

Leonard Bolc
Ryszard Tadeusiewicz
Leszek J. Chmielewski
Konrad Wojciechowski (Eds.)

LNCS 6375

Computer Vision and Graphics

International Conference, ICCVG 2010
Warsaw, Poland, September 2010
Proceedings, Part II

2
Part II

 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

Alfred Kobsa

University of California, Irvine, CA, 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

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Leonard Bolc Ryszard Tadeusiewicz
Leszek J. Chmielewski
Konrad Wojciechowski (Eds.)

Computer Vision and Graphics

International Conference, ICCVG 2010
Warsaw, Poland, September 20-22, 2010
Proceedings, Part II

Volume Editors

Leonard Bolc
Konrad Wojciechowski
Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland,
E-mail: {pjawst, konradw}@pjawst.edu.pl

Ryszard Tadeusiewicz
AGH University of Science and Technology, Institute of Automatics
Al. Mickiewicza 30, 30-059, Kraków, Poland
E-mail: rtad@agh.edu.pl

Leszek J. Chmielewski
Warsaw University of Life Sciences
Faculty of Applied Informatics and Mathematics
Department of Informatics
Nowoursynowska 159, 02-776 Warsaw, Poland
E-mail: leszek_chmielewski@sggw.pl

Konrad Wojciechowski
The Silesian University of Technology, Institute of Computer Science
Akademicka 16, 44-100 Gliwice, Poland
E-mail: konrad.wojciechowski@polsl.pl

Library of Congress Control Number: 2010934223

CR Subject Classification (1998): I.4, I.2.10, I.5, I.3.5, I.2, I.3

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition,
and Graphics

ISSN 0302-9743
ISBN-10 3-642-15906-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15906-0 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.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The International Conference on Computer Vision and Graphics, ICCVG, organized since 2002, is the continuation of the International Conferences on Computer Graphics and Image Processing, GKPO, held in Poland every second year from 1990 to 2000. The founder and organizer of these conferences was Prof. Wojciech Mokrzycki. The main objective of ICCVG is to provide a forum for the exchange of ideas between researchers in the closely-related domains of computer vision and computer graphics.

ICCVG 2010 gathered about 100 authors. The proceedings contain 95 papers, each accepted on the grounds of two independent reviews. During the conference two special sessions were organized: *Advances in Pattern Recognition, Machine Vision and Image Understanding* and *Human Motion Analysis and Synthesis*.

The content of the issue has been divided into three parts. The first and second parts are related to the two special sessions mentioned above, containing 11 chapters each. The third part, named like the whole book, Computer Vision and Graphics, collects all the remaining chapters.

ICCVG 2010 was organized by the Association for Image Processing, Poland (Towarzystwo Przetwarzania Obrazów – TPO), the Polish-Japanese Institute of Information Technology (PJWSTK), and the Faculty of Applied Informatics and Mathematics, Warsaw University of Life Sciences (WZIM SGGW).

The Association for Image Processing integrates the Polish community working on the theory and applications of computer vision and graphics. It was formed between 1989 and 1991.

The Polish-Japanese Institute of Information Technology, founded in 1994 by the Computer Techniques Development Foundation under the agreement of the Polish and Japanese governments, is one of the leading, non-state (private) Polish universities. We are highly grateful for the fact that the institute has been hosting and supporting the Conference.

The Faculty of Applied Informatics and Mathematics, established in 2008 at the Warsaw University of Life Sciences, offers two programs of studies: Informatics, and Informatics and Econometrics. Its main advantage is its focus on merging technical education with applied sciences, including application of computer sciences to the management and analysis of the agricultural industry.

We would like to thank all the members of the Scientific Committee, as well as the additional reviewers, for their help in ensuring the high quality of the papers. We would also like to thank Grażyna Domańska-Żurek for her excellent work on technically editing the proceedings, and Bernadeta Bonio and Paweł Wieman for their engagement in the conference organization and administration.

September 2010

Leonard Bolc
Ryszard Tadeusiewicz
Leszek J Chmielewski
Konrad Wojciechowski

Organization

- Association for Image Processing (TPO)
- Polish-Japanese Institute of Information Technology (PJWSTK)
- Faculty of Applied Informatics and Mathematics,
Warsaw University of Life Sciences (WZIM SGGW)

Conference General Chairs

J.L. Kulikowski (Poland)
L.J. Chmielewski (Poland)
K. Wojciechowski (Poland)

Conference Committee Members

A. Bagdanov (Spain)	N. Kiryati (Israel)	A. Przelaskowski (Poland)
I. Bajla (Austria)	R. Klette (New Zealand)	J. Punys (Lithuania)
E. Bengtsson (Sweden)	J. Korbicz (Poland)	J. Roerdink (The Netherlands)
M. Berndt-Schreiber (Poland)	R. Kozera (Australia)	P. Rokita (Poland)
P. Bhattacharya (USA)	K. Krawiec (Poland)	L. Rutkowski (Poland)
G. Borgefors (Sweden)	H.J. Kreowski (Germany)	K. Saeed (Poland)
A. Borkowski (Poland)	A. Krzyżak (Canada)	G. Sanniti di Baja (Italy)
D. Chetverikov (Hungary)	M. Kurzyński (Poland)	R. Šára (Czech Republic)
L. Chmielewski (Poland)	B. Kwolek (Poland)	G. Schaefer (UK)
L. Czuni (Hungary)	Y.B. Kwon (South Korea)	B. Scotney (Ireland)
S. Dellepiane (Italy)	A. Materka (Poland)	V. Skala (Czech Republic)
M. Domański (Poland)	N. Mavridis (United Arab Emirates)	W. Skarbek (Poland)
M. Flasiński (Poland)	W. Mokrzycki (Poland)	A.N. Skodras (Greece)
B. Gabryś (UK)	H. Niemann (Germany)	G. Sommer (Germany)
A. Gagalowicz (France)	M. Nieniewski (Poland)	C. Stępień (Poland)
G. Giacinto (Italy)	S. Nikiel (Poland)	A. Śluzek (Singapore)
D. Gillies (UK)	L. Noakes (Australia)	R. Tadeusiewicz (Poland)
E. Hancock (UK)	A. Nowakowski (Poland)	J.M.R.S. Tavares (Portugal)
L. Heutte (France)	M. Ogiela (Poland)	K. Tombre (France)
V. Hlavac (Czech Republic)	M. Orkisz (France)	E. Derya Übeyli (Turkey)
M. Iwanowski (Poland)	R.M. Palenichka (Canada)	V. Valev (Bulgaria and Saudi Arabia)
A. Józwik (Poland)	H. Palus (Poland)	W. Malina (Poland)
H. Kälviäinen (Finland)	W. Pamula (Poland)	M. Woźniak (Poland)
A. Kasiński (Poland)	F.J. Perales López (Spain)	Z. Wróbel (Poland)
J. Kacprzyk (Poland)	J. Piecha (Poland)	J. Zabrodzki (Poland)
W. Kasprzak (Poland)	E. Piętka (Poland)	W. Zamojski (Poland)
R. Kasturi (USA)	P. Porwik (Poland)	M. Zaremba (Canada)

Table of Contents – Part II

Computer Vision and Graphics

A Constraint Satisfaction Framework with Bayesian Inference for Model-Based Object Recognition	1
<i>Włodzimierz Kasprzak, Lukasz Czajka, and Artur Wilkowski</i>	
Visual Programming Environment Based on Hypergraph Representations	9
<i>Peter Kapec</i>	
MRI Brain Segmentation Using Cellular Automaton Approach	17
<i>Rafał Henryk Kartaszyński and Paweł Mikołajczak</i>	
Improved Context-Based Adaptive Binary Arithmetic Coding in MPEG-4 AVC/H.264 Video Codec	25
<i>Damian Karwowski and Marek Domański</i>	
Local Polynomial Approximation for Unsupervised Segmentation of Endoscopic Images	33
<i>Artur Klepaczko, Piotr Szczypiński, Piotr Daniel, and Marek Pazurek</i>	
A Proper Choice of Vertices for Triangulation Representation of Digital Images	41
<i>Ivana Kolingerová, Josef Kohout, Michal Rulf, and Václav Uher</i>	
A Method for Novel Face View Synthesis Using Stereo Vision	49
<i>Jacek Komorowski and Przemysław Rokita</i>	
Outer Surface Reconstruction for 3D Fractured Objects	57
<i>Anatoly Kornev, Laurent Babout, Marcin Janaszewski, and Hugues Talbot</i>	
Application of the FraDIA Vision Framework for Robotic Purposes	65
<i>Tomasz Kornuta</i>	
Video Analysis Based on Mutual Information	73
<i>Lenka Krulíková, Michal Mardiak, Juraj Pavlovic, and Jaroslav Polec</i>	
Speeding Up Powerful State-of-the-Art Restoration Methods with Modern Graphics Processors	81
<i>Tim Kubertschak, Rico Nestler, Torsten Machleidt, and Karl-Heinz Franke</i>	

