University of Technology, Finland for the help in microscope image acquisition and UPM and Stora Enso for providing paper samples.

## References

1. ISO 8791/1-1986 (E): (Paper and board - determination of roughness and smoothness ( air leak methods). part 1: General method)
2. Wagberg, P., Johansson, P.: Surface profilometry - a comparison between optical and mechanical sensing on printing papers. *Tappi Journal* **76** (1993) 15–121
3. Cresson, T., Luner, P.: The characterization of paper formation. part 2: The texture analysis of paper formation. *Tappi* **73** (1990) 175–184
4. Bernie, J., Douglas, W.: Local grammage distribution and formation of paper by light transmission image analysis. *Tappi* **79** (1996) 193–202
5. Bouydain, M., Colom, J., J., P.: Using wavelets to determine paper formation by light transmission image analysis. *Tappi* **82** (1999) 153–158
6. Mandelbrot, B.: *The Fractal Geometry of Nature*. Freeman and Company (1977)
7. Barnsley, M.: *Fractals Everywhere*. Academic Press Inc. (1988)
8. Bisoi, A.K., Mishra, J.: On calculation of fractal dimension of images. *Pattern Recognition Letters* **22** (2001) 631–637
9. Kent, H.J.: The fractal dimension of paper surface topography. In: TAPPI/CPPA International Printing and Graphics Arts Conference, Vancouver, Canada (1990) 73–78
10. Johansson, J.O.: Models of Surface Roughness with Applications in Paper Industry. PhD thesis, Lund Institute of Technology (2002)
11. Gopalakrishnan, S.: Development of a prototype system for on-line monitoring of surface roughness using fractal geometry. Master's thesis, Institute of System Research (1994)
12. Klinker, G.J.: A physical approach to color image understanding. A K Peters, Wellesley, Massachusetts (1993)
13. Visa, A.: Texture Classification and Segmentation Based on Neural Network Methods. PhD thesis, Helsinki University of Technology (1999)
14. Shannon, C.E.: A mathematical theory of communication. *Bell System Technical Journal* **vol. 27** (1948) 79–423,623–656
15. Mardia, K.V., Kent, J.T., Bibby, J.M.: *Multivariate Analysis*. 2nd edn. Academic Press, London, UK (1980)
16. Georgilli, A., et al.: An efficient procedure to compute fractal dimensions by box counting. *Physical Letters A* **115(5)** (1986) 202–206
17. Peitgen, H., et. al: *Chaos and Fractals: New Frontier of Science*. Springer–Verlag (1992)
18. Liebovitch, L.S., Toth, T.: A fast algorithm to determine fractal dimensions by box counting. *Physical Letters A* **141(8,9)** (1989) 386–390
19. Gagnepain, J., RoQues-Carmes, C.: Fractal approach to two-dimensional and three-dimensional surface roughness. *Wear* **109** (1986) 119–126

# Estimation of Critical Parameters in Concrete Production Using Multispectral Vision Technology

Michael E. Hansen<sup>1</sup>, Bjarne K. Ersbøll<sup>1</sup>, Jens M. Carstensen<sup>2</sup>,  
and Allan A. Nielsen<sup>1</sup>

<sup>1</sup> Informatics and Mathematical Modelling (IMM),  
Richard Petersens Plads, Building 321,  
Technical University of Denmark

<sup>2</sup> Videometer A/S, Lyngsø Allé 3,  
DK-2970 Hørsholm  
[meh@imm.dtu.dk](mailto:meh@imm.dtu.dk)

**Abstract.** We analyze multispectral reflectance images of concrete aggregate material, and design computational measures of the important and critical parameters used in concrete production. The features extracted from the images are exploited as explanatory variables in regression models and used to predict aggregate type, water content, and size distribution. We analyze and validate the methods on five representative aggregate types, commonly used in concrete production. Using cross validation, the generated models prove to have a high performance in predicting all of the critical parameters.

1 Introduction

The importance of concrete in modern society plays an increasing role. Look around you and you will find concrete structures everywhere. Concrete is a composite material which is made up of a ... and a ... [1]. The binder (or cement paste) "glues" the filler together to form a synthetic conglomerate. The constituents used for the binder are cement and water, while the filler can be a fine or coarse aggregate.

(cement + water) + aggregate = concrete

The key ingredient in concrete production is water. When mixed with cement, water forms a paste that binds the aggregate together. The water causes the hardening of concrete through a process called hydration; a chemical reaction in which the major compounds in cement form chemical bonds with water molecules and become hydrates or hydration products. The role of water is important because the . . . . . is the most critical factor in the production of "perfect" concrete. Too much water reduces concrete strength, while too little will make the concrete unworkable. Concrete needs to be workable so that it may

be consolidated and shaped into different forms (i.e. walls, domes, etc.). Because concrete must be both strong and workable, a careful balance of the cement to water ratio is required when making concrete.

Although a small amount of water is needed to complete the chemical reaction with cement, additional water is necessary to make the concrete workable. As the paste is thinned out with water, its quality is lowered. It will have less strength and less durability. For quality concrete a proper proportion of water and cement is essential. This proportion is called water-cement ratio. The water-cement ratio is determined by dividing the weight in kilograms of the total actual mixing water by the weight in kilograms of cement used in the mix. Especially, when the aggregate is sand or finer particles, an unspecified amount of water is always present. When adding additional water to the concrete, this water has to be taken into account, in order to obtain the optimal water-cement ratio. The "latent" amount of water can vary from 1–10% of the weight (e.g. approximately 100 liters of water distributed evenly around all grains in 1 m<sup>3</sup> of aggregate material). Another critical parameter is the size distribution, or gradation, of the aggregate. This is one of the most influential aggregate characteristics in determining how the concrete will perform. It helps determine almost every important property including stiffness, stability, durability, permeability, workability, fatigue resistance, frictional resistance and resistance to moisture damage.

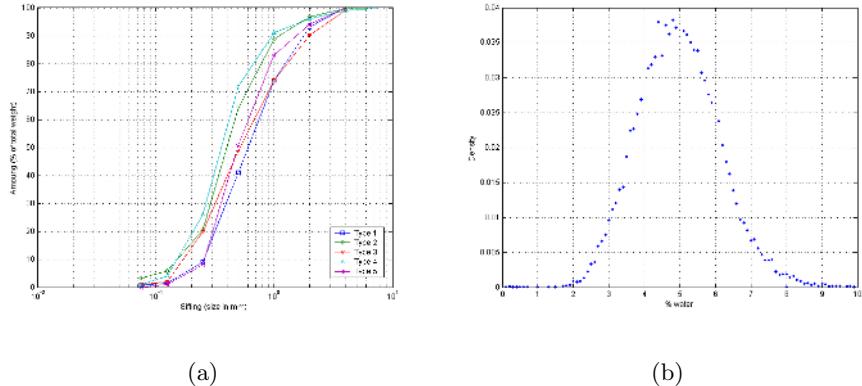
In order to get the optimal concrete mixture, the critical parameters have to be accounted for online in the concrete production. It is the objective of this study to evaluate the use of vision technology as a tool to estimate the following important critical parameters of aggregate samples: 1) The water content, and 2) the aggregate size distributions.

## 2 Materials

### 2.1 Sample Preparation

The experiment was based on five different types of aggregates (sand/gravel), commonly used when producing concrete. They were chosen to represent as different types as possible: different origin (from lake or hill), different preparation (washed or not), and finally the sand types should be classified for use in different environments.

The next step was to prepare the aggregates to have desired size distributions with desired water contents. First, the samples were prepared to have specific size distributions. It was decided to adapt to the current methods used in the production line. Therefore, three types of size distributions were chosen ("Fine", "Normal" and "Coarse"). One for each aggregate type. To accommodate the fact, that the technician work needed for this task is exhausting, aggregate material having "Fine" and "Coarse" size distributions were only prepared for the types 1, 3, and 5, whereas the types 2 and 4 only were represented by the "Normal" size distribution. This choice were based on knowledge about the ma-



**Fig. 1.** (a) Size distribution of the different aggregate types 1, ..., 5. (b) The distribution of water content in aggregates obtained from 23281 samples of different types used in concrete production

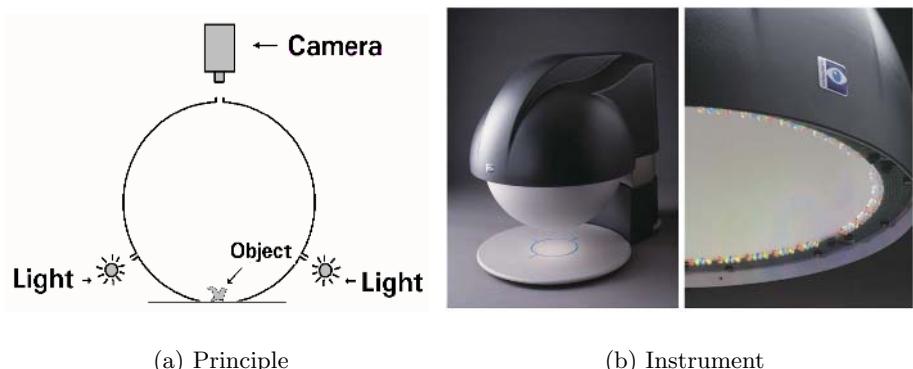
material, hence 1, 3 and 5 should be regarded as representative for 2 and 4. Figure 1(a) illustrate the size distribution, "Normal", represented for all types.

Finally, for each combination of type and size distribution, samples with a desired water content was prepared. Over the years, the concrete producer, 4K-Concrete, has manually monitored the amount of water present in the aggregates that they use. In order to have as realistic data material as possible, data about the measured water contents were used to identify the "real world" distribution of water in aggregate material (see Figure 1(b)). Based on this distribution, samples were prepared to have following water contents: 1.25%, 2.5%, 3.75%, 5.0%, 6.25%, 7.5% and 8.75% of the sample material weight. Establishment of these levels in the samples were based on the following steps: First all samples were dried out in an oven at 105°C, and finally, from the dried material samples were taken and water were added according the their weight, giving the correct water content.

Triplicates were produced of each combination of type, size distribution and water content, and put into sealed containers. From these containers material was filled into petri dishes and images were acquired (Section 2.2). Since the water content of the triplicates was approximated, the exact content was finally found by drying and weighing, after image acquisition. Finally, we had a total amount of samples of  $52 \times 3 = 156$  (including the replicates).

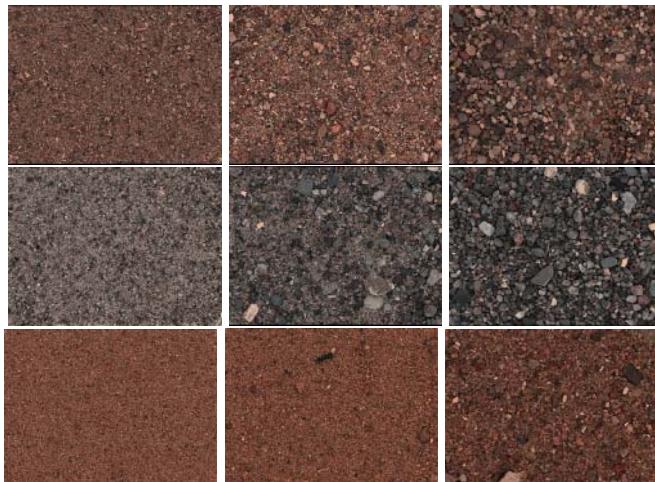
## 2.2 Image Acquisition

After preparation, the samples were put into petri dishes and digitized using the VideometerLab (illustrated in Figure 2). VideometerLab is a vision-based system for color and texture measurements. The camera is looking through an integrating (Ulbricht) sphere and the petri dishes were placed in an opening on



**Fig. 2.** VideometerLab is one implementation of the multispectral vision technology. (a) Illustration of the principle of imaging with an integrating sphere creating uniform illumination. Illumination of the object will come from reflections from the coating on the inner surface of the sphere. (b) The different diodes in the instrument

the opposite side of the sphere (Figure 2(a)). Using this system the dishes receives a uniform and diffuse light, and shading effects, shadows, and gloss-related effects are minimized. Furthermore the geometry of the illumination system is relatively simple to apply in an optical model. This means that the errors that are inherent in the system can be estimated and corrected for.



**Fig. 3.** Examples raw image data after acquisition. Illustration of three types of aggregates with different size distributions. The rows are ordered according to type: 1, 3 and 5. The columns are ordered according to size distribution: "Fine", "Normal" and "Coarse". All images have a water content of 2.5%

The multispectral measurements are obtained by strobing light emitting diodes (LED) with different spectral characteristics, and the reflectance from the surface of a petri dish is detected by a mega-pixel black-and-white camera producing a high resolution multispectral image.