Multi-layered Framebuffer Condensation: The l-buffer Concept	89
<i>Jarosław Konrad Lipowski</i>	
Accurate Overlap Area Detection Using a Histogram and Multiple Closest Points	98
<i>Yonghuai Liu, Ralph R. Martin, Longzhuang Li, and Baogang Wei</i>	
Automatic Extraction of the Lower Boundary of the Mandibular Bone in Dental Panoramic Radiographs	110
<i>Przemysław Maćkowiak</i>	
Football Player Detection in Video Broadcast	118
<i>Stawomir Maćkowiak, Jacek Konieczny, Maciej Kurc, and Przemysław Maćkowiak</i>	
Spectrum Evaluation on Multispectral Images by Machine Learning Techniques	126
<i>Marcin Michalak and Adam Świtoński</i>	
Fatigue Detector Using Eyelid Blinking and Mouth Yawning	134
<i>Helmi Adly Mohd Noor and Rosziati Ibrahim</i>	
Inferior Maxillary Bone Tissue Classification in 3D CT Images	142
<i>Silvia Moreno, Sandra L. Caicedo, Tonny Strulovic, Juan C. Briceño, Fernando Briceño, Soledad Gómez, and Marcela Hernández Hoyos</i>	
Vector Median Splatting for Image Based Rendering	150
<i>Krzysztof Okarma, Aleksandra Miętus, and Mateusz Teclaw</i>	
Feature Extraction Using Reconfigurable Hardware	158
<i>Wiesław Pamula</i>	
Automated Counting and Characterization of Dirt Particles in Pulp	166
<i>Maryam Panjeh Fouladgaran, Aki Mankki, Lasse Lensu, Jari Käyhkö, and Heikki Kälviäinen</i>	
Keypoint-Based Detection Near-Duplicate Image Fragments Using Image Geometry and Topology	175
<i>Mariusz Paradowski and Andrzej Śluzek</i>	
Three Cameras Method of Light Sources Extraction in Augmented Reality	183
<i>Przemysław Wiktor Pardel and Konrad Wojciechowski</i>	
The Context-Sensitive Grammar for Vehicle Movement Description	193
<i>Jan Piecha and Marcin Staniek</i>	
The Method for Verifying Correctness of the Shape's Changes Calculation in the Melting Block of Ice	203
<i>Maria Pietruszka and Dominik Szajerman</i>	

A Real Time Vehicle Detection Algorithm for Vision-Based Sensors	211
<i>Bartłomiej Placzek</i>	
Sequential Reduction Algorithm for Nearest Neighbor Rule	219
<i>Marcin Raniszewski</i>	
GPU-Supported Object Tracking Using Adaptive Appearance Models and Particle Swarm Optimization	227
<i>Bogusław Rymut and Bogdan Kwolek</i>	
GFT: GPU Fast Triangulation of 3D Points	235
<i>Jairo R. Sánchez, Hugo Álvarez, and Diego Borro</i>	
Methods for Visualization of Bone Tissue in the Proximity of Implants	243
<i>Hamid Sarve, Joakim Lindblad, Carina B. Johansson, and Gunilla Borgfors</i>	
An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation	251
<i>Adam Schmidt, Marek Kraft, and Andrzej Kasiński</i>	
The Visual SLAM System for a Hexapod Robot	260
<i>Adam Schmidt and Andrzej Kasiński</i>	
Displacement Calculation of Heart Walls in ECG Sequences Using Level Set Segmentation and B-Spline Free Form Deformations	268
<i>Andrzej Skalski, Paweł Turcza, and Tomasz Zieliński</i>	
Fast and Accurate Machined Surface Rendering Using an Octree Model	276
<i>Joanna Porter-Sobieraj and Maciej Świechowski</i>	
PATSI — Photo Annotation through Finding Similar Images with Multivariate Gaussian Models	284
<i>Michał Stanek, Bartosz Broda, and Halina Kwasnicka</i>	
Generation of Temporally Consistent Depth Maps Using Noise Removal from Video	292
<i>Olgierd Stankiewicz and Krzysztof Wegner</i>	
Smooth Detail Features on Multiresolution Surface	300
<i>Nanik Suciati and Koichi Harada</i>	
Using Parallel Graph Transformations in Design Support System	309
<i>Rafał Świdorski and Barbara Strug</i>	
Fast, Parallel Watershed Algorithm Based on Path Tracing	317
<i>Michał Świercz and Marcin Iwanowski</i>	

Detection of Tumor Tissue Based on the Multispectral Imaging	325
<i>Adam Świtoński, Marcin Michalak, Henryk Josiński, and Konrad Wojciechowski</i>	
A New Method for ECG Signal Feature Extraction	334
<i>Adam Szczepański, Khalid Saeed, and Alois Ferscha</i>	
Error Concealment Method Selection in Texture Images Using Advanced Local Binary Patterns Classifier	342
<i>Želmira Tóthová, Jaroslav Polec, Tatiana Orgoniková, and Lenka Krulíková</i>	
Early Warning System for Air Traffic Control Using Kinetic Delaunay Triangulation	350
<i>Tomáš Vomáčka and Ivana Kolingerová</i>	
GPU Calculated Camera Collisions Detection within a Dynamic Environment	357
<i>Adam Wojciechowski and Grzegorz Wróblewski</i>	
Utilization of Multi-spectral Images in Photodynamic Diagnosis	367
<i>Andrzej Zacher</i>	
The Spectral Analysis of Human Skin Tissue Using Multi-spectral Images	376
<i>Andrzej Zacher</i>	
Can the Coronary Artery Centerline Extraction in Computed Tomography Images Be Improved by Use of a Partial Volume Model?	385
<i>Maria A. Zuluaga, Edgar J.F. Delgado Leyton, Marcela Hernández Hoyos, and Maciej Orkisz</i>	
Author Index	393

Table of Contents – Part I

Advances in Pattern Recognition, Machine Vision and Image Understanding

Visual Codebooks Survey for Video On-Line Processing	1
<i>Vítězslav Beran and Pavel Zemčík</i>	
Application of Shape Description Methodology to Hand Radiographs Interpretation	11
<i>Marzena Bielecka, Andrzej Bielecki, Mariusz Korkosz, Marek Skomorowski, Wadim Wojciechowski, and Bartosz Zieliński</i>	
Localisation and Tracking of an Airport’s Approach Lighting System . . .	19
<i>Shyama Prosad Chowdhury, Karen Rafferty, and Amit Kumar Das</i>	
Algorithm for Blood-Vessel Segmentation in 3D Images Based on a Right Generalized Cylinder Model: Application to Carotid Arteries	27
<i>Leonardo Flórez Valencia, Jacques Azencot, and Maciej Orkisz</i>	
Cognitive Hierarchical Active Partitions Using Patch Approach	35
<i>Konrad Jojczyk, Michał Pryczek, Arkadiusz Tomczyk, Piotr S. Szczepaniak, and Piotr Grzelak</i>	
Ontological Models as Tools for Image Content Understanding	43
<i>Juliusz L. Kulikowski</i>	
Unsupervised, Fast and Precise Recognition of Digital Arcs in Noisy Images	59
<i>Thanh Phuong Nguyen, Bertrand Kerautret, Isabelle Debled-Rennesson, and Jacques-Olivier Lachaud</i>	
The Role of Sparse Data Representation in Semantic Image Understanding	69
<i>Artur Przelaskowski</i>	
Semantic Interpretation of Heart Vessel Structures Based on Graph Grammars	81
<i>Miroslaw Trzupek</i>	
Interpretation of Images and Their Sequences Using Potential Active Contour Method	89
<i>Stanisław Walczak, Arkadiusz Tomczyk, and Piotr S. Szczepaniak</i>	
Inductive Learning Methods in the Simple Image Understanding System	97
<i>Krzysztof Wójcik</i>	

Human Motion Analysis and Synthesis

A Generic Approach to Design and Querying of Multi-purpose Human Motion Database	105
<i>Wiktor Filipowicz, Piotr Habela, Krzysztof Kaczmarek, and Marek Kulbacki</i>	
Surveillance Video Stream Analysis Using Adaptive Background Model and Object Recognition	114
<i>Paweł Forczmański and Marcin Seweryn</i>	
Nonlinear Multiscale Analysis of Motion Trajectories	122
<i>Bartosz Jabłoński and Marek Kulbacki</i>	
Matlab Based Interactive Simulation Program for 2D Multisegment Mechanical Systems	131
<i>Henryk Josiński, Adam Świtoński, Karol Jędrasiak, Andrzej Polański, and Konrad Wojciechowski</i>	
Electrooculography Signal Estimation by Using Evolution-Based Technique for Computer Animation Applications	139
<i>Robert Krupiński and Przemysław Mazurek</i>	
Articulated Body Motion Tracking by Combined Particle Swarm Optimization and Particle Filtering	147
<i>Tomasz Krzeszowski, Bogdan Kwolek, and Konrad Wojciechowski</i>	
GPU-Accelerated Tracking of the Motion of 3D Articulated Figure	155
<i>Tomasz Krzeszowski, Bogdan Kwolek, and Konrad Wojciechowski</i>	
An Efficient Approach for Human Motion Data Mining Based on Curves Matching	163
<i>Van-Hanh Nguyen, Frederic Merienne, and Jean-Luc Martinez</i>	
Estimation System for Forces and Torques in a Biped Motion	185
<i>Andrzej Polański, Adam Świtoński, Henryk Josiński, Karol Jędrasiak, and Konrad Wojciechowski</i>	
Classification of Poses and Movement Phases	193
<i>Adam Świtoński, Henryk Josiński, Karol Jędrasiak, Andrzej Polański, and Konrad Wojciechowski</i>	

Computer Vision and Graphics

Region Covariance Matrix-Based Object Tracking with Occlusions Handling	201
<i>Ivar Austvoll and Bogdan Kwolek</i>	

Minimalist AdaBoost for Blemish Identification in Potatoes	209
<i>Michael Barnes, Grzegorz Cielniak, and Tom Duckett</i>	
The Colour Sketch Recognition Interface for Training Systems	217
<i>Marcin Bernaś</i>	
Interactive Hydraulic Erosion Using CUDA	225
<i>Richard Bezin, Alexandre Peyrat, Benoit Crespin, Olivier Terraz, Xavier Skapin, and Philippe Meseure</i>	
Smoothing, Enhancing Filters in Terms of Backward Stochastic Differential Equations	233
<i>Dariusz Borkowski</i>	
An Analysis of Different Clustering Algorithms for ROI Detection in High Resolutions CT Lung Images	241
<i>Alfonso Castro, Carmen Bóveda, Alberto Rey, and Bernardino Arcay</i>	
A System to Measure Gap Distance between Two Vehicles Using License Plate Character Height	249
<i>Kin Yip Chan, Andrzej Ordys, and Olga Duran</i>	
Efficient Neural Models for Visual Attention	257
<i>Sylvain Chevallier, Nicolas Cuperlier, and Philippe Gaussier</i>	
Fuzzy Hough Transform-Based Methods for Extraction and Measurements of Single Trees in Large-Volume 3D Terrestrial LIDAR Data	265
<i>Leszek J. Chmielewski, Marcin Bator, Michał Zasada, Krzysztof Stereńczak, and Paweł Strzebiński</i>	
Image Recognition Techniques Applied to Automated and Objective QoE Assessment of Mobile WWW Services	275
<i>Michał Choraś, Adam Flizikowski, Rafał Kozik, and Witold Hołubowicz</i>	
Detection of Near-Regular Object Configurations by Elastic Graph Search	283
<i>Gerardus Croonen and Csaba Beleznai</i>	
Traffic Scene Segmentation and Robust Filtering for Road Signs Recognition	292
<i>Bogusław Cyganek</i>	
Fast Distance Vector Field Extraction for Facial Feature Detection	300
<i>Wojciech Czarnecki, Szymon Gburek, and Maciej Smiatacz</i>	
Vision-Based Vehicle Speed Measurement Method	308
<i>Witold Czajewski and Marcin Iwanowski</i>	

A Particle-Based Method for Large-Scale Breaking Waves Simulation . . .	316
<i>Emmanuelle Darles, Benoit Crespin, and Djamchid Ghazanfarpour</i>	
A Hierarchical Classification Method for Mammographic Lesions Using Wavelet Transform and Spatial Features	324
<i>Silvana G. Dellepiane, Irene Minetti, and Sara Dellepiane</i>	
Easy Rigging of Face by Automatic Registration and Transfer of Skinning Parameters	333
<i>Ludovic Dutreue, Alexandre Meyer, Veronica Orvalho, and Saïda Bouakaz</i>	
Terrain Modeling with Multifractional Brownian Motion and Self-regulating Processes	342
<i>Antoine Echelard, Jacques Lévy Véhel, and Olivier Barrière</i>	
Image Encryption through Using Chaotic Function and Graph	352
<i>Sa'ed Faridnia and Karim Fae'z</i>	
Robust Stamps Detection and Classification by Means of General Shape Analysis	360
<i>Paweł Forczmański and Dariusz Frejlichowski</i>	
Pre-processing, Extraction and Recognition of Binary Erythrocyte Shapes for Computer-Assisted Diagnosis Based on MGG Images	368
<i>Dariusz Frejlichowski</i>	
Analysis of Four Polar Shape Descriptors Properties in an Exemplary Application	376
<i>Dariusz Frejlichowski</i>	
Single Frame Rate-Quantization Model for MPEG-4 AVC/H.264 Video Encoders	384
<i>Tomasz Grajek and Marek Domański</i>	
A Robust Method for Nose Detection under Various Conditions	392
<i>M. Hassaballah, Tomonori Kanazawa, Shinobu Ido, and Shun Ido</i>	
Segmentation of Moving Cells in Bright Field and Epi-Fluorescent Microscopic Image Sequences	401
<i>Marcin Iwanowski and Anna Korzyńska</i>	
Shape Representation and Shape Coefficients via Method of Hurwitz-Radon Matrices	411
<i>Dariusz Jakóbczak</i>	
Characteristics of Architectural Distortions in Mammograms - Extraction of Texture Orientation with Gabor Filters	420
<i>Magdalena Jasionowska, Artur Przelaskowski, and Rafał Józwiak</i>	
Author Index	431

A Constraint Satisfaction Framework with Bayesian Inference for Model-Based Object Recognition

Włodzimierz Kasprzak, Łukasz Czajka, and Artur Wilkowski

Institute of Control and Computation Eng., Warsaw University of Technology
W.Kasprzak@ia.pw.edu.pl
<http://www.ia.pw.edu.pl/>

Abstract. A general (application independent) framework for the recognition of partially hidden 3-D objects in images is presented. It views the model-to-image matching as a constraint satisfaction problem (CSP) supported by Bayesian net-based evaluation of partial variable assignments. A modified incremental search for CSP is designed that allows partial solutions and calls for stochastic inference in order to provide judgments of partial states. Hence the detection of partial occlusion of objects is handled consistently with Bayesian inference over evidence and hidden variables. A particular problem of passing different objects to a machine by a human hand is solved while applying the general framework. The conducted experiments deal with the recognition of three objects: a simple cube, a Rubik cube and a tea cup.

1 Introduction

The recognition of 3D objects in images has been studied for many decades [1], [2]. One aspect of these works is the design of computational frameworks for model-based object recognition, which reflect the structure and uncertainty of the data and provide appropriate tools for knowledge representation, learning and inference. Using a declarative language for model description in such frameworks allows large parts of the recognition system to be application-independent. Many different frameworks have been proposed, e.g. attributed graphs [3], semantic networks [4], relation structure grammars [5]. One of the main problems of such approaches is how to manage and evaluate partial model-to-data matches. It is known that uncertain and vague knowledge can be modelled efficiently in terms of probabilistic theory [6] or fuzzy logic [7]. In this paper we propose a framework that views the 3-D object recognition problem as a modified constraint satisfaction problem ([6], [8]), combined with the Bayesian approach to statistical inference [6], [9]. Both parts are dominantly of declarative nature and additionally there exists well-known ML learning procedures for Bayesian probability distributions [9].

In section 2 the particular problem of hand-held objects is presented and the segmentation stage of image analysis is explained. Our model-based search

and hypothesis judgment approach is explained in section 3. Then the approach is illustrated in section 4 as applied to the recognition of 2 different object shapes: boxes with texture and objects with elliptic boundaries.

2 The Problem

2.1 Objects in Human Hand

The goal of low-level image processing is to separate the human hand and the background from the other objects. In our system this is done based on color processing and moment-based contour filtering, as explained in the earlier paper [10]. First, the input image is subject to color processing in the YUV space. This allows to detect the human hand and (optionally) to focus on its neighborhood only. Morphological operations, edge and contour detection steps follow. Specific contour classes, like small "rectangular" chains, are labelled on base of features extracted by geometric moment functions. Next, the sufficiently long chains are approximated by linear segments. Consecutive segments are then approximated by arcs (fig. 1). Finally, a model-based detection of objects allows to assign model instances to segments groups detected in the image (fig. 1).