Using this technology (Figure 2(b)), multispectral reflectance images were captured of all samples, each of which contained 9 frames (1035rows $\times$ 1380columns $\times$ 9bands $\times$ 32bit/pixel): 472nm (blue), 515nm (green), 592nm (amber), 630nm (red), 875nm (nir[low]), 428nm (ultra blue), 503nm (cyan), 612nm (orange), 940nm, (nir[high]). In Figure 3 six images captured can be seen. The images illustrate some of the diversity in type and size distribution.

### 3 Methods

During this study, we found that it is necessary to generate models for each of the aggregate types. There were an "overlap" between some of the aggregates in such a way, that two aggregates with different size distributions, could have the approximately same appearance at two different water contents. In order to overcome this, it was decided to have type specific models which led to a significant improvement of the methods. The reflectance bands themselves did not contain all the information, but a combination between them was more describing. The original bands were combined into new ones. The new "bands" were based on the ....., ....., and the ..... between the original bands. Simple global features were extracted from all bands: 5%, 10%, 20%, ..., 90%, 95% quantiles. Other and more complex measures were evaluated, but it was found that although a slight improvement was obtained, the gain was insignificant compared to processing time and complexity.

#### 3.1 Identification of Aggregate Type

Although information about type is present, when the concrete recipes is "chosen" in the factory, we sought for a method to validate and control that the information given is correct. When the features has been extracted and a large number of variables are employed, the risk of obtaining a poor classification increases due to the increased likelihood of noisy variables. A pre-selection methodology was implemented prior to classification, to screen out noisy and non-discriminating features.

Consider the  $k = 1, \dots, 5$  types with  $n_1, \dots, n_k$  multivariate  $p$ -dimensional observations  $\{\mathbf{X}_{ij}\}$ , where  $i$  is the group index and the  $j$  is the observation number. The group means are denoted  $\{\bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_k\}$  and the overall mean is denoted  $\bar{\mathbf{X}}$ , i.e.

$$\bar{\mathbf{X}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \bar{\mathbf{X}}_{ij}, \quad i = 1, \dots, k, \quad (1)$$

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{n_i} \bar{\mathbf{X}}_{ij}, \quad i = 1, \dots, k \quad \text{where} \quad N = \sum_{i=1}^k n_i. \quad (2)$$

As in a one way analysis of variance (ANOVA) the sum of squares matrix,  $\Sigma_T$ , the between-group matrix,  $\Sigma_B$ , and the within-group matrix,  $\Sigma_W$ , are defined by

$$\Sigma_B = \sum_{i=1}^k n_i (\bar{\mathbf{X}}_i - \bar{\mathbf{X}}) (\bar{\mathbf{X}}_i - \bar{\mathbf{X}})^T, \quad (3)$$

$$\Sigma_W = \sum_{i=1}^k \sum_{j=1}^{n_i} (\bar{\mathbf{X}}_{ij} - \bar{\mathbf{X}}_i) (\bar{\mathbf{X}}_{ij} - \bar{\mathbf{X}}_i)^T, \quad (4)$$

$$\Sigma_T = \Sigma_B + \Sigma_W = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{n_i} (\bar{\mathbf{X}}_{ij} - \bar{\mathbf{X}}) (\bar{\mathbf{X}}_{ij} - \bar{\mathbf{X}})^T. \quad (5)$$

Wilks' lambda can be defined as the ratio of the determinant of the between-group variance,  $\Sigma_W$ , to the determinant of the total variance,  $\Sigma_T$  for each feature to obtain an initial set of discriminating features [5]

$$\Lambda = \frac{|\Sigma_W|}{|\Sigma_T|}. \quad (6)$$

Wilks' lambda,  $\Lambda$ , can be transformed into an  $F$ -distribution,

$$V_j = - \left[ (n - 1) - \frac{p + 2}{2} \right] \ln(\Lambda), \quad (7)$$

in which  $n$  is the number of samples, and  $p$  the number of predictor variables (in  $\mathbf{X}_{ij}$ ). This allows the selection of discriminatory features with an appropriate confidence level [4, 6]. The test is analogous to the  $F$ -test used to test the significance of a regression model. The statistic  $V_j$  approximately follows a  $\chi^2$ -distribution with  $p$  degrees of freedom. In order to further reduce the amount of features and also remove redundant information, principal component analysis (PCA) [7] was applied and the principal components (PC's) explaining 95-99% of the variation was used in the further analysis.

After the elimination/reduction of features, that were found to be insignificant on a 5% significance level, canonical discriminant analysis (CDA) [7] of the remaining features were used to obtain the proper projection of data based on "optimal separation". We are looking for projections that maximize the ratio of the variation between groups and the variation within groups. The idea of maximizing this ratio was proposed by Fisher in 1936 [3], and this ratio equals the Rayleigh coefficient (or Fisher ratio)

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \Sigma_B \mathbf{w}}{\mathbf{w}^T \Sigma_W \mathbf{w}}, \quad (8)$$

i.e. the transformation is defined by the eigenvectors  $\mathbf{w}$  of  $\Sigma_B$  with respect to  $\Sigma_W$  and is found by  $\Sigma_B \mathbf{w} = \lambda \Sigma_W \mathbf{w}$ . The new variates are found by  $\mathbf{Y} = \mathbf{w}^T (\mathbf{X}_{ij} - \bar{\mathbf{X}})$ .

### 3.2 Estimation of Water Content

The explaining variates with respect to water content may differ from those for type. Therefore a new feature elimination procedure was applied. The dependent variables are now on a continuous scale, and a slightly different approach than the one in Section 3.1 has to be followed. Based on regression analysis [7] only features were selected, that could be proven to contain information. Redundant information was removed by applying PCA and the PC's retaining 95% of the variance was kept for further analysis. The PC's were then used as input to a final regression analysis, modelling the water content of the samples, purely based on image information.

### 3.3 Estimation of Aggregate Size Distribution

Let the values in a multispectral be described by  $I_n(x, y)$ ,  $n = 1, \dots, N$ , where  $n = 1, \dots, 9$  is a specific band, and  $I(x, y)$  the corresponding pixel (reflectance) at the coordinate  $(x, y)$ . Since only the structural information is of interest, band information is reduced through an average

$$\tilde{I}(x, y) = \sum_{n=1}^N \omega_n I_n(x, y) \quad (9)$$

where  $\omega_n$  is a weight put on each of the bands. These weights can be found by by PCA or a similar method, but in this study all weights were  $\omega_n = \frac{1}{N}$ . Finally, we center and standardize  $\tilde{I}(x, y)$  to have  $E[\tilde{I}(x, y)] = 0$  and  $V[\tilde{I}(x, y)] = 1$ .

Next, we apply a scale-space approach and convolve with a Gaussian kernel where  $\sigma$  is specifying the scale. The result can be seen as low-pass filtering that removes finer details as the scale,  $\sigma$ , increases. The filtered image is obtained by convolution

$$h(r; \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r^2}{2\sigma^2}}, \quad \tilde{I}_\sigma(x, y) = (\tilde{I}_\sigma(x, y) * h_{\hat{x}}) * h_{\hat{y}} \quad (10)$$

where  $h_{\hat{x}}$  and  $h_{\hat{y}}$  are the horizontal and vertical 1D separable kernels of the Gaussian filter. Features were extracted from  $\tilde{I}_\sigma(x, y)$ . The features were simple statistics: mean, standard deviation, skewness and kurtosis. In order to evaluate if the statistics were due to edges from many smaller objects or from one large or several larger objects in  $\tilde{I}_\sigma(x, y)$ , the mean and standard deviation of the gradient

$$G(x, y) = |\nabla \tilde{I}_\sigma(x, y)| = \left| \left( \frac{\delta}{\delta x} \hat{x} + \frac{\delta}{\delta y} \hat{y} \right) \tilde{I}_\sigma(x, y) \right| \quad (11)$$

did show to give significant improvement to the method.

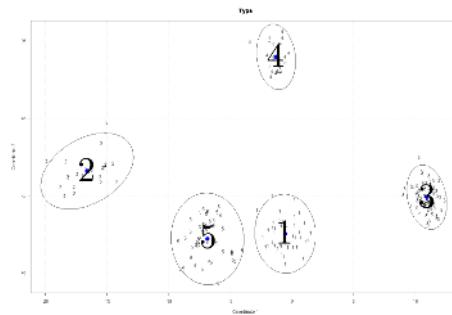
The strategy we have chosen to follow from here, is to predict the amount of aggregate material remaining in each of the sieves (for each types). Again, we end up with many features, and we apply the same strategy as when predicting the water contents. Having removed the insignificant variates, the PCA was used to reduce the number of features and remove redundancy. And the PC's retaining 99.9% of the variance were used in the final regression analysis.

## 4 Results

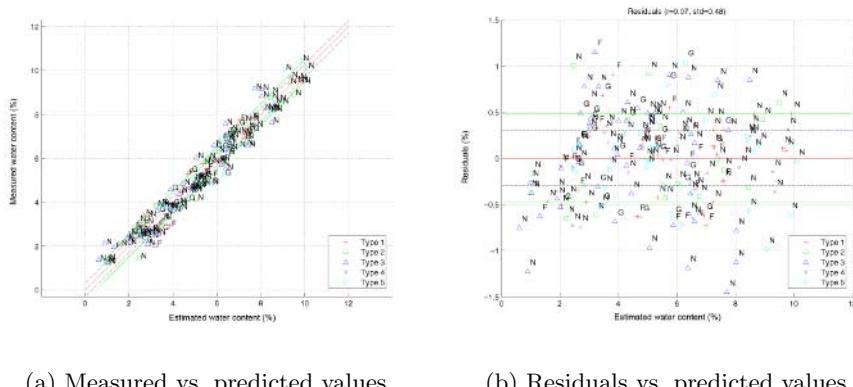
First step was to find a model for aggregate type. After the type was identified, aggregate type specific models were applied in order to predict the water content, and size distribution. The following results are based on a full-cross validation (leave-one out), which means that we took one (or more) samples out before training the models. Finally the sample(s) were identified, and the critical parameters were estimated.

Figure 4 show the cross-validation results of the type classification. As can be seen, all types are identified with no errors.

After identifying the aggregate type, the water content was predicted. Figure 5 shows the relations between the measured water content (Figure 5(a)) and the



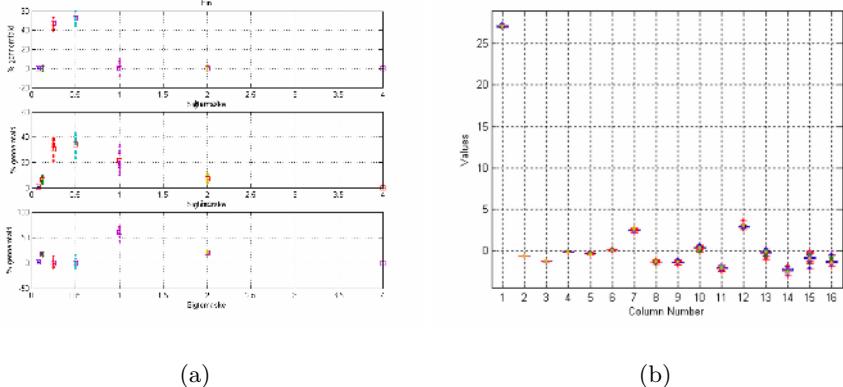
**Fig. 4.** The first and second canonical discriminant function of the aggregate types. The results are obtained by a leave one out cross-validation. The figure shows, that there is a clear and consistent separation between the five types



(a) Measured vs. predicted values.

(b) Residuals vs. predicted values.

**Fig. 5.** Results after estimation of water content. The results are obtained by leave one out cross-validation



**Fig. 6.** Results after size distribution estimation for aggregate type 1. (a) Showing the size distribution values (predicted: ●, laboratory value: □) of sieved aggregate material (relative to the total amount in all sieves). (b) The regression weights. All results are obtained by leave one out cross-validation

residuals (Figure 5(a)) as function of the predicted values. From the figures it can be seen, that there is a clear relation between the estimated and the predicted water contents. The standard deviation of the residuals is  $\sigma = 0.48\%$ .

Finally, the size distributions were estimated. Figure 6 show the result for the one of the types (type 1). Figure 6(a) shows the predicted values (●) together with the in laboratory measured value (□). From the figures we conclude, that there is a large agreement between the predicted and measured values. Figure 6(b) plots the model weights applied on to the 15 PC's that are input to the regression model. Whereas some of the PC's could be regarded as insignificant for some of the aggregate types, they were proven to be significant for other types.

The standard deviations for the residuals for all sieves and all types were lying in the range of  $\sigma \in \{0.04\%; 7.24\%\}$  when evaluating the triplicates as separate observations. When pooling the triplicates into one observation the standard deviations fell to be in the range of  $\sigma \in \{0.03\%; 5.2\%\}$

## 5 Discussion

In this study, we have shown that it is possible to estimate the critical parameters related to aggregates in concrete production using multispectral vision technology. Reflectance images are captured, and simple low-level features calculated and used as parameters in models describing both water content and size distribution. Although prior knowledge about aggregate type was proven to be necessary in order to obtain satisfying results, the types could be perfectly identified. Based on type specific models, estimates of water content and size distribution were obtained showing high proficiency.