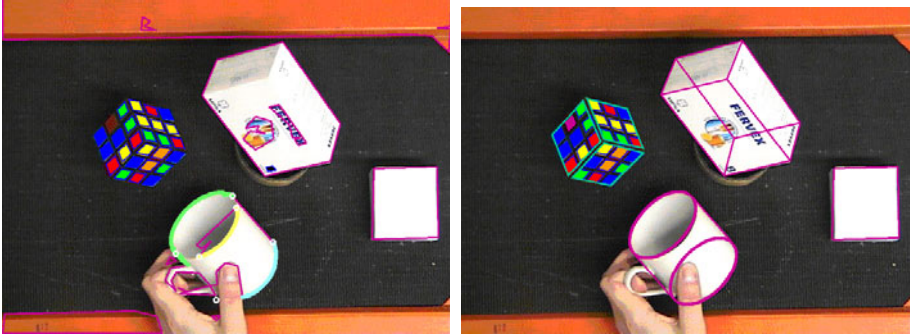


Fig. 1. After segment detection Fig.4 Object recognition results

2.2 Elliptic Arc Detection

Straight line segments are detected by a conventional method of edge chain approximation. Here a more detailed description of our approach to elliptic arc detection follows. This arc detection algorithm consists of 4 steps:

1. Detection of pairs of consecutive line segments, to be approximated by arcs (figures 2 and 2);
2. Extending existing arcs by including neighbor line segments (fig. 3) if they together can be approximated by an arc;
3. Connecting consecutive arcs together if they form a larger arc of the same type, i.e. convex or concave (e.g. ACBJ and JFE in fig. 3 are not connected);

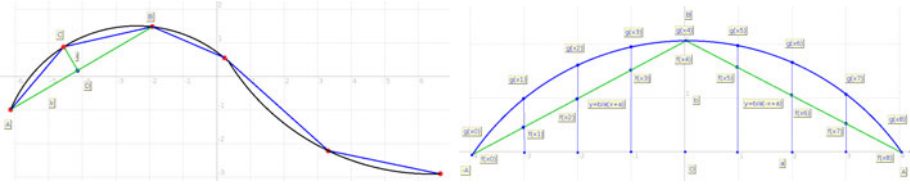


Fig. 2. The first stage of arc detection - arc ACB is found **Fig. 4.** Approximating two line segments by an arc

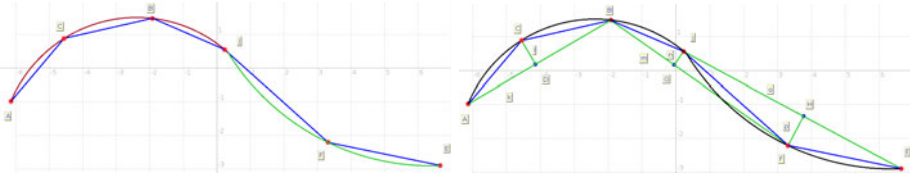


Fig. 3. Extending an arc by including a next line segment - arc ACB is extended to ACBJ **Fig. 6** There is no arc BJJ but there is a second arc JFE

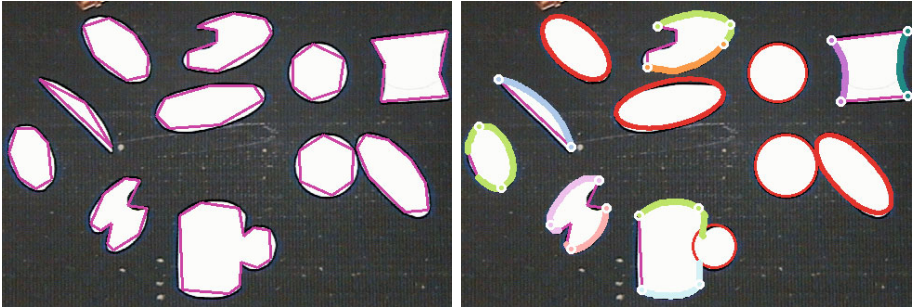


Fig. 4. Detected line segments that approximate contour chains **Fig. 8** Line segment chains approximated by arcs, circles and ellipses

4. Verifying the shape of an arc sequence. If the consecutive arcs form a closed contour then it is tried to approximate such contour by a circle or ellipse.

Examples of image segmentation results are provided in figures 4 (detection of line segments) and 4 (approximation of linear segment chains by arcs, circles and ellipses).

3 The Framework for Object Modelling and Recognition

3.1 CSP

The search space definition in a **discrete Constraint Satisfaction Problem** consists of following elements:

- A state set S , where a particular **state**, $\mathbf{s} = (d_1, d_2, \dots, d_n)$, is defined by assignments to its variables, $X = x_1, x_2, \dots, x_n$, where each x_i , ($i = 1, \dots, n$), can take values from a (finite) domain D_i .
- **Actions**, $a \in A$, mean transitions between states: $a_k : s_i \rightarrow s_j$.
- The **goal test** checks a set of constraints, $C(X)$, which induces allowed combinations of assignment values for subsets of state variables.
- A **solution state** is every state that satisfies the goal test, i.e. the sequence of actions is not relevant, but the final state only. The state $\mathbf{s} = (d_1, d_2, \dots, d_n)$, satisfies the goal test if: $C(d_1, d_2, \dots, d_n) = True$.

In particular, in our problem: the variables in X correspond to line segments of the object model, the values in D represent the current image segments and an action is assigning a value to some variable in given state. The variables and the set of constraints, $C(X)$, can be represented as a graph, $G(X, C(X))$ where nodes X represent variables and arcs $C(X)$ represent constraints between particular variables.

The structure of our CSP search is presented in table [1](#). While starting from an empty assignment the goal is to match (assign) eligible image segments (values) with model entities (variables). We introduced two **modifications** to the basic CSP search. The first modification is due to the definition of a **Bayesian network** for every problem. The subfunction **Score** calculates probability value of a partial solution, that consists of eligible assignments to variables. This score is due to stochastic inference process performed in a dedicated Bayesian net, created for current CSP problem. An empty assignment to a variable is also possible.

The basic algorithm for CSP is a depth-first tree search with a backtracking step when the path is not consistent with given constraints. The second modification of a typical CSP is that now partial paths can be potential solutions. The backtrack step is performed now when currently selected (extended) path does not satisfy the constraints of given problem or its score is lower than the score of predecessor path. In our view this is not a general failure but a situation where the previous state corresponds to a partial solution. The current path is conditionally stored as a possible partial solution when it is of higher score than the previous best one.

Still, if we succeed to find a complete path (with assignments for all variables) then we immediately stop the search with this final solution.

3.2 Bayesian Net

This is a simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions. Syntax:

- a set of nodes, one per variable;
- a directed, acyclic graph (link means that "direct influence");
- incoming links of given node represent a conditional distribution for this node given its parents, $P(X_i | Parents(X_i))$.

Table 1. Modified backtracking search for the framework of CSP and Bayesian net

function BacktrackingSearch(<i>csp</i>) returns <i>Solution</i>	
static <i>solution</i> = { } ;	
<i>path</i> = { }	
<i>solution</i> = RecursiveBacktracking(<i>solution</i> , <i>path</i> , <i>csp</i>)	
return <i>solution</i>	
function RecursiveBacktracking(<i>solution</i> , <i>path</i> , <i>csp</i>) returns <i>solution</i>	
IF	<i>path</i> is complete (Stop test)
THEN	return <i>solution</i>
<i>var</i> ← SelectUnassignedVariable(<i>csp.variables</i> , <i>path</i>)	
<i>valueList</i> ← GetDomainValues(<i>var</i> , <i>path</i> , <i>csp</i>)	
FOR	EACH <i>value</i> ∈ <i>valueList</i>
IF	(<i>path</i> ∪ { <i>var</i> ← <i>value</i> }) are consistent with <i>csp.constraints</i> AND Score(<i>path</i> ∪ { <i>var</i> ← <i>value</i> }) > Score(<i>path</i>)
THEN	add { <i>var</i> ← <i>value</i> } to <i>path</i> IF Score(<i>path</i>) > Score(<i>solution</i>) THEN <i>solution</i> = <i>path</i> <i>result</i> = RecursiveBacktracking(<i>solution</i> , <i>path</i> , <i>csp</i>)
	IF <i>result</i> ≠ <i>failure</i> THEN return <i>result</i>
	remove { <i>var</i> ← <i>value</i> } from <i>path</i>
return <i>failure</i>	

In the simplest discrete case, conditional distribution is represented as a conditional probability table (CPT), giving the distribution over X_i for each combination of parent values.

3.3 Example: A Cube Model

Let the hierarchic structure of a generic cube structure is given, i.e. the concept "cube" consists of 12 "edges" numbered as (0, 1, 2, ..., 11). In our framework this object has two other corresponding models. First, there is a "planar" graph of constraints (fig. 5), where line segments correspond to vertices and arcs to constraining predicates. For this particular object these constraints may be as follows: A = *line segments are connected*; B = *line segments are parallel*; C = *line segments are of similar length*. Second corresponding model is a Bayesian network that represents stochastic dependencies between the "high-level" concept "cube", intermediate-level concepts "views" and low-level "edges" (that can be observed in the image) (fig. 6).

The score of a partial state, in which some number of variables X_i have already been assigned image segments l_k , but not all of them, is obtained due to stochastic inference in Bayesian net, i.e. the computation of posterior probability of "cube" cause, given evidences. For example if segments are assigned to X_0 and X_2 then one computes the probability: $P(\text{cube} | X_0 = l_1, X_1 = l_2)$. This leads to a

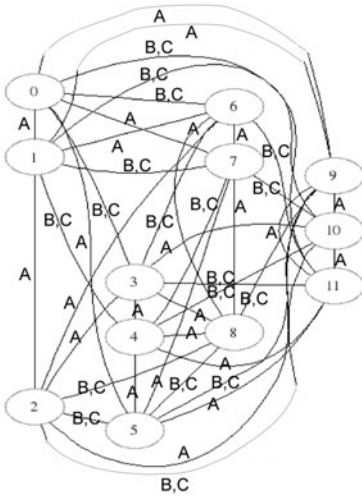


Fig. 5. Graph of constraints for a cube

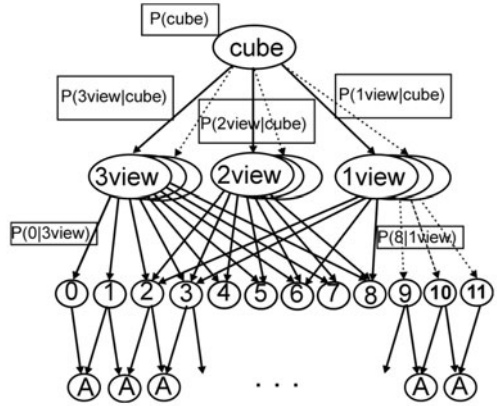


Fig. 6. Bayesian net for a cube

summation of pdf over all domain values for remaining (non-evidence) variables, X_2, \dots, X_l . Thus, scores of partial matches or a complete match, between image segments and model entities, are naturally obtained by the same evaluation method.

4 Results

4.1 A Rubik Cube Model

Although a Rubik cube seems to be a straightforward extension of a simple cube but it adds an important feature, a well-defined texture of its faces. Hence instead of a wire-frame model we need to create a surface model for it. First we detect image segments that are rectangular polygons filled by some color and assign them to the single-face model. Polygons satisfying face constraints are grouped into faces. Thus the graph of constraints (fig. 7) contains variables at "two abstraction levels", i.e. the CSP variables correspond to polygons or to faces. The polygon constraints have the following meaning: $A = two\ polygons\ are\ neighbors$, $B = polygons\ have\ parallel\ edges$, $C = polygons\ are\ not\ neighbors$, $D = polygons\ are\ of\ similar\ size$. The constraints between face variables are: $fA = faces\ are\ neighbors$, $fB = the\ edges\ of\ two\ faces\ are\ pair-wise\ parallel$.

The corresponding Bayesian net has a similar structure in the simple cube case, although now the "face" nodes does not share their "polygon" children (fig. 8). Some results of model-based detection of a Rubik cube are shown in fig. 9-12.

segment, $C =$ the segments are parallel, $D =$ the segments are of similar length. Some results of model-based detection of a tea cup are shown in fig. [13](#)[16](#).

5 Summary

An application-independent framework has been designed and tested for objects of several types that are passed to the robot by a human. It combines advantages of two modelling tools: an easy structure representation and efficient search methods of the CSP model, with well-defined learning and inference methods of a probabilistic Bayesian network model. We demonstrated practically how to handle in this framework wire-frame and textured objects of both linear and curved edges. The aim of future work is to extend the graph of constraints to handle many-level objects, with hierarchies like existing in a semantic network model, and to use continuous pdf in the Bayesian network.

Acknowledgments

The paper has been supported by the Polish Ministry of Science and Higher Education with the grant N N514 1287 33.

References

1. Besl, P., Jain, R.: Three-Dimensional Object Recognition. *Computing Surveys* 17(1), 75–145 (1985)
2. Faugeras, O.: Three-dimensional computer vision. A geometric viewpoint. The MIT Press, Cambridge (1993)
3. Tsai, W., Fu, K.: Subgraph error-correcting isomorphisms for syntactic pattern recognition. *IEEE Trans. SMC* 13, 48–62 (1983)
4. Niemann, H., Sagerer, G., Schroder, S., Kummert, F.: ERNEST: A semantic network system for pattern understanding. *IEEE Trans. PAMI* 12, 883–905 (1990)
5. Kasprzak, W.: A Linguistic Approach to 3-D Object Recognition. *Computers & Graphics* 11(4), 427–443 (1987)
6. Russel, S., Norvig, P.: *Artificial Intelligence. A modern approach*, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
7. Chan, K., Cheung, Y.: Fuzzy-attribute graph with application to chinese character recognition. *IEEE Trans. SMC* 22, 402–410 (1992)
8. Haralick, R., Shapiro, L.: The Consistent Labeling Problem, Part I. *IEEE Trans. PAMI* 1(2), 173–184 (1979)
9. Duda, R.O., Hart, P.E., Stork, D.: *Pattern Classification and Scene Analysis*, 2nd edn. J. Wiley, New York (2001)
10. Kasprzak, W., Szykiewicz, W., Czajka, L.: Rubik’s Cube Reconstruction from Single View for Service Robots. *Machine Graphics & Vision* 15(2/3), 451–460 (2006)

Visual Programming Environment Based on Hypergraph Representations

Peter Kapec

Slovak University of Technology, Bratislava, Slovakia
kapec@fiit.stuba.sk

Abstract. In this paper we present a concept of a visual programming environment that uses hypergraphs for representing, querying and visualizing software artifacts. The hypergraph representation allows to store semantic relations between software artifacts and can be visualized with well-known graph drawing algorithms. The proposed visual programming environment focuses not only on software visualization, but also offers context visualization for programming tasks. We present visualizations of an existing software system and demonstrate how contextual information can be displayed when browsing and modifying source code artifacts.

1 Introduction

Software, opposed to other engineering products, is intangible. This feature is one of the first mentioned to students of software engineering. In academic papers, we can find many notes about the intangibility of software: it has no form, no physical shape or size; software is an abstract and invisible collation of computer instructions; only the representation of software is what can be communicated between people and between people and computers. Therefore, software visualization aims to help us with the intangible software to make it “more” tangible. Software visualization is a large area and we can identify in this field two main sub-domains [13]: *algorithm visualization* and *program visualization*.

Algorithm visualization deals with software on higher-level than just code or data. The motivation is to find similarities/differences between executions of algorithms. Showing comparisons can be well used in pedagogy, which was demonstrated by the usage of systems like BALSAL [5].

Program visualization is oriented directly to the program implementation. We can classify visualizations according to the program aspect the visualization uses: code or data/data structures and how the visualization is done: static or dynamic. Program visualization is directed mainly to software development, where it tries to offer systems that provide effective development environments.

From the viewpoint of software engineering, it is interesting to see how these subfields interact in a development process. We can identify two main fields in which software visualization supports software engineering:

- *Software development:*

Here visual programming languages can be used – these languages use

visual notations to define a program and are not what most people understand under visual programming e.g. UML diagrams and IDEs for textual programming languages.

– *Understanding software:*

Here various visualizations that uncover hidden patterns or relations can be useful. Visualizations can focus on static program aspects like code structures, program data, code metrics etc., or on dynamic program aspects – displaying program execution and displaying internal program data changing during program execution. Of course input/output data visualization can be useful, however this belongs more to “standard” information visualization, because input/output data depend on specific area the system is used for (although similar information visualization techniques may be used also in software visualization).

Algorithm visualizations are often used to present programming concepts to programming beginners, but they might be used in both software understanding and software development. Of course, software engineering is more complex than just programming. The management of processes involved in software development cycles is a research itself.

In visual programming field two terms are often mentioned: *visual programming languages* (VPLs) and *visual programming environments* (VPEs). Difference between these two terms is that VPLs are an instrument for program specification and are defined with their syntax and semantics. VPEs are tools for creating, modifying and executing visual languages [2], but often VPLs and VPEs are implemented as one system. Visual programming languages can be classified into two groups:

- *pure visual languages*
 - they execute programs directly from visual representations without transformation to textual representation
- *hybrid visual/textual languages*
 - they convert visual representations into a textual form that is then compiled or interpreted, often existing textual programming language are used

In this paper we propose a hybrid visual/textual programming environment that is based on hypergraph representations of software artifacts and uses graph visualizations in 3D space. We do not try to develop a new VPL, rather than a visual environment that displays relations between software artifacts and allows to zoom in to perform programming tasks on source code level. The underlying hypergraph-based model is used for representation, querying and visualization thus eliminating intermediate steps and transformations during the visualization process.

Following sections are organized as follows. Section 2 introduces hypergraphs and discusses how they can be used to represent informations about software and how hypergraphs can be queried. Section 3 presents a visual programming environment that utilizes hypergraphs for visualizations. Section 4 mentions related work in software visualization, visual programming and knowledge representation areas. Final section concludes this paper with possible future work.

2 Hypergraphs

Common graphs in which an edge connects exactly two nodes is just a special case of a hypergraph. Hyperedges found in hypergraphs allow connecting more than two edges – hypergraphs are related to set theory as can be seen in definition [1](#).

Definition 1. Let $V = \{v_1, \dots, v_n\}$ be a finite set, whose members are called nodes. A hypergraph on V is pair $H = (V, \epsilon)$, where ϵ is a family $(E_i)_{i \in I}$ of subsets of V . The members of ϵ are called hyperedges.

With following definitions [2](#), [3](#) we can mathematically transform a hypergraph into a bipartite graph, using hypergraph's incidence matrix.