Although the performance of the methods was high, many question still has to be answered, i.e. what are the significant features and how can they be interpreted? Also, the bands when estimating the size distributions are pooled. It might be, that there are more information hidden within each of the single bands.

The methods suggested are well suited to implemented in an industrial application. The features are simple, robust and easy to be calculate. Further studies have been planned, in order to reveal further knowledge about these and other questions.

## Acknowledgments

This work was supported by The Danish Ministry of Science, Technology and Innovation. We would like to thank Dorthe Mathiesen (Danish Technological Institute) and Freddie Scheye (4K) for constructive discussions and Finn Østergaard for skillful technical assistance.

## References

1. S. Mindess and J.F. Young, Concrete. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.
2. M. R. Rixom and N. P. Mailuaganam, Chemical Admixtures for Concrete. R. & F.N. Spon, NY, 1986.
3. R.A. Fisher (1936), "The use of multiple measurements in taxonomic problems", *Annals of Eugenics* 7, 179–188.
4. M. S. Bartlett, "The Statistical Significance of Canonical Correlations", *Biometrika*, Vol. 32, No. 1. (Jan., 1941), pp. 29–37
5. S. S. Wilks, "Certain Generalizations in the Analysis of Variance", *Biometrika*, Vol. 24, No. 3/4. (Nov., 1932), pp. 471–494.
6. D. N. Lawley, "Tests of Significance in Canonical Analysis", *Biometrika*, Vol. 46, No. 1/2. (Jun., 1959), pp. 59–66.
7. J. Lattin, J. D. Carroll and P. E. Green, "Analyzing Multivariate Data", Brooks/Cole, 2003.

# Automated Multiple View Inspection Based on Uncalibrated Image Sequences

Domingo Mery and Miguel Carrasco

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile  
Av. Vicuña Mackenna 4860(143), Santiago de Chile  
[dmery@ing.puc.cl](mailto:dmery@ing.puc.cl)

**Abstract.** The Automated Multiple View Inspection (AMVI) has been recently developed for automated defect detection of manufactured objects. The approach detects defects by analysing image sequences in two steps. In the first step, potential defects are automatically identified in each image of the sequence. In the second step, the potential defects are tracked in the sequence. The key idea of this strategy is that only the existing defects (and not the false detections) can be successfully tracked in the image sequence because they are located in positions dictated by the motion of the test object. The AMVI strategy was successfully implemented for calibrated image sequences. However, it is not simple to implement it in industrial environments because the calibration process is a difficult task and unstable. In order to avoid the mentioned disadvantages, in this paper we propose a new AMVI strategy based on the tracking of potential defects in uncalibrated image sequences. Our approach tracks the potential defects based on a motion model estimated from the image sequence itself. Thus, we obtain a motion model by matching structure points of the images. We show in our experimental results on aluminium die castings that the detection is promising in uncalibrated images by detecting 92.3% of all existing defects with only 0.33 false alarms per image.

**Keywords:** defect detection, automated visual inspection, multiple view geometry.

## 1 Introduction

Recently, a new methodology, called the Automated Multiple View Inspection (AMVI), has been developed for automated defect detection [1]. In contrast to the classic inspection methods that analyse individual images, AMVI detects defects by analysing image sequences. Thus, AMVI is similar to the way a (human) inspector examines a test object: first, the inspector detects anomalous details in an image sequence obtained from the test object in motion; and second, the inspector tracks in the image sequence the irregularities detected in the first step. If the inspector can track them, i.e., if the irregularities are visible

among the image sequence, he or she classifies the test object as defectively. Similarly, the suggested computer-aided method AMVI is able to detect defects in two steps. In the first step, called . . . . ., potential defects are automatically identified in each image of the sequence using a single filter and no a priori knowledge of the structure of the test object. In the second step, called . . . . ., an attempt is made to track the identified potential defects in the image sequence. Therefore, only the existing defects (and not the false detections) can be successfully tracked in the image sequence because they are located in positions dictated by the motion of the test object. Thus, two or more views of the same object taken from different viewpoints can be used to confirm and improve the diagnostic done by analysing only one image. A similar idea is also used by radiologists that analyse two different view X-rays of the same breast to detect cancer in its early stages. Hence, the number of cancers flagged erroneously and missed cancers may be greatly reduced (see for example [2], where a novel method that finds automatically correspondences in two different views of the breast is presented).

The exploitation of the multiple view analysis represents a new methodology in the automated visual inspection. Indeed, this multiple view strategy is opening up new possibilities in this field by taking into account the useful information about the correspondence between the different views of the test object.

The AMVI strategy was implemented in [1] for automated inspection of aluminium die castings using calibrated radioscopic image sequences. In this case, the projection model  $3D \rightarrow 2D$  is estimated off-line using a calibration object, and the potential defects are tracked on-line according to the principles of multiple view geometry [3], where the geometric constraints between different views of the sequence can be easily established using multi-focal tensors. The preliminary results obtained using AMVI methodology are promising in . . . . image sequences achieving impressive discrimination of false alarms while detecting the real defects of the object test. However, it is not simple to implement it in industrial environments because the calibration process is a difficult task and the vibrations of the imaging system induce inaccuracies in the estimated parameters of the multiple view geometric model. Thus, the calibration is not stable and the imaging system must be calibrated periodically.

In order to avoid the mentioned disadvantages, in this paper we propose a new AMVI strategy based on the tracking of potential defects in . . . . image sequences. Hence, it is not necessary to calibrate the imaging system. The reason why uncalibrated images can be used is because there are several constraints characteristic of the inspection problem that simplify the tracking task. Our approach tracks the potential defects based on a motion model estimated from the image sequence self. By means of this motion model, the potential defects can be successfully tracked in the uncalibrated image sequence.

In this paper we present the mentioned new methodology based on uncalibrated image sequences. The rest of the paper is organised as follows: Section 2 explains our approach for uncalibrated AMVI. Section 3 shows preliminary

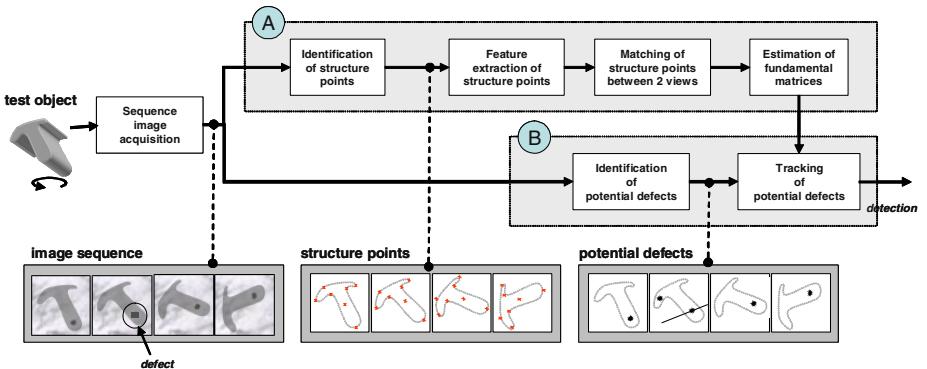
results obtained with the proposed methodology. Finally, Section 4 gives concluding remarks and perspectives for future works.

## 2 Uncalibrated AMVI

The main objective of our research is to perform a robust automated visual inspection based on the tracking of potential defects without calibrating the imaging system. The reason why uncalibrated images can be used is because there are several constraints characteristic of our inspection problem that can be considered to perform the tracking in an uncalibrated way: e.g., the scene captured by an image of the sequence consists of only one rigid object in motion, there is no significant frame to frame motion and the 2D trajectories are smooth, the velocity of the test object is constant, and generally the motion of the test object is only rotational or translational. Since the aim of AMVI is to track the potential defects only (and not to estimate the structure of the test object), we propose to track the potential defects without computing the 3D → 2D model.

Our approach estimates a motion model using a simple methodology: i) it identifies structure points in each image of the sequence; ii) it finds corresponding structure points between consecutive frames of the sequence; and iii) it estimates the fundamental matrices from the corresponding points using a robust algorithm. By means of this motion model, the potential defects can be successfully tracked in the uncalibrated image sequence. Thus, the key idea of the uncalibrated AMVI is that the information obtained from the tracking of structure points can be used to track the potential defects in the sequence.

The proposed method for uncalibrated AMVI can be structured in two general steps (see Fig. 1): A) motion estimation and B) defects detection. The second step is similar to the calibrated AMVI method explained in previous section. In the following, both steps are described in further detail.



**Fig. 1.** Block diagram of the uncalibrated automated multiple view inspection: A) estimation of motion model, B) detection of defects

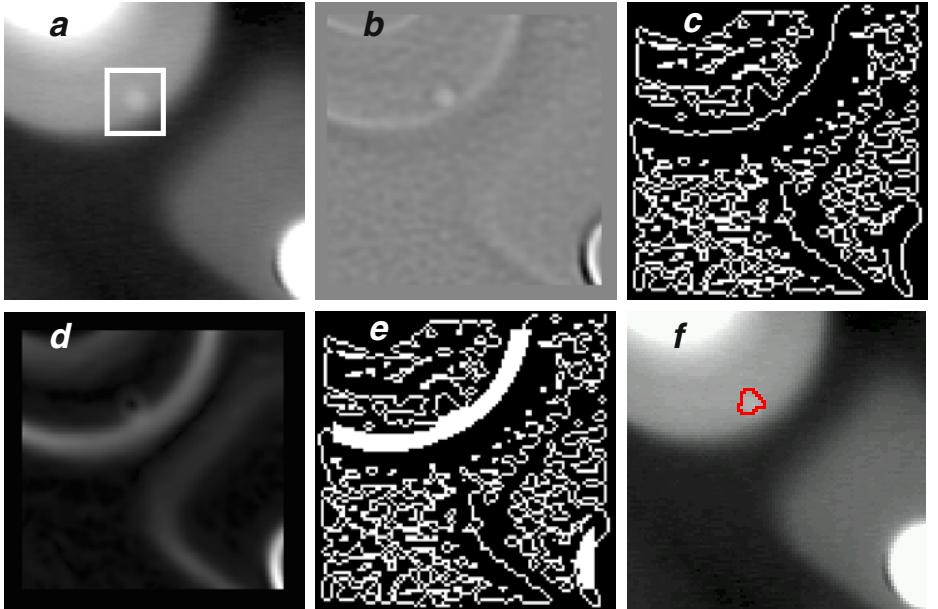
**A.1) Identification of structure points:** The whole test object is segmented using the Otsu's approach [4], where the foreground is separated from the background using a global threshold. In this case, it is assumed that the intensity of the object is clearly differentiated from the background, i.e., the image presents a bimodal histogram. The segmented regions of the object are labelled. For each labelled region the centre of mass and the external points (top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom and left-top) are defined as structure points of the object. Incomplete regions are not considered.

**A.2) Feature extraction of structure points:** In order to match the identified structure points, certain features must be measured. In our approach, the coordinates, and the mean grey level of the neighbourhood are extracted for each identified structure point.

**A.3) Matching of structure points:** Since the motion of the test object is slow, i.e., the trajectories in the sequence are smooth, the structure points in two consecutive frames can be easily matched by considering the nearness of the structure points and the similarity in the grey level.

**A.4) Estimation of fundamental matrices:** Structure points can be erroneously matched in previous step. In addition, for rotated circular regions the external points in different views are not necessarily corresponding points. For this reason, a robust algorithm that estimates the fundamental matrix is required. We use the RANSAC approach [3] that estimates the fundamental matrix without considering outliers (mismatches).

**B.1) Identification of potential defects:** The segmentation of potential defects identifies regions in each image of the sequence that may correspond to real defects. Two general characteristics of the defects are used to identify them: a) a defect can be considered as a connected subset of the image, and b) the grey level difference between a defect and its neighbourhood is significant. The potential defects are identified without a-priori knowledge. First, a Laplacian-of-Gaussian (LoG) kernel and a zero crossing algorithm [5] are used to detect the edges of the X-ray images. The LoG-operator involves a Gaussian lowpass filter which is a good choice for the pre-smoothing of our noisy images that are obtained without frame averaging. The resulting binary edge image should produce at real defects closed and connected contours which demarcate . However, a defect may not be perfectly enclosed if it is located at an edge of a regular structure as shown in Fig. 2c. In order to complete the remaining edges of these defects, a thickening of the edges of the regular structure is performed as follows: a) the gradient of the original image is calculated (see Fig. 2d); b) by thresholding the gradient image at a high grey level a new binary image is obtained; and c) the resulting image is added to the zero crossing image (see Fig. 2e). Afterwards, each closed region is segmented. In order to identify the potential defects, features are



**Fig. 2.** Detection of flaws: a) radioscopic image with a small flaw at an edge of a regular structure, b) Laplacian-filtered image with  $\sigma = 1.25$  pixels (kernel size =  $11 \times 11$ ), c) zero crossing image, d) gradient image, e) edge detection after adding high gradient pixels, and f) detected flaw using the variance of the crossing line profile

extracted from crossing line profiles of each segmented region [6]. Crossing line profiles are grey level profiles along straight lines crossing each segmented region in the middle. If the variance of the crossing line profiles is high, the segmented region is classified as potential defect. This is a very simple detector of potential defects with a large number of false alarms flagged erroneously. However, the advantages are as follows: a) it is a single detector (it is the same detector for each image), b) it is able to identify potential defects independent of the placement and the structure of the test object, i.e., without a-priori information of the design structure of the test object, and c) the detection rate of real defects is very high (more than 90%).