Definition 2. Let $H = (V, \epsilon)$ be a hypergraph with $m = |\epsilon|$ edges and $n = |V|$ nodes. The edge-node incidence matrix of H is:

$$M_H \in M_{m \times n}(\{0, 1\}) \quad (1)$$

and defined as:

$$m_{i,j} = \begin{cases} 1 & \text{if } v_j \in E_i \\ 0 & \text{else} \end{cases} \quad (2)$$

With hypergraph's edge-node incidence matrix we can construct a bipartite graph by following definition:

Definition 3. For a hypergraph $H = (V, \epsilon)$ with an incidence matrix M_H the bipartite incidence graph

$$B_H = (N_V \cup N_\epsilon, E) \quad (3)$$

is defined as follows:

$$\begin{aligned} E &= \{\{m_i, n_j\} : m_i \in N_\epsilon, n_j \in N_V, \text{ and } m_{i,j} = 1\} \\ N_\epsilon &= \{m_i : E_i \in \epsilon\} \\ N_V &= \{n_j : v_j \in V\} \end{aligned} \quad (4)$$

Using this transformations into bipartite graph we can directly utilize existing graph layout algorithms without the need to modify them and we do not need to develop new layout algorithms specially for hypergraphs.

To make hypergraphs more suitable for knowledge representation we can modify the edge-node incidence matrix in a way that it will contain not only 1 or 0 values. Replacing the 1 value with an incidence object we can formalize an alternative hypergraph definition as shown in following definition [4](#).

Definition 4. A hypergraph is a five-tuple $H = (V, \lambda_V, E, \lambda_E, I)$ where V, E, I are disjoint finite sets and we call V the vertex set of H , E the edge set of H , I the incidence set of H and $\lambda_V : V \rightarrow P(I)$ is a mapping that satisfies following conditions:

$$\forall v \neq v' \quad \lambda_V(v) \cap \lambda_V(v') = 0 \quad \cup_{v \in V} \lambda_V(v) = I \quad (5)$$

and $\lambda_E : E \rightarrow P(I)$ is mapping that satisfies the following conditions:

$$\forall e \neq e' \quad \lambda_E(e) \cap \lambda_E(e') = 0 \quad \cup_{e \in E} \lambda_E(e) = I \quad (6)$$

The added incidence object can be used to store additional information related to nodes and hyperedges it connects, but this information itself is not part of a node nor hyperedge. In incidence object we can store information about the direction and order of hyperedge tentacles, or e.g. the node's role in a hyperedge. Using definition [4](#) we can formalize popular knowledge representation formats [11](#).

2.1 Hypergraph Representations of Software Artifacts

Software consist not only from source code, but includes many related artifacts like documentations, data sets for testing, diagrams, revisions, user interfaces etc. These software artifact occur during the whole software development cycle, are created in various development tools and are stored in different file formats. Often one system is implemented in several programming languages mixing interpreted and compiled languages.

Many integrated development environments in use today offer great usability improvements and advanced project management features for developers. However the fragmentation into distinct files reduces observability of relations between artifacts.

As proposed in [15](#) hypergraphs can be used instead as an unifying structure to store all relevant software artifacts and their relations without borders imposed by separations into files. Hypergraph nodes can be used to represent software artifacts from source code level like variables, functions, classes, language constructs etc. Nodes can represent also documentations, revisions and various objects related to management like developer names, tasks, deadlines etc. Hyperedges are suitable to represent relations between software artifacts, e.g. we can represent relations on source code level like class-inheritance, call-graph, separation into modules etc. More important are however relations between not directly related artifacts e.g. how specification introduced entities in design and how they were implemented. Many different software artifacts and relations can be identified.

2.2 Querying Hypergraphs

Hypergraph representation of software artifacts can act as knowledge repositories that can be queried. We developed both a visual and textual query mechanism that use hypergraphs for querying, where query results are again hypergraphs. The query hypergraph can be displayed similarly to visualizations displayed in Figure 1, but for convenience textual notation can be used.

The query hypergraph contains conditions for nodes, incidences and edges. These conditions are searched in the queried hypergraph. The query language focuses on edges, because they represent important relations and can be written in following textual form: $E(I_1 : N_1, I_2 : N_2, \dots, I_n : N_n)$, where E is the pattern for hyperedge name, I_1 represents an incidence connected to E and to node N_1 , and similarly for other $I_i : N_i$ incidence-node pairs. These patterns can be regular expressions allowing complex searches. Queries for multiple relations can be chained with *and* and *or* operators. The following query illustrates how we can obtain all functions and their parameters from a software project.

```
is-instance-of(instance:*, type:function) and
has-parameters(function:*, parameter:*)}
```

With similar queries we can filter the large hypergraph visualization thus making the visualization more comprehensible.

3 Visual Programming Environment

We developed a visual programming environment that follows the well known information seeking mantra: *overview, zoom and filter, details on demand* [17].

The overview is provided in 3D space by graph visualizations that display software artifacts as nodes and relations between them. The Figure 1 illustrates a visualization of an existing software product. More than thousand nodes and four hundred hyperedges displayed were extracted after analyzing the source code and related documentation. Analyzes focused on not directly visible relations like call-graph, return values or documentation involved.

For the hypergraph layout we transform our hypergraph representation into a bipartite graph (see Section 2) and then use a modified graph layout algorithm based on force-directed placement [9]. Using a virtual camera the user can interactively explore the hypergraph visualization. Displaying large hypergraphs would be difficult to comprehend, thus the user can filter out only software artifacts of interest using hypergraph queries mentioned in Section 2.2.

For programming tasks it is necessary to have access to the textual source code. For these purposes every node can be selected and transformed into a floating window containing a text editor, see Figure 2. These text editor windows behave like billboards that are always parallel to the projection plane, thus are not distorted by perspective. These windows are also affected by the graph layout algorithm.

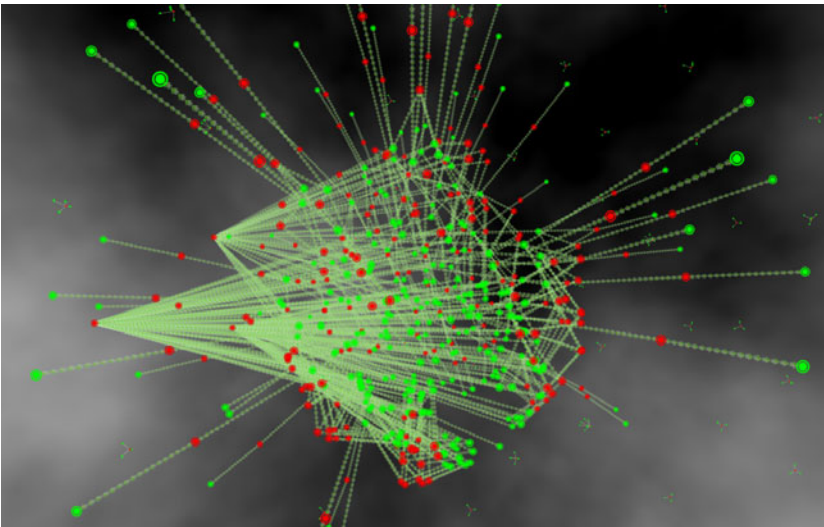


Fig. 1. Hypergraph overview visualization of an existing software system

From software artifacts displayed in these billboards links, which are actually hyperedges, to other nodes or billboards show different relations. In Figure 2 we can see in foreground a window that contains the source code of a function. The three other windows in background contain other functions that are called from the body of the function shown in foreground window. This relation is displayed as a hyperedge highlighted as red sphere; yellow spheres describe the caller/callee roles the functions play in this call relation.

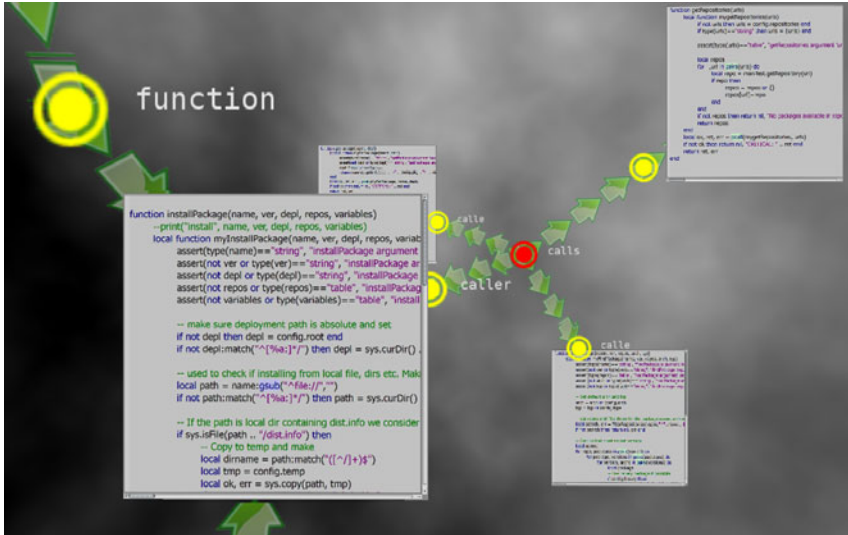


Fig. 2. Displaying software artifacts as windows containing textual editors

To display contextual information relevant to a software artifact displayed as node or window, users can use queries to get related artifacts. Found artifacts are at first displayed as nodes and are positioned with the layout algorithm near to the node/window of interest. Afterwards the user can switch these new nodes into windows to access source code fragments.

The goal is to augment this approach to automatically display contextual information around focused fragments based on relation relevancy in hypergraph representation. Displaying only hyperedges with relevancy above certain threshold will conserve screen space and reduce the complexity of the visualized graph.

4 Related Work

The presented visual programming environment was influenced by several research projects in areas including software and graph visualization, visual programming, user interfaces and knowledge representations.

A detailed overview of the software visualization field can be found in [6]. Software visualizations based on graph visualizations can help in software comprehension and searching tasks as shown in [11]. The initial view that shows the whole hypergraph representation of software artifacts is similar to [11]. Although we do not use similarity metrics to cluster graphs, similar effect can be achieved by hypergraph queries. Currently force-directed and energy-based graph visualization algorithms are popular, because they are relatively easy to program and produce aesthetically pleasing graph layouts [3].

Among many VPLs developed in past years several VPLs introduced interesting ideas. The Cube [12] was the first VPL that used 3D representations. Our proposed VPE utilizes 3D views but does not try to represent language constructs as pure visual objects, but rather than we use textual notations as used in Self VPE or Subtext [7] programming language. The TUBE [16] is a prototype-based hybrid visual language that uses Topic Maps as the underlying representation, which can be formalized with hypergraphs [1]. Opposed to TUBE we use directly the underlying hypergraph representations and we do not try to develop a new programming language, rather than our VPE works on top of existing textual languages. Although some VPLs are used in practice e.g. LabView, the usability of visual programming languages is still very questionable [10] and proper evaluations especially in practice have to be done.

Similar concepts of displaying 2D windows in 3D space have been widely researched [8], mostly as window/workspace managers in 3D virtual environments, but such approaches in VPEs are rare. Our VPE approach is similar to the Code Bubbles system [4]. The main difference is that our VPE uses 3D graph visualizations to provide overview and the underlying unifying hypergraph software artifact representation.

The hypergraph-based knowledge representation is based on the work by Auilans [1] and was also inspired by the Hypergraph Data Model (HDM) [14] and Hypergraph Query Language (HQL) [18]. However our hypergraph representation is simpler than HDM and our hypergraph queries return hypergraphs rather than tables produced by HQL.

5 Conclusions

The presented VPE is currently in early development phase. We are currently working towards a seamless integration of conventional textual programming with hypergraph representation so that dynamic changes in textual representations are automatically reflected in hypergraphs and vice versa. Future work will cover experiments with various graph-layout and interaction techniques, especially for cases when nodes do not contain textual data. Special attention will be dedicated to usability evaluation of presented VPE.

Acknowledgement. This work was supported by the grant KEGA 244-022STU-4/2010: Support for Parallel and Distributed Computing Education. We would like to thank master degree student Michal Paprčka who helped implementing 3D visualizations.

References

1. Auillans, E.A.: A formal model for Topic Maps. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 69–83. Springer, Heidelberg (2002)
2. Bardohl, R., Minas, M., Taentzer, G., Schürr, A.: Application of graph transformation to visual languages, pp. 105–180. World Scientific Publishing Co., Singapore (1999)
3. Battista, G.D.: Algorithms for Drawing Graphs: an Annotated Bibliography Computational Geometry. Theory and Applications 4, 235–282 (1994)
4. Bragdon, A., et al.: Code Bubbles: Rethinking the User Interface Paradigm of Integrated Development Environments. In: Proc. of ICSE'10, the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 455–464. ACM, New York (2010)
5. Brown, M.H., Sedgewick, R.: A system for algorithm animation. SIGGRAPH Comput. Graph. 18(3), 177–186 (1984)
6. Diehl, S.: Software Visualization - Visualizing the Structure, Behaviour and Evolution of Software. Springer, Heidelberg (2007)
7. Edwards, J.: Subtext: uncovering the simplicity of programming. In: Proc. of OOPSLA 2005, pp. 505–518. ACM, NY (2005)
8. Elmqvist, N.: 3Dwm: A Platform for Research and Development of Three-Dimensional User Interfaces Technical Report no.2003-04 (2003)
9. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Software - Practice & Experience* 21, 1129–1164 (1991)
10. Green, T.R.G., Petre, M.: When visual programs are harder to read than textual programs. In: Proceedings of ECCE-6, 6th European Conference on Cognitive Ergonomics, pp. 167–180 (1992)
11. Lewerentz, C., Simon, F.: Metrics-based 3d visualization of large object-oriented programs. In: VISSOFT 2002: Proceedings of the 1st International Workshop on Visualizing Software for Understanding and Analysis. IEEE Computer Society, Los Alamitos (2002)
12. Najork, M.A.: Programming in three dimensions. PhD thesis, Champaign, IL, USA (1994)
13. Price, B.A., Baecker, R.M., Small, I.S.: A principled taxonomy of software visualization. *Journal of Visual Languages & Computing* 4(3), 211–266 (1993)
14. Poulouvassili, A., McBrien, P.: A general formal framework for schema transformation. *Data and Knowledge Engineering* 28(1), 47–71 (1998)
15. Rauschmayer, A., Renner, R.: Knowledge-representation based software engineering. Technical Report 0407, Ludwig-Maximilians-Universität München, Institut für Informatik (2004)
16. Rauschmayer, A., Renner, P.: Tube: Interactive Model-Integrated Object-Oriented Programming. In: Proc. IASTED Int. Conf. Software Engineering and Applications, SEA (2004)
17. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th edn. Pearson Addison Wesley, London (2004)
18. Theodoratos, D.: Semantic integration and querying of heterogeneous data sources using a hypergraph data model. In: Eaglestone, B., North, S.C., Poulouvassilis, A. (eds.) BNCOD 2002. LNCS, vol. 2405, pp. 166–182. Springer, Heidelberg (2002)

MRI Brain Segmentation Using Cellular Automaton Approach

Rafał Henryk Kartaszyński and Paweł Mikołajczak

Department of Computer Science, Maria Curie Skłodowska University
Maria Curie-Skłodowska 1 square, 20 - 031 Lublin, Poland
hatamoto@op.pl, mikfiz@goblin.umcs.lublin.pl

Abstract. In this article new approach to the MRI brain segmentation is presented. It is based on the Cellular Automaton (CA). The method is truly interactive, and produces very good results comparable to those achieved by Random Walker or Graph Cut algorithms. It can be used for CT and MRI images, and is accurate for various types of tissues. Discussed here version of the algorithm is developed especially for the purpose of the MRI brain segmentation. This method is still in the phase of development and therefore can be improved, thus final version of the algorithm can differ from the one presented here. The method is extensible, allowing simple modification of the algorithm for a specific task. As we will also see it is very accurate for two-dimensional medical images. Three-dimensional cases require some unsophisticated data post processing [1], or making some modifications in the manner in which the automaton grows into the third dimension from the two-dimensional layer. To prove quality of this method, some tests results are presented at the end of this paper.

1 Introduction

The method is based on a cellular automata, firstly introduced by Ulam and von Neumann in 1966 [2]. It can be used to solve difficult segmentation problems, furthermore it is multilabel - segments many object simultaneously (computation time does not depend on the number of labels). It is interactive: requires the user to provide starting points for the algorithm (not many seeds are needed and their entering is not laborious), but in turn enables him to observe the segmentation process and make modifications in it. Interactivity is very important for physicians who like to have some (often large) influence on medical images processing. Furthermore, a radiologist will be able to place seed points very accurately and in characteristic places of a specific organ (the reason why will be explained later), and will check the correctness of segmentation afterwards.

2 Cellular Automaton

A cellular automaton is a discrete model studied in the computability theory, mathematics, and theoretical biology [3]. It consists of an infinite, regular grid

of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the state of a cell at time t is a function of the states of a finite number of cells (neighborhood) at time $t - 1$. These neighbors are a selection of cells relative to the specified cell and do not change. Though the cell itself may be in its neighborhood, it is not usually considered a neighbor. Every cell has the same rule for updating, based on the values in this neighborhood. Each time the rules are applied to the whole grid a new generation is produced.

Formally, cellular automata is a triplet

$$A = (S, N, \delta); \quad (1)$$

where

S - not empty states set

N - neighboring system

δ - transition function describes way of calculation cell state in time $t + 1$ basing on state of its neighbors in time t .

State of cell p , on the other hand also consists of three values:

$$S_p = (l_p, \Theta_p, I_p) \quad (2)$$

where:

l_p - current cell label

Θ_p - cell strength, being real number and we may assume that $\Theta_p \in [0, 1]$

I_p - is intensity (value) of pixel (or voxel) in image corresponding to cell p .