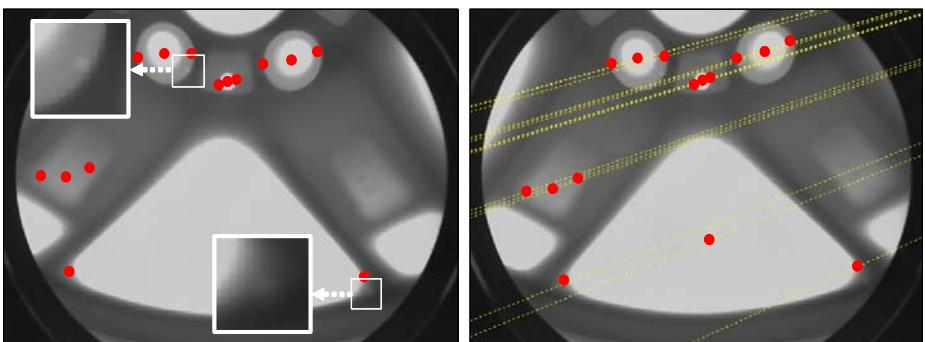
**B.2) Tracking of potential defects:** Tracking requires the position of the centre of mass of each detected potential defect. In this work,  $a = (a, p)$  will denote the identified potential defect  $a$  in image  $p$ . It is assumed that the image sequence has  $N$  images ( $1 \leq p \leq N$ ) and  $n_p$  points were identified in image  $p$  ( $1 \leq a \leq n_p$ ). The position of potential defect  $a = (a, p)$  is arranged in a vector  $\mathbf{m}_p^a$ . One obtains then the position vector  $\mathbf{m}_p^a = (x_p^a, y_p^a)$ . This step matches two potential defects (of two views), potential defect  $a = (a, p)$  with potential defect  $b = (b, q)$ , for  $p \neq q$ , if  $\mathbf{m}_p^a$  and  $\mathbf{m}_q^b$  satisfy the epipolar constraint. In addition, a similarity condition is used, i.e., the matching is established if the potential defects are similar enough. To evaluate this criterion a

is calculated as the Euclidean distance between the normalised feature vectors of the potential defects. In this case, the features area and grey level value of the potential defects are used to establish the similarity. If the false alarms cannot be eliminated with the tracking in two views, a tracking in three and four views using trifocal tensors should be used (see for example [7]). Trifocal tensors can be estimated from the fundamental matrices [3].

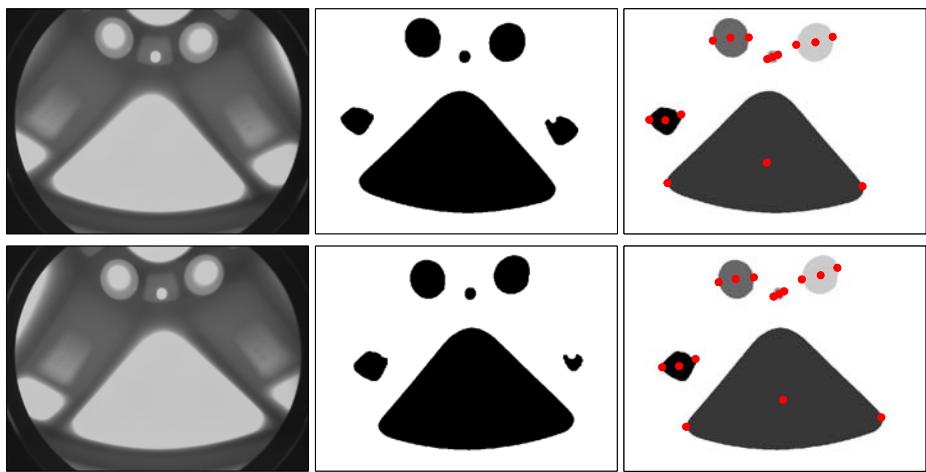
### 3 Experimental Results

In this section, we present results obtained from several radioscopic images of aluminium die castings using the proposed approach. For example, two different radioscopic images of an aluminium wheel, used in our experiments, are shown in Fig. 3. These images will be used to illustrate the steps of our method outlined in Section 2. The algorithm starts with the identification of structure points in each view (see Fig. 4). Afterwards, the position of each point and its mean grey level of a  $3 \times 3$  centred mask are measured. These values are used to match the structure points in both views. Using the matched points, the fundamental matrix of these two views is estimated with a RANSAC algorithm. Fourteen obtained epipolar lines are drawn in Fig. 3. We observe that the epipolar lines lie on the marked points successfully.

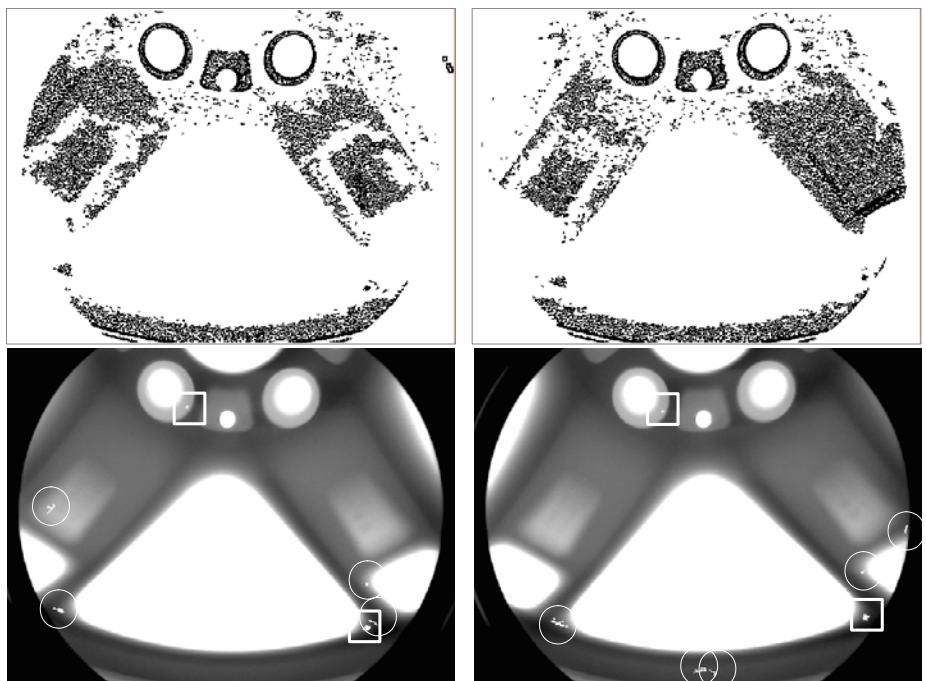
After the estimation of the fundamental matrix, we identify the potential defects in each image. The results are shown in Fig. 5. We observe that left image has six potential defects (only two of them are real defects). On the other hand, right image has seven potential defects (only two of them are real defects). In both images, the existing defects are identified. However, there are nine false alarms (four in left image and five in right image) that do not correspond to real defects. The false alarms should be eliminated by the tracking step.



**Fig. 3.** Radioscopic images of a wheel using in our experiments. In each view, there are two defects (see white squares in left view). In addition, the obtained epipolar lines are drawn for fourteen points



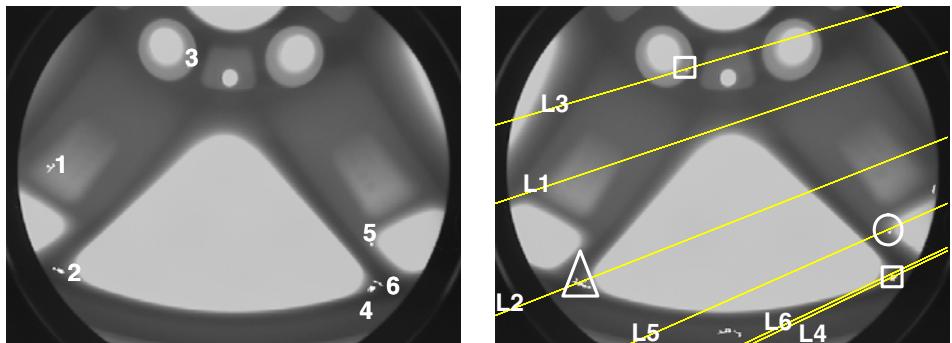
**Fig. 4.** Finding structure points in two views: left) original images, middle) segmentation after Otsu's method, right) extracted structure points



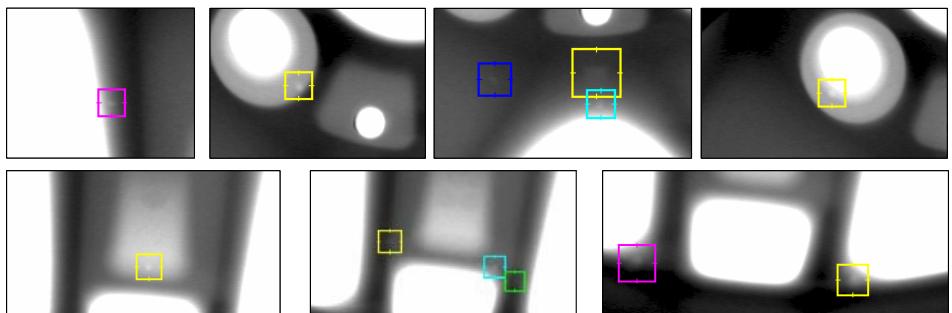
**Fig. 5.** Identification of potential defects in both views: top) segmented regions, bottom) potential defects after crossing line profile approach (squares: real defects, circles: false alarms)

The results of the tracking step are shown in 6. In this step, we try to find in right image the corresponding potential defects of the six potential defects of left image. In the AMVI strategy, the search for the corresponding potential defect in both images is restricted to the epipolar lines and to similar potential defects. The similarity of the candidates is evaluated by comparing their areas and mean grey values. In this approach, we eliminate those potential defects that cannot be matched. In this example, the existing defects are detected flagging only one false alarm.

In order to verify the performance of our method, a broader set of images were analysed. In our experiments, radioscopic image sequences (without frame



**Fig. 6.** Matching of potential defects in both images. The six epipolar lines of the potential defects of left image are denoted as L1, ... L6 in right image. Potential defects 1 and 6 cannot be matched (there is no potential defect on the epipolar lines), i.e., potential defects 1 and 6 are eliminated. The epipolar line L2 meets a potential defect in right image (see triangle), however both potential defects are not similar enough, i.e., potential defect 2 is eliminated. Potential defects 3, 4 and 5 fulfil epipolar and similarity conditions, i.e., they are detected as defects, however potential defect 5 is a false alarm. In this case, the existing defects are detected successfully (see squares) with one false alarm (see circle)



**Fig. 7.** Details of the radioscopic images used in our experiments. The squares in all images of this section were drawn intentionally to show the reader where the defects are located

averaging) of aluminium wheels with twelve known flaws were inspected. Three of these defects were existing blow holes ( $\emptyset = 1.5 \sim 7.5$  mm). They were initially detected by a visual (human) inspection. The remaining nine flaws were produced by drilling small holes ( $\emptyset = 2.0 \sim 4.0$  mm) in positions of the casting which were known to be difficult to detect. Some details of the radioscopic images showing the defects are presented in Fig. 7.

We test this approach in twelve stereo images, i.e., in 24 radioscopic images of  $572 \times 768$  pixels. Since there are twelve defects viewed in different images of the sequences, the total number of the imaged flaws was 39 in the 24 images. In this experiment, the identification of potential defects was perfect, i.e., all existing defects were successfully segmented. However, there were many false alarms. The results of the identification are summarised in Table 1. On average, there were 3.25 defects and 3.46 false alarms per image, i.e., 51.5% of the identified potential defects were false alarms.

After the tracking step, 92.3% of the potential defects were successfully detected with only 0.33 false alarms per image. Table 2 compares the results obtained in the identification and tracking steps. We observe that the tracking step is able to filter out 90.0% (from 41.5 to 4) of the false alarms eliminating only 7.7% (from 39 to 36) of the existing defects.

**Table 1.** Performance of the identification of potential defects

Images	Number of images	Detected defects/image	False alarms/image
Left images	12	39/12 = 3.25	40/12 = 3.33
Right images	12	39/12 = 3.25	43/12 = 3.58
All images	24	78/24 = 3.25	83/24 = 3.46

**Table 2.** Performance of the tracking of potential defects

Step	Detected defects	No detected defects	False alarms	Detection performance	False alarm rate	False alarms per image
Identification	39	0	41.5	100%	51.5%	3.46
Tracking	36	3	4	92.3%	10.0%	0.33

## 4 Concluding Remarks

In this paper we presented a new automated multiple view inspection strategy based on the tracking of potential defects in uncalibrated image sequences. Our approach tracks the potential defects based on a motion model estimated from the image sequence self. Since no calibration is required, we believe that the implementation of the automated multiple view inspection will be now possible in industrial environments.

We have shown in our experimental results on aluminium die castings that the detection is promising in uncalibrated images by detecting 92.3% of all existing defects with only 0.33 false alarms per image. However, since the performance

of the method has been verified on only 24 images, an evaluation on a broader data base is necessary.

In our experiments, we use a tracking based on only two views because the quote of false alarms is low. Nevertheless, as future work the tracking step will be performed using three and four views in order to increase the performance of the algorithm.

## Acknowledgment

This work was supported by FONDECYT – Chile under grant no. 1040210.

## References

1. Mery, D., Filbert, D.: Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. *IEEE Trans. Robotics and Automation* **18** (2002) 890–901
2. Kita, Y., Highnam, R., Brady, M.: Correspondence between different view breast X-rays using curved epipolar lines. *Computer, Vision and Understanding* **83** (2001) 38–56
3. Hartley, R.I., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2000)
4. Haralick, R., Shapiro, L.: Computer and robot vision. Addison-Wesley Publishing Co., New York (1992)
5. Castleman, K.: Digital image processing. Prentice-Hall, Englewood Cliffs, New Jersey (1996)
6. Mery, D.: Crossing line profile: a new approach to detecting defects in aluminium castings. *Lecture Notes in Computer Science* **2749** (2003) 725–732
7. Mery, D., Ochoa, F., Vidal, R.: Tracking of points in a calibrated and noisy image sequence. *Lecture Notes in Computer Science* **3211** (2004) 647–654

# Interactive 3-D Modeling System Using a Hand-Held Video Camera

Kenji Fudono<sup>1</sup>, Tomokazu Sato<sup>2</sup>, and Naokazu Yokoya<sup>2</sup>

<sup>1</sup> Victor Company of Japan

<sup>2</sup> Nara Institute of Science and Technology, Japan

**Abstract.** Recently, a number of methods for 3-D modeling from images have been developed. However, the accuracy of a reconstructed model depends on camera positions and postures with which the images are obtained. In most of conventional methods, some skills for adequately controlling the camera movement are needed for users to obtain a good 3-D model. In this study, we propose an interactive 3-D modeling interface in which special skills are not required. This interface consists of “indication of camera movement” and “preview of reconstruction result.” In experiments for subjective evaluation, we verify the usefulness of the proposed 3D modeling interfaces.

## 1 Introduction

In recent years, 3-D models of real objects have been often used for several purposes such as entertainment, education, and design. Generally, these 3-D models are constructed by experts who have special skills and devices for 3-D modeling. On the other hand, high-quality 3-D graphics have become very familiar to general people because 3-D graphics are available even on a cellular telephone today. Such a situation gives an increased demand to import 3-D models of real objects to personal web pages, games, and so on. For this purpose, simple 3-D modeling methods for real objects are necessary for users who have no special skills and devices to model the 3-D objects. To reconstruct 3-D models of real objects, several methods have been developed in the literature; methods using a video camera[1, 2], methods using a laser rangefinder[3], and methods using structured lights[4]. However, the accuracy of reconstructed 3-D models depends on the way of measurement. Thus, measurement skill is necessary to obtain good 3-D models.

To remove the difficulty in 3-D measurement, several support systems to obtain good 3-D models have been investigated[5, 6, 7]. These systems indicate how to move a measuring device based on a result of a reconstructed model. The indications allow users who have no special skills of modeling to get good 3-D models in a short time. However, it is difficult for personal users to use such systems, because these systems are designed for special and expensive devices such as a laser rangefinder. Although 3-D modeling systems that use only cheap devices have been developed[1, 2, 8, 9, 10], such systems do not indicate how to

move cameras. There is also a problem that these systems take a long time to reconstruct the models due to expensive computational cost. It is difficult for users to efficiently learn how to move the camera.

In this study, we propose an interactive 3-D modeling system by which users can obtain the model efficiently. The proposed system realizes two new functions: “indication of camera movement” and “real-time preview of reconstruction result”. Users without special skills can easily obtain 3-D models by following the indication from the system. Users can also get a good 3-D model in a short time owing to a real-time preview of reconstructing a model.

## 2 Interactive Modeling System

In this section, we first describe a design policy and outline of the proposed interactive modeling system. Each process of the interactive modeling system for personal users is then detailed.

### 2.1 Design Policy and System Outline

The purpose of the proposed modeling system is to allow personal users to get good 3-D models efficiently. To realize this purpose, the following three requirements should be satisfied:

- (a) realization of real-time modeling using cheap devices,
- (b) realization of real-time indication of camera movement,
- (c) realization of real-time preview of reconstruction results.

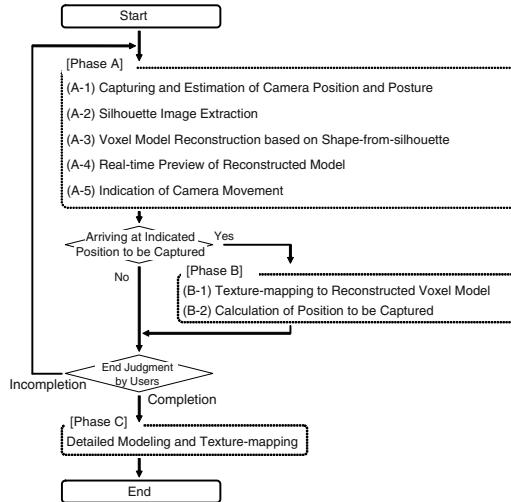
Our modeling system assumes that an object is located on a marker sheet and users move a hand-held video camera by following the indication from the system. To satisfy the requirements above, the system provides the following functions:

**(1) Real-time Modeling Using a Hand-held Video Camera:** While users capture objects by using a hand-held video camera, the system reconstructs 3-D models of the objects in real-time. This function satisfies the requirement (a).

**(2) Capturing Support Interface:** The system estimates the best view position from which images can be captured to acquire good 3-D models. The motion path from the current camera position to the best view position is shown to user on a computer display. User can easily obtain a 3-D model by following the indication of camera motion provided by the system. This function satisfies the requirement (b).

**(3) Preview of Reconstruction Model:** Preview of generating a 3-D model of the object is displayed and updated in every frame. User can preview reconstruction results without any waiting time. This function satisfies the requirement (c).

Figure 1 shows a flow diagram of the proposed system. The system consists of three phases: the phase A is a real-time process for 3-D modeling and preview, the phase B is an intermittent process for computationally expensive texture



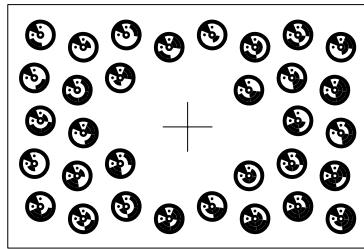
**Fig. 1.** Flow of interactive modeling system

generation and the best view decision, and the phase C is a refinement process to acquire a detailed modeling result.

The phase A reconstructs a 3-D shape of the object in real-time. First, the position and posture of the hand-held camera are estimated by recognizing markers (A-1). A silhouette image which discriminates object regions and background regions is generated (A-2). A voxel model is then reconstructed based on shape-from-silhouette (A-3). Preview of the reconstructed model is generated (A-4). Finally, the best view position is indicated (A-5). The phase B intermittently performs processes that are difficult to perform in real-time. First, texture-mapping to the reconstructed model is performed (B-1). The new best view position is then calculated (B-2). Above-mentioned phases A and B are repeated until users decide that further capturing is unnecessary. The phase C reconstructs a more detailed model than that reconstructed in the process (A-3) by using the whole captured image sequence. Note that the intrinsic parameters of the hand-held video camera are assumed to be known in this paper. To acquire a good silhouette image, it is also assumed that a marker sheet is located under a target object and wall and table have the same color.

## 2.2 Capturing and Estimation of Camera Position and Posture

In this process (A-1), the current position and posture of the hand-held video camera are estimated by using recognized markers in a captured image. In this section, first, markers are extracted from the input image that is captured by the hand-held video camera. Next, extracted markers are identified based on the patterns of the markers. Coordinates of markers on both world coordinate and image coordinate are recognized by identifiers of markers, and finally the position and posture of the camera are estimated from the coordinate values of the markers.



**Fig. 2.** Marker sheet

### Extraction and Recognition of Markers

Figure 2 shows the marker sheet. Circular markers printed in this sheet are those proposed by Naimark et al.[11]. Each marker has 6 bits identifier that makes it possible to discriminate one another. Moreover, one marker has 3 identifiable points. The system gets multiple identifiable points by extracting and recognizing the markers from the captured frame, and acquires the coordinates in image and world coordinate systems.

### Estimation of Camera Position and Posture

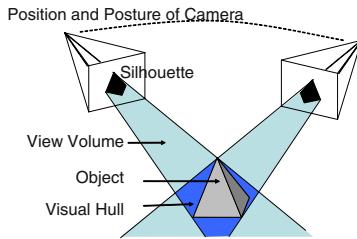
The camera position and posture are estimated by solving the Perspective n-Point (PnP) problem from the relation between image coordinates and world coordinates using a standard computer vision technique[12]. Three parameters ( $X, Y, Z$ ) as a camera position and three parameters (pitch, roll, and yaw angles) as a camera posture are actually calculated by solving the PnP problem.

### 2.3 Silhouette Image Extraction

In this process (A-2), a silhouette image is extracted from a captured frame by using the estimated position and posture information of the camera. The silhouette image is used as an input for the shape-from-silhouette process (A-3). In this section, first, colors of background wall and desk are detected from the input image by using the camera position and posture information. Background regions are then extracted based on the detected background colors, and a silhouette image is generated.

### Detection of Background Colors

Backgrounds consist of several regions; marker sheet, table, and wall behind the object. In this study, it is assumed that the base color of marker sheet, the region of table, and the region of wall surface have basically the same color. However, there may be a little difference in each color. To determine the background colors, firstly, the colors of the unprinted subregions around the extracted markers on the marker sheet are determined. Then the table regions are extracted based on the camera position and posture information, and the colors of the table regions are also detected.



**Fig. 3.** Silhouette constraint

### Extraction of Object Regions

An input image is divided into object and background regions by using the differences of the brightness and the chromaticness of background colors detected in the previous step. After the detection of background and object regions by paper and wall colors, a silhouette image is generated by merging extracted object regions.

### 2.4 Voxel Model Reconstruction Based on Shape-from-Silhouette

In this process (A-3), a 3-D voxel model is reconstructed based on shape-from-silhouette. In this section, first, the framework of shape-from-silhouette is briefly summarized. A method of voxel model reconstruction is then described.

#### Shape-from-silhouette

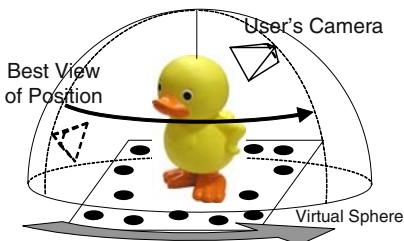
Shape-from-silhouette is a 3-D reconstruction method which is based on the silhouette constraint[13]. As shown in Figure 3, shape-from-silhouette approach reconstructs a 3-D model by assuming that “a target object is included in a view volume determined by the object’s silhouette from camera center of the projection to the space.” Intersections of view volumes generated from multiple camera positions are called visual hulls. A shape of visual hull is an approximated shape of the underlying object captured by multiple cameras.

#### Voxel Model Reconstruction

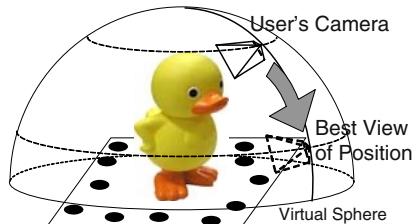
As one of the shape-from-silhouette methods, we employ a method that sets a cuboid in a voxel space which comprises the object preliminarily. The shape of the voxel model approximates a shape of object model by gradually carving the voxels of the cuboid which are outside of the view volume[14]. To reconstruct the voxel model efficiently, we use the parallel volume intersection method based on plane-to-plane projection proposed by Wu et al.[15].

### 2.5 Real-Time Preview of Reconstructed Model

In this process (A-4), the reconstructed voxel model is rendered and updated in every frame. The user can look at the 3-D model by using the mouse operation. The user can also confirm the progress of the reconstruction by this preview of the generated model.



**Fig. 4.** Rotation of Marker Sheet under Object



**Fig. 5.** Up-and-down Movement of Camera



**Fig. 6.** Rotation Arrows



**Fig. 7.** Arrows for Camera Movement

## 2.6 Indication of Camera Movement

In this process (A-5), how to move a camera is indicated for user by superposition to present the best view position from which the target object should be captured. The best view position is calculated in the intermittent process (Phase B). In this study, we prepare two types of indications: “(1) Indication of Rotation Movement” and “(2) Indication of Up-and-down Movement.” In our system, the best view camera position is expressed by longitude and latitude on a virtual sphere which is located at the center on the marker sheet. As shown in Figure 4, the longitude of the camera position and the best view position are matched by rotating the marker sheet under the target object. Subsequently, as shown in Figure 5, the latitude of the camera position and the best view position are matched by up-and-down camera movement. Indications (1) and (2) are not shown at the same time. The indication (1) is shown first. When the indication (1) is finished, the indication (2) is then shown. Each indication is explained below in some details.

### Indication of Rotation of Marker Sheet under Object

First, the system calculates the shortest rotation direction from the current camera position to the best view position. Then, as shown in Figure 6, the system shows rotation arrows by superposition on the sheet. The amount of rotation is shown on the arrows using color and the indicator at the bottom of

a screen. Indication (1) is finished when the longitude difference between the camera and the best view position becomes sufficiently small.

### **Indication of Up-and-down Movement of Camera**

As shown in Figure 7, the system shows arrows for a camera movement by superposition in a real scene. The amount of movement is shown using color and the indicator at the bottom of a screen. This indication is finished when the latitude difference between the camera and the best view position becomes sufficiently small. When a user completes to move the camera to the best view position by following indications, the system goes to the phase B.

## **2.7 Texture-Mapping to Reconstructed Voxel Model**

In this process (B-1), voxels are painted by projecting colors of an input image to a reconstructed model. The procedure of texture-mapping is detailed below.

### **Detection of Surficial Voxels**

Only surficial voxels of a reconstructed model should be painted. In this process, voxels that are surrounded by the other voxels are removed to detect surficial voxels  $V_i$  ( $i = 1, \dots$ , the number of surficial voxels).

### **Visibility Test**

In this section, surficial voxel  $V_i$  that is visible from each captured position  $C_j$  ( $j = 1, \dots$ , the number of captured frames) is detected. If there is no voxels between a surficial voxel  $V_i$  and a captured position  $C_j$ , a surficial voxel  $V_i$  is visible from  $C_j$ . Visibility tests for all surficial voxels are performed by all the captured frames.

### **Coloring Voxel**

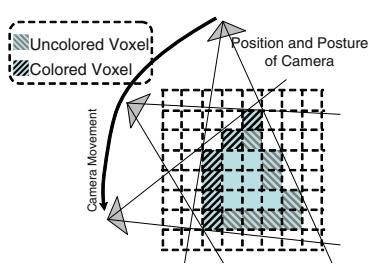
A surficial voxel  $V_i$  visible from a captured position  $C_j$  is projected to an image plane of a captured position  $C_j$ , and the color of the surficial voxel is set by the color of projected pixel on the image plane. If the surficial voxel is visible from multiple captured positions, the color of the surficial voxel is set to the average color of projected pixels on the image planes.

## **2.8 Calculation of Position to Be Captured**

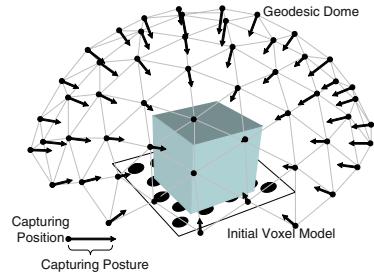
As shown in Figure 8, some uncolored surficial voxels exist because they are invisible from all the input images. In this process (B-2), the system computes the best view position from which the most uncolored voxels can be observed. Users can get a good 3-D model efficiently by following the indication from the system. First, to reduce the computational cost, candidates of positions to capture are enumerated. The best indication position to capture is then chosen from the candidates.

### **Candidates Enumeration**

To calculate the position from which the most number of uncolored voxels can be observed, it is necessary to count visible voxels from all the positions and



**Fig. 8.** Colored and Uncolored Voxels



**Fig. 9.** Part of Candidates of Best View

postures of the camera. This is computationally expensive. In our system, the candidates of the best view positions are enumerated. The candidates are vertices of a geodesic dome as shown in Figure 9. The geodesic dome is located on the center of the marker sheet. The radius of the dome is set so that the whole of initial voxel model can be captured by a camera. Each posture of candidates faces the center of the dome.

#### Determination of Best View Position

Uncolored surficial voxels are counted for all the candidate positions. By the result of uncolored voxel count, the system selects the best view position from which the most number of uncolored surficial voxels are visible.

#### 2.9 Detailed Modeling and Texture-Mapping

When users decide further capturing is unnecessary by viewing a preview of a reconstructed model, the system goes to the phase C. In the phase C, more detailed 3-D model is generated by the off-line processing. The detailed modeling process generates the 3-D model by using the shape-from-silhouette method in higher resolution of a voxel space than that in the phase A. The system also performs more accurate texture-mapping than the process (B-1) by using area information at the visibility test.

### 3 Experiment

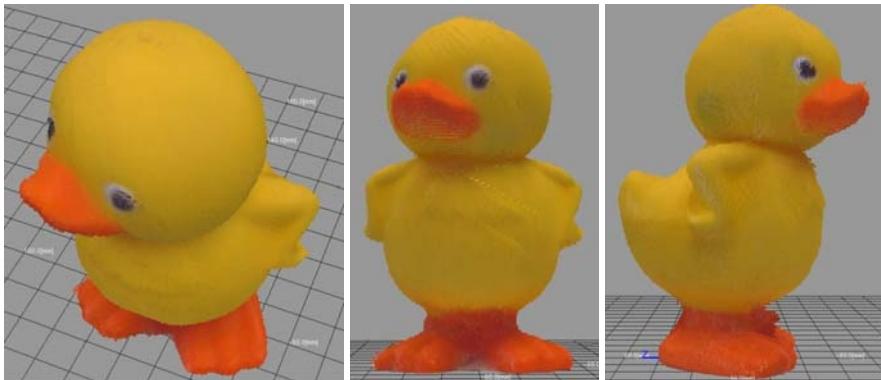
To verify the validity of the proposed system for personal users who have no special skills for modeling, we have carried out experiments with the proposed modeling system. In experiments, a prototype system is developed using a PC (CPU: pentium4 3.2GHz, Memory: 2GB) and a hand-held video camera (capture resolution:  $640 \times 480$  pixels, frame rate: 30 fps). Intrinsic parameters of the camera were estimated by using the Tsai's method [16] in advance. A marker sheet was printed on A3 paper by a laser printer. Figure 10 shows the modeling environment. Figure 11 shows a modeling object. The voxel space for the real-

**Fig. 10.** Modeling Environment**Fig. 11.** Modeling Object**Table 1.** Results of Experiments and Questionnaires

Items	Score
Capturing Time [second]	150.0
Accuracy of Reconstructed Model [1:Bad - 4:Good]	3.3
Capturing Labor [1:Tired - 4:Untired]	3.2

time modeling is constructed of  $64 \times 80 \times 64$  voxels. Fifteen examinees used our system. Seven of 15 examinees are inexperienced in modeling real objects.

After the trials of 3-D modeling by examinees, we sent out questionnaires about the accuracy of the reconstructed model and capturing labor. Table 1 shows results of the experiments and the questionnaires. Figure 12 shows an example of reconstructed detailed model. The voxel space of detailed model is constructed of  $150 \times 150 \times 150$  voxels (voxel size:  $0.86 \times 1.39 \times 0.98$ mm). The average frame rate of the phase A was 7.6 fps. The phase B process took 389 milliseconds on an average. The phase C process took 360 seconds on an average. In experiments, examinees could get good 3-D models by capturing in about 150 seconds, and we verified the usefulness of the system. However, some problems were found in the indication interfaces.



**Fig. 12.** Reconstructed Detailed Model

## 4 Conclusion

In this paper, we have proposed an interactive modeling system using a hand-held video camera. The proposed system has new two functions: “indication of camera movement” and “real-time preview of reconstruction result.” In experiments, we have verified that users who have no special skills for modeling can get a good 3-D model easily in a short time. In future work, the system should be evaluated by more examinees who have no modeling skills, and the indication interface should be reformed.

## References

1. NTT DATA SANYO SYSTEM. Cyber modeler handy light. <http://www.nttd-sanyo.co.jp/>, 2002.
2. UZR GmbH & Co KG. imodeller 3D. <http://www.imodeller.com/en/>, 2001.
3. Leica Geosystems HDS LLC. Hds2500. <http://hds.leica-geosystems.com/>, 2000.
4. KONICA MINOLTA. Vivid 910. <http://konicaminolta.jp/>, 2002.
5. J. E. Banta, L. M. Wong, C. Dumont, and M. A. Abidi. A next-best-view system for autonomous 3D object reconstruction. *IEEE Trans. Systems, Man and Cybernetics*, Vol. 3, No. 5, pp. 589–598, 2000.
6. K. Haga and K. Sato. A shape measurement with support light by a handy projector. *Proc. the 9th Pattern Measurement Symp. on Society of Instrument and Control Engineers (SICE)*, pp. 35–38, 2004 (In Japanese).
7. M. Matsumoto, M. Imura, Y. Yasumuro, Y. Manabe, and K. Chihara. Support system for measurement of relics based on analysis of point clouds. *Proc. the 10th Int. Conf. on Virtual Systems and Multimedia (VSMM)*, p. 195, 2004.
8. L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz. Photometric method for determining surface orientation from multiple images. *Proc. the 9th IEEE Int. Conf. on Computer Vision (ICCV)*, Vol. 1, pp. 618–625, 2003.

9. G. G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer. Methods for volumetric reconstruction of visual scenes. *Int. Journal on Computer Vision (IJCV)*, Vol. 57, No. 3, pp. 179–199, 2004.
10. H. Kim and I. Kweon. Optimal photo hull recovery for the image-based modeling. *Proc. the 6th Asian Conf. on Computer Vision (ACCV)*, Vol. 1, pp. 384–389, 2004.
11. L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. *Proc. the 1st IEEE/ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 27–36, 2002.
12. R. Klette, K. Schluns, and A. koschan, editors. *Computer Vision: Three-dimensional Data from Image*. Springer, 1998.
13. H. Baker. Three-dimensional modeling. *Proc. the 5th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Vol. 2, pp. 649–655, 1977.
14. Y. Kuzu and V. Rodehorst. Volumetric modeling using shape from silhouette. *Proc. the 4th Turkish-German Joint Geodetic Days*, pp. 469–476, 2001.
15. X. Wu, T. Wada, S. Tokai, and T. Matsuyama. Parallel volume intersection based on plane-to-plane projection. *IPSJ Trans. on Computer Vision and Image Media*, Vol. 42, No. SIG6(CVIM2), pp. 33–43, 2001.
16. R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. *Proc. Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 364–374, 1986.

# Automatic Segmentation of the Prostate from Ultrasound Data Using Feature-Based Self Organizing Map

Amjad Zaim

Amman University, Biomedical Engineering Department  
19328 Amman, Jordan  
Zaim\_amjad@yahoo.com

**Abstract.** Traditional segmentation methods cannot provide satisfying results for extraction of prostate gland from Transrectal Ultrasound (TRUS) images because of the presence of strong speckle noise and shadow artifacts. Most ultrasound image segmentation techniques that adopt model-based approach such as active contour are considered semi-automatic because they require initial seeds or contours to be manually identified. In this paper, we propose a method for automatic segmentation of prostate using feature-based self organizing map (SOM). Median filtering and top hat transform are first applied to remove speckle noise. A technique is developed to remove ultrasound-specific speckles using texture-based thresholding. An SOM algorithm is employed to identify prostate pixels taking spatial information, gray-level as well as texture information to form its input vector. The clustered image is then processed to produce a fully connected prostate contour. A number of experiments comparing extracted contours with manually-delineated contours validated the performance of our method.

## 1 Introduction

Prostate cancer is the most commonly diagnosed cancer in men and the second highest North American mortality rate among all cancers in men, surpassed only by lung cancer [1]. Almost all common methods of diagnoses and treatments, such as needle biopsy and brachytherapy, respectively, rely on 2-D or 3-D ultrasound data to accurately locate the prostate gland and device an effective plan for therapy. Transrectal ultrasound utilizes a cylinder shape probe that images the prostate through the rectum and ultimately reconstruct a 3-D model of the prostate [1,2]. TRUS is relatively inexpensive and easy to use compared to other imaging modalities such as MRI or CT. In addition, the safety associated with ultrasound allows for real time monitoring of the prostate gland and accounts for any anatomical displacement or deformation. However, there are several drawbacks of ultrasound compared to other imaging modalities. These include low signal to noise ratio especially in low-contrast regions, the inability to image through bone or air, speckles which arise from constructive-destructive interference of the reflected waves and other artifacts. As a result, most modern treatment planning systems require that the prostate boundaries are manually delineated by an experienced sonographer, which requires extensive labor time and comes at the

expense of spatial resolution particularly when large number of 2D images are available. Many efforts have been aimed at devising automatic or semi-automatic algorithms that could segment the prostate boundaries from the ultrasound images accurately and effectively and with minimal human intervention. A 3D discrete active deformable model was built by researchers to outline the prostate using initial polygonal contours defined in a number of slices and using edge maps to drive the deformation model [3,4]. Others have developed an algorithm for detecting prostate edges as a visual guidance for the user to manually follow [5]. Statistical shape models have also been developed to segment and differentiate between the various shapes of prostates using prior knowledge of the prostate region in ultrasound images. Neural Network has also been utilized to recognize the prostate geometry from a database of prostate shapes. Gabor filtering was designed to extract prostate features and train a KSVM neural network [6]. Adaptive edge-detection methods were also employed [7]. While some of these studies have reported accurate segmentation results, most still require substantial degree of user-interaction either to generate initial contours or to accumulate large number of prostate contours of various shapes. The contribution of the proposed model comes from its ability to detect prostate boundaries with minimal human intervention. The proposed method first applies a set of noise-removal filters to reduce speckle effects. SOM neural network is then employed to cluster similar regions using the spatial information, gray-level as well as texture information to form its input vector. The clustered image is then processed to produce a fully connected prostate contour. A number of experiments comparing the extracted contours with manually-delineated contours validated the performance of our method.

## 2 Segmentation Method

Our segmentation scheme is divided into three major tasks applied sequentially. The first task is to reduce noise and speckles that typically exist in ultrasound images and usually interfere in the segmentation process. This is accomplished via a set of pre-processing morphological operations as well as texture-based thresholding. The second task is to classify image pixels into discrete finite sets of regions including a prostate region via a two-stage SOM. The segmentation process associated with the use of the SOM network employs spatial, texture and gray-level information. Finally, the segmented image is further processed by a set of image processing algorithms to remove holes and scattered regions in the segmented image.

### 2.1 Noise Removal Filtering

**Median Filtering.** Median filter is first applied to reduce impulsive noise. This non-linear filtering modifies the gray-levels of the image while preserving the original information. The center pixel of a 5x5 window is substituted with the median value of all the pixels in the window.

**Top-Hat Transform.** Top-hat transform filter is applied by morphological opening the image by a flat-top hexagon structure element with a radius of 9. Morphological

operation is a well-known procedure in image processing. The opening of an image A is obtained by first eroding the image with a structure element B, after which one performs a dilation on this eroded image with the same structure element [8,9]. Mathematically, opening of A by B is defined as:

$$A \circ B = (A \ominus B) \oplus B \quad (1)$$

This process removes peaks of image surfaces smaller than hexagon leaving slowly varying background. This gray-level variation is further removed by subtracting the opened image from the original image. The resulting image difference image contains little or no information in the regions of low-signal amplitude. The resulting image is shown in the right side of Figure 1.

**Morphological Closing.** To reduce the impact of low-contrast areas left by the previous process, we perform morphological closing of the image with a hexagonal structuring element of radius 6. Morphological closing fuses narrow breaks and long gulfs in binary and gray-scale images [9]. The Closing of an image A by structuring element B, is simply the opposite of opening A by B. Closing here is therefore is defined as the dilation of A by B followed by erosion of the results by B or:

$$A \bullet B = (A \oplus B) \ominus B \quad (2)$$

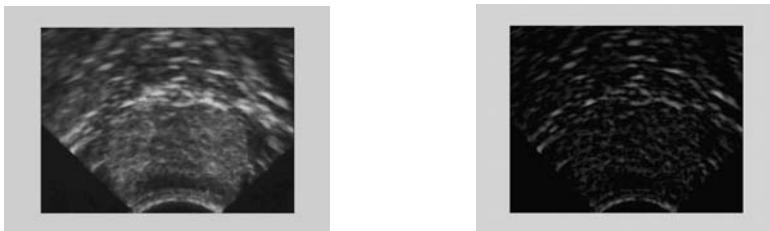
The initial dilation process removes the dark details and brighten the image. The subsequent erosion process darkens the image without reintroducing the details removed by dilation. We used a hexagonal shape structuring element of 6-pixel radius. This radius was chosen to be small enough to follow the details of the prostate contour but large enough to bridge small gaps in the image (Figure 2; left).

**Region-Growing Thresholding.** Looking at the resulting image (Figure 2; left) indicates that the previous filtering technique was able to outline speckle pattern that is visible in the image as connected regions of slightly “banana-shaped” white lines. This type of noise is a result of the impact of sound beam on perpendicular surfaces in the ultrasound field. To identify these speckles, we search through the gray-levels of the image for the point where regions are more connected and hence, less gray-level transitions. Spatial gray level co-occurrence or GLCM estimates image properties related to second order statistics and is one of the most well-known texture features [10]. The contrast measure of GLCM is defined as follows: let  $r$  be a window in the image and  $A_{kl}$  number of pairs of adjacent pixels within  $r$  with grey-values  $k$  and  $l$  respectively. We can define the contrast of  $r$  as:

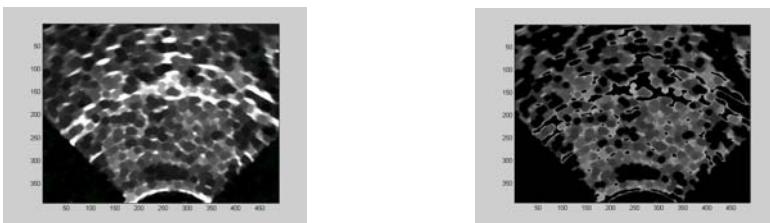
$$\text{Contrast} = \frac{\sum_{k,l=0}^{255} (k-l)^2 A_{k,l}}{\sum_{k,l=0}^{255} A_{k,l}} \quad (3)$$

The contrast is a measure for how many grey-value transitions there are in the region under consideration; the more adjacent pixels with a big difference in grey-value there are in  $r$ , the higher the contrast is. We utilize this feature to search the gray-level

space for the threshold that maximizes the contrast and thus reduces speckles without leaving scattered gaps in the prostate. For our purpose, a threshold of 45 was found to be optimal to remove this type of speckles.



**Fig. 1.** Original TRUS image of the prostate (*left*). The image after applying morphological top-hat transform (*right*)



**Fig. 2.** Result of morphological closing (*left*). The same image after thresholding using GLCM contrast (*right*)

## 2.2 Self-organizing Map

SOM is considered an unsupervised neural network that can serve as a clustering tool for high-dimensional data [11]. It constructs a topology in which the high-dimensional space is mapped onto map units in such a way that relative topology distances between input data are preserved. The map units usually form a two-dimensional regular lattice. In our image every input unit is represented by a 4 dimensional feature vector. The features are center  $x$ ,  $y$  coordinates, the gray-scale level, as well as the contrast texture feature. The  $x$  coordinate and the  $y$ -coordinate encode the spatial information of a pixel, and the gray-level value encode its intensity information. The fourth feature is a texture feature derived from an image block around the pixel of interest as calculated by Equation 1. Each four-dimensional feature vector is regarded as an input vector of the SOM network. The output of the SOM network is  $n$  classes. The SOM network is trained as follows. Each map unit is associated with a reference vector. At first, all the reference vectors are randomly designated. Each input vector is compared with all the reference vectors and the unit whose reference vector is most similar to the input vector is identified. Then, the reference vectors

neighboring to that of the identified unit are moved towards the input vector. Once training is accomplished, input data vectors that are topologically close are mapped to the same class.

Since we use Euclidean distance to measure the competition in SOM learning, it is necessary to normalize the input vectors, and weight features by importance. This prevent features of lesser importance from overriding more important features in the mappings. In this application, we found that the gray level intensities of the pixels are more important than other features to discriminate their classes. The size and the radius of the receptive fields of the SOM are adjustable by architecture parameters [11].

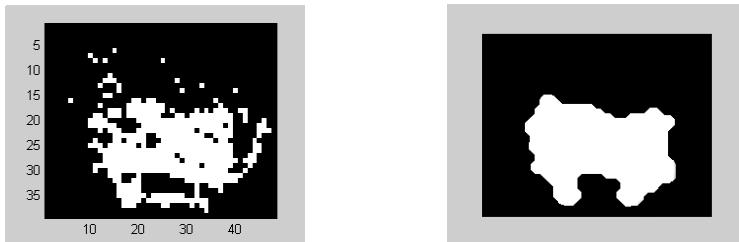
Our SOM consists of 2 layers. The first layer is a 2x3 network oriented in the 2D space. A total of 6 neurons allow Input data vectors that are topologically close to be mapped to the same class. That means the input vector space is divided into 6 classes. Each pixel is associated with a certain class after clustering. A second layer maps the 6 neurons from the first layer into a 2-class output. The two classes correspond to “prostate” and “non-prostate” classes. The first layer helps remove noise but retain textural features that are similar to prostate textures. However, pixels belong to the same class are not always connected. The second SOM layer refines the clustering operation by allowing clusters that are in close proximity to be identified and labeled. The spatial resolution was reduced 5 folds to accommodate the extensive computation of SOM. This is done by simply averaging every 5-pixel size window and assigning the mean value of the pixel intensity  $I$  to the new image pixel  $P$  such that:

$$P_{i,j} = \frac{1}{25} \sum_i^{i+5} \sum_j^{j+5} I_{i,j} \quad (4)$$

where  $i=1,5,\dots,N$  and  $j=1,5,\dots,M$  and the image size is  $M \times N$ . The result is a binary image with a map of the prostate with some scattered isolated points and holes within the contour image (Figure 3).

### 2.3 Removal of Holes and Scattered Points

The resulting binary image contains the prostate with holes within its contour as well as scattered regions outside its contour. We extract the prostate area by using morphological closing and opening operations in succession. These processes are commonly used to smooth, fill in, and/or remove objects in a grayscale or binary image[10]. As noted earlier, morphologic closing is equivalent to a dilation followed by an erosion with a structuring element. Similarly, morphologic opening is equivalent to an erosion followed by a dilation. The erosion stage of opening eliminates the scattered points outside the prostate. The dilation stage of closing fills the holes inside the prostate and restores boundary shapes. In our case, we used a disk-shaped structuring element with a radius of 6 pixels. Experiments have shown that this radius is large enough to fill most common size holes and hence yields satisfying results. As shown in figure 3, the resultant image contains a fully connected prostate region with all scattered points eliminated and all holes filled.



**Fig. 3.** Binary image produced by SOM clustering (*left*) and the result of closing followed by opening of the image (*right*)

### 3 Experimental Results

In this section, we evaluate our algorithm by comparing the algorithm-based segmentations and the manual segmentations on ten US images. The original images are 8-bits pixels of size 489x382. These images, however, were resampled by a ratio of 5 after segmentation for speed at the expense of spatial resolution. We asked one expert radiologist to manually segment 9 of the ultrasound images of 3 different individuals. An error analysis on the overlapping area between the segmented areas using the manual and automatic segmentation method is shown in Table 1. In order to calculate the maximal shortest distance error, we find the distance to the closest point on the contour drawn by the expert and we take the maximum of the distances over the contour produced by the algorithm. The overlap area error is the overlap between the manual segmentation and automatic segmentation contours. Our algorithm has demonstrated an accuracy of at least 91%. We believe that the degraded spatial resolution caused by subsampling presents a major source of error that can be drastically reduced if the full image size is considered. The speed of our algorithm has also been tested on a regular 789 MHz desktop PC and recorded an average execution time of 12 seconds.

**Table 1.** Comparison of the automated and the hand labeled segmentation results

Individual	Distance (Pixels)	Overlap Area Error%
1	9	2.6
2	17	8.7
3	13	5

### 4 Conclusion

An automatic segmentation scheme has been presented in this paper, for extracting prostate from TRUS images. The proposed method encodes gray-level features of prostate TRUS images to be used as input to SOM network. While our method does not give high accuracy compared with other contour deformation methods, it requires

no human intervention at any point in the segmentation process. Future work will focus on a hierarchical strategy that incorporates multiresolution information and improves speed.

## References

1. Zaim, A., Keck, R., Selman, S., Jankun, J. "Three-Dimensional Ultrasound Image Matching System for Photodynamic Therapy", *Proceedings of BIOS-SPIE*, vol. 4244: 327-337, 2001.
2. Jankun, J., Zaim, A. "An image-guided robotic System for photodynamic Therapy of the Prostate," *SPIE Proceeding*, vol. 39, pp22-22, 1999.
3. Ghanei, A., Soltanian-Zadeh, H., Ratkesic, A., Yin, F. "A three-dimensional deformable model for segmentation of human prostate from ultrasound image", *Medical Physics*, Vol. 28, pp. 2147-2153, 2001.
4. Hu, N., Downey, D., Fenster, A., Ladak, H. "Prostate surface segmentation from 3D ultrasound images", *IEEE International Symposium on Biomedical Imaging*, pp. 613-616, Washington, D. C., 2002.
5. Pathak, S., Chalana, V., Haynor, D., Kim, Y. "Edge-Guided Boundary Delineation in Prostate Ultrasound Images", *IEEE Trans. Med. Img.*, Vol. 19, pp. 1211-1219, 2000.
6. Shen, D., Zhan, Y., Davatzikos, C. "Segmentation Prostate Boundaries from Ultrasound Images Using Statistical Shape Model", *IEEE Trans. On Med. Img.*, Vol.22, pp. 539-551, Apr.2003.
7. Aarnink, R., Huyanen, A., Giesen, J., De la Rosette, D., Debruyne, F., Wijkstra, H. "Automated prostate volume determination with ultrasonographic imaging," *Journal of Urology*, vol. 155, pp. 1038-1039, 1996.
8. Castleman, R. *Digital Image Processing*, Upper Saddle River, New Jersey: Prentice-Hall, 1996.
9. Niblack, W. *An Introduction to Digital Image Processing*, Massachusetts: Upper Saddle River, New Jersey: Prentice-Hall, 1996.
10. Gonzalez, R. *Digital Image Processing*, Massachusetts: Addison-Wisely, 1996.
11. Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, 2001.

# Author Index

- Aanæs, Henrik 551  
Agartz, Ingrid 272  
Ageenko, Eugene 1107  
Åhlén, Julia 1148  
Ahola, Heikki 141  
Ahola, Jero 970  
Ahonen, Timo 882  
Akimov, Alexander 312  
Al-Jarwan, I.A. 1196  
Alavi, F.N. 1208  
Alberola-López, Carlos 1117  
Althoff, K. 282  
Altrichter, Márta 760  
Anderson, Jakob 567  
Andersson, Fredrik 950  
Andersson, Mats 292, 1086  
Andersson, Mattias 105  
Andreopoulos, Alexander 729  
Aoki, Kohta 65  
Ardo, Hakan 449  
Astola, Jaakko 1177  
Åström, Kalle 182, 609, 740  
Austvoll, Ivar 659
- Barata, Teresa 429  
Bartoli, A. 531  
Baruthio, J. 263  
Bednarik, Roman 780  
Beichel, Reinhard 481  
Bengtsson, E. 1148  
Bernarding, Johannes 302  
Bhalerao, Abhir 439  
Bhattacharya, Bhargab B. 930  
Bhowmick, Partha 930  
Bischof, Horst 45, 481  
Biswas, Arindam 930  
Bochko, Vladimir 389, 1218  
Bogunović, Hrvoje 1157  
Bogush, A.L. 1066  
Boldo, Didier 831  
Borgefors, Gunilla 253  
Bornefalk, Hans 649  
Bowen, Adam 85  
Brandt, S.S. 577
- Brekke, Camilla 75  
Brooks, Steve P. 429  
Brun, Anders 920  
Burkhardt, H. 841
- Calitouiu, Dragos 821  
Carrasco, Miguel 1238  
Carstensen, Jens M. 1228  
Cecchi, Guillermo 810  
Čech, Jan 598  
Cherednichenko, Svetlana 978  
Chihara, Kunihiro 161, 399  
Chillet, D. 263  
Christmas, W.J. 343  
Cleju, Ioan 872
- d'Angelo, P. 689  
Degerman, J. 282  
de Luis-García, Rodrigo 1117  
de Vieilleville, François 988  
Deriche, Rachid 1117  
Di Gesù, Vito 184  
Diamant, Emanuel 17  
Divjak, Matjaž 619  
Doi, M. 95  
Domingues, Antonio 1076  
Driouchi, Driss 1076  
Duin, R.P.W. 998, 1009
- Edenbrandt, Lars 740  
Egiazarian, Karen 1177  
Eidheim, Ole Christian 750  
El-Baz, Ayman 1128, 1138  
Ellenrieder, Marc M. 669  
Ericsson, Anders 709, 719, 740  
Ersbøll, Bjarne K. 1228
- Farag, Aly 1128, 1138  
Felsberg, Michael 491  
Feschet, Fabien 910, 988  
Fränti, Pasi 312, 780, 872, 978  
Fraundorfer, Friedrich 45  
Friberg, Lars 740  
Fritz, Gerald 629, 639  
Frydrych, M. 151

- Fudono, Kenji 1248  
 Fukunaga, Kunio 802
- Gaspard, F. 531  
 Georgsson, Fredrik 470  
 Gimelfarb, Georgy 1128, 1138  
 Girdziušas, Ramūnas 1096  
 Gurevich, Igor 214  
 Gustavsson, T. 282
- Haasdonk, B. 841  
 Hagita, Norihiro 130  
 Haindl, Michal 1037  
 Häme, Tuomas 141  
 Hamouz, M. 119  
 Hanbury, Allan 35  
 Hanheide, Marc 669  
 Hansen, Michael E. 1228  
 Happonen, A.P. 1047  
 Hauta-Kasari, Markku 369  
 Hautamäki, Ville 978  
 Heikkilä, Janne 224  
 Herberthson, Magnus 920  
 Hiraishi, Akira 419  
 Horváth, Gábor 760  
 Huang, Cherng-yue 359  
 Hult, Roger 272  
 Hung, Chih Cheng 511
- Iivarinen, Jukka 588  
 Iilonen, Jarmo 119, 970  
 Imura, Masataka 161, 399  
 Izumi, Masao 802
- Jaaskelainen, Timo 369  
 Jacob-Da Col, M.-A. 263  
 Jain, Anil K. 1  
 Jan, Jiří 1017  
 Jensen, Nils 302  
 Jonsson, Erik 491  
 Josephson, Klas 719
- Kalenova, Diana 389  
 Kalliomäki, I. 940  
 Kälviäinen, Heikki 119, 409, 970  
 Kamarainen, Joni-Kristian 119, 409, 970  
 Kannala, Juho 224  
 Kärkkäinen, Ismo 978  
 Karlsson, Johan 709, 719  
 Kätsyri, J. 151  
 Kinnunen, Tomi 780, 978
- Kirkegaard, Jakob 679  
 Kitahara, Itaru 130  
 Kittler, Josef 119, 343  
 Knutsson, Hans 292, 501, 920, 1086  
 Kobayashi, Ken-ichi 419  
 Kogure, Kiyoshi 130  
 Kolář, Radim 1017  
 Kolesnikov, Alexander 312, 1186  
 Kolonias, I. 343  
 Korkkalainen, Ate 1218  
 Kostin, A. 343  
 Kouropeteva, Olga 521  
 Kozloski, James 810  
 Krüger, Lars 669  
 Krüger, Volker 567  
 Kruse, Björn 105  
 Kumar, Manish 629  
 Kunttu, Iivari 892, 901  
 Kuparinen, Toni 1218  
 Kyrki, Ville 557
- Laaksonen, Jorma 770, 1096  
 Laamanen, Hannu 369  
 Lachaud, Jacques-Olivier 988  
 Lampinen, J. 151, 940  
 Larsen, Rasmus 205  
 Lavest, J.-M. 531  
 Lee, Jiann-Shu 359  
 Lensu, Lasse 409  
 Lenz, Reiner 105  
 Lepistö, Leena 892, 901  
 Lettner, Martin 459  
 Liaw, Chishyan 359  
 Lindell, T. 1148  
 Lindh, Tuomo 970  
 Lingrand, Diane 25  
 Lončarić, Sven 1157  
 Lonsdale, Markus Nowak 740  
 Lucas, Yves 1076  
 Ludányi, Zoltán 760  
 Lukin, Vladimir 1177
- Machida, Takashi 790  
 Manabe, Yoshitsugu 161, 399  
 Marché, Pierre 1076  
 Martinsson, H. 531  
 Matas, Jiri 541  
 Megson, Graham 1208  
 Mery, Domingo 1238  
 Mester, Rudolf 322

- Mičušík, Branislav 35  
 Mielikainen, Jarno 1218  
 Mihaila, Andrei 780  
 Miyata, Kimiyoshi 369  
 Miyazawa, Kanae 419  
 Moeslund, Thomas B. 679  
 Molinier, Matthieu 141  
 Montagnat, Johan 25  
 Mühllich, Matthias 322  
 Mullins, Andrew 85  
 Murase, Hiroshi 130  
 Nagahashi, Hiroshi 65  
 Naganawa, Mika 399  
 Nagy, Benedek 1027  
 Nakauchi, Shigeki 55, 419  
 Nejdl, Wolfgang 302  
 Nie, Zhengang 1208  
 Nielsen, Allan A. 1228  
 Nilsson, Jens 950  
 Nusbaum, Dorin 821  
 Oe, Motoko 171  
 Ohtsuki, R. 95  
 Oja, Erkki 333  
 Okun, Oleg 521  
 Olsén, Christina 470  
 Olsen, Søren I. 852  
 Oommen, B. John 821  
 Opelt, Andreas 862  
 Oskarsson, M. 609  
 Paalanen, Pekka 119, 970  
 Pacák, P. 998, 1009  
 Palander, K. 577  
 Paletta, Lucas 629, 639  
 Parkkinen, Jussi 369  
 Partanen, Jarmo 970  
 Parviainen, Juha 1218  
 Passat, N. 263  
 Payne, Janet S. 960  
 Peck, Charles 810  
 Pettersson, Johanna 501  
 Pietikäinen, Matti 115, 521, 882  
 Pinz, Axel 862  
 Pock, Thomas 481  
 Podlasov, Alexey 1107  
 Ponomarenko, Nikolay 1177  
 Prehn, Thomas 567  
 Priese, Lutz 6  
 Rahtu, Esa 224  
 Rajpoot, Nasir 85  
 Rao, A. Ravishankar 810  
 Rautkorpi, Rami 588  
 Rodionov, Oleg 1218  
 Rosin, Paul L. 195  
 Rousson, Mikael 1117  
 Ruotsalainen, U. 1047  
 Šára, Radim 598  
 Saatchi, Sara 511  
 Sablatník, Robert 459  
 Sadovnikov, Albert 409  
 Saito, Hideo 130  
 Salmela, Petja 409  
 Salo, Mikko 224  
 Sasaki, Hiroshi 161  
 Sato, Tomokazu 171, 1248  
 Schmock, Kerstin 557  
 Seifert, Christin 629, 639  
 Šimberová, Stanislava 1037  
 Sintorn, Ida-Maria 253  
 Skjermo, Jo 750  
 Solberg, Anne H.S. 75  
 Solem, Jan Erik 551  
 Solli, Martin 105  
 Stewénius, H. 609  
 Stöbel, Dirk 669  
 Stonham, John 960  
 Strand, Robin 1057  
 Sturm, Patrick 6  
 Sundgren, D. 1148  
 Svensson, Björn 1086  
 Taillandier, Franck 699  
 Takala, Valtteri 882  
 Takeda, Tadayuki 161  
 Takemura, Haruo 790  
 Tamminen, T. 151  
 Tanaka, Hidenori 130  
 Toivanen, Pekka 389, 1218  
 Tominaga, Shoji 95, 379  
 Trias-Sanz, Roger 831  
 Tsai, Ching-tsorng 359  
 Tsotsos, John K. 729  
 Tuzikov, A.V. 1066  
 Uegaki, Naoki 802  
 Uranishi, Yuuki 399

- Valantinas, Jonas 1167  
Vartiainen, Erik 1218  
Verzakov, S. 998, 1009  
Visa, Ari 892, 901  
Voigt, Gabriele von 302  
Vossen, A. 841  
  
Wöhler, C. 689  
Wang, Haojun 6  
Westin, Carl-Fredrik 920  
Wilson, Roland 85, 439  
Windridge, D. 343  
Winter, Martin 45  
Wrangsjö, Andreas 501  
  
Wu, Xiaolin 235, 872  
Wu, Yunsong 1208  
  
Yan, F. 343  
Yang, Zhirong 770  
Yasumuro, Yoshihiro 161, 399  
Yokoya, Naokazu 171, 790, 1248  
Yuan, Zhijian 333  
  
Zaim, Amjad 1258  
Zavidovique, Bertrand 184  
Zazula, Damjan 619  
Zemerly, M.J. 1196  
Zhang, Lei 235  
Zimmermann, Karel 541