3 Application of Cellular Automaton for Segmentation

In our case cellular automaton is a sheet of graph paper (for two dimensional images), where each square is a cell, and each cell corresponds to a pixel of image being segmented. In three dimensional cases we would have a set of two dimensional sheets placed one on another. Because this segmentation algorithm is multi-label each cell state consists of one of the labels of areas we are segmenting plus neutral territory label. During the evolution of an automaton other cells slowly conquer neutral territory. Obviously, each cell has eight neighbors on the same plane, and if we are dealing with a three dimensional case, there are also eighteen ones on the planes above and beneath. In a more general instance we may use, for example, von Neumann's:

$$N(p) = q \in Z^n : \|p - q\|_1 := \sum_{i=1}^n |p_i - q_i| = 1 \quad (3)$$

or Moor's

$$N(p) = q \in Z^n : \|p - q\|_\infty := \max_{i=1..n} |p_i - q_i| = 1 \quad (4)$$

neighboring system.

The algorithm requires the user to provide starting points (seeds) for segmentation. In the simplest case two kinds of seeds should be given, i.e. seeds corresponding to the object we are segmenting and corresponding to its surroundings (see Fig. 2.a, 2.c). Of course, more than two classes can be provided, thus segmentation of several objects will be done.

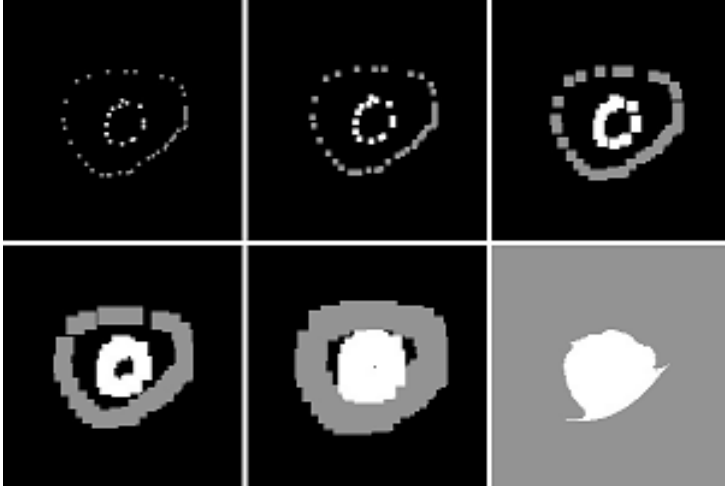


Fig. 1. Evolution of automaton segmenting tumor from Fig. 2.c: black - neutral territory, white - organ labeled bacteria, grey - background labeled bacteria. Evolution time steps are as follows (from upper left): 1, 3, 6, 11, 20, 40 (lower right).

Let us use a biological metaphor to supply intuitive explanation of automata evolution. An image (two or three dimensional) is a discrete universe (set of pixels / voxels) which has to be labeled in order to extract its part. Adding starting seed points is equal to the creation of new bacteria's colonies matching image parts, pixels not marked as seeds are considered as neutral territory (see Fig 1. where black pixels are neutral territory, white ones belong to ROI's cells and grey to cells surrounding ROI). The labeling process can be treated as a struggle for domination of different types of bacteria, grouped in several colonies (seeds). Because time in this automaton is discrete, in each time step every bacterium tries to occupy the neighboring pixel not belonging to its colony. After every step each colony grows, and soon every pixel of the image belongs to one of them. The evolution process is finished when, in one of the steps, no new cells have been conquered.

The stop condition, for the evolution process, can be a case when during one time step no new cells are being conquered and the cell state ceases to change. Unfortunately this approach may lead to an execution of a lot of time steps, some of which are completely unnecessary. Because in most cases we wish to segment one organ which is a small part of the image, the better way is to narrow the calculations to a small box containing the interesting part of the image, thus shortening the evolution process. Further improvement can be made

by consideration of changes (to be more precise: lack of state changes) close to the object boundaries. We put seed points (the ones belonging to the object and the ones belonging to its surroundings) near the boundary of the ROI (on both the inner and outer side), so during the evolution, the situation on the ROI boundary is quickly stabilized and the process can be stopped, and the interior of the ROI (if not yet conquered by cells corresponding to it) can be automatically filled. Another way to save time is to execute a fixed number of time steps. How many? It depends on the type of segmentation we perform (number of seed points, distance between outermost points, etc.), and can be estimated empirically. For example, results presented in this article (for two-dimensional cases) were achieved with forty time steps (no noticeable differences have been found between forty and, for example, one hundred steps).

4 Results of the Two-Dimensional Segmentation

We will now present some results of the cellular automata segmentation (see Fig. 2). As it can be seen it is a very accurate method allowing to segment different tissues from various types of medical images. It should be stressed that accuracy of segmentation depends strongly on the appropriate choice of seed points.

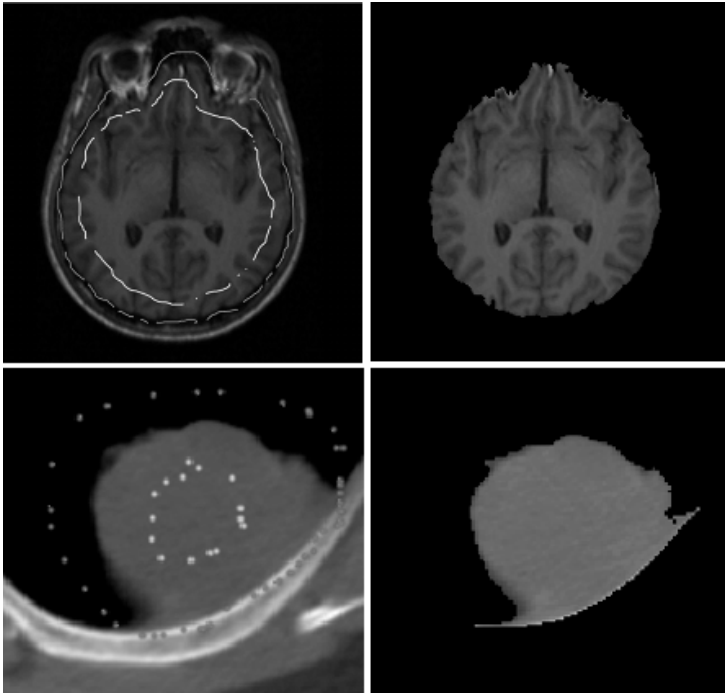


Fig. 2. (a) Seeds for MRI brain segmentation (top left); (b) Segmented MRI brain (top right); (c) Seeds for lung tumor segmentation (lower left); (d) Segmented lung tumor (lower right)

5 Results of the Three-Dimensional Segmentation

When we deal with three-dimensional data sets we would like to set seed points for one layer and let the automata segment the rest of the organ. The presented algorithm can be easily applied to two and three dimensions. If seeds for only one layer are given, 3D segmentation turns out to be rather accurate, though sometimes nearby tissues are recognized as a part of the segmented organ (see Fig. 3). This problem can be easily fixed by post processing of the data set. Applying morphological operations [4]: dilation and erosion filtering, destroys small connections between the main organ and the over-segmented tissue. Next connected components labeling [5] [6] is performed to select the only organ of interest to us. As we can see in Fig. 4 such a processing is very effective. Apart from post processing of the data set, some pre processing (morphological

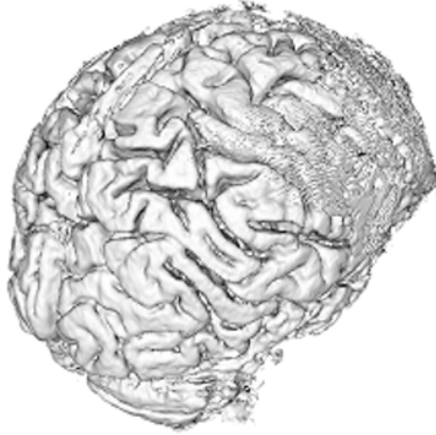


Fig. 3. MRI Brain segmented by pure CA

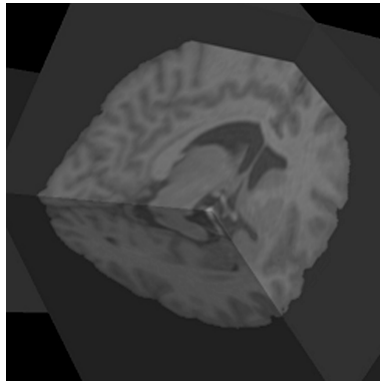


Fig. 4. MRI Brain segmented by CA and post-processed

operations) can also be necessary (thresholding at the Otsu level). The main difficulty may be finding a proper place to put the seed points at.

6 Observations

Let us firstly discuss the correct introduction of seed points. In the case of 2D this is rather simple and does not need additional explanation. One should remember to mark (as seeds) all (most of) characteristic pixels of the object and its surroundings - background. When this is done properly we may be sure that each point will grow in proper direction and the boundary will be correctly found. In 3D the same rules apply, but some further guidelines must be given. When choosing layers in the data set to place seeds in, we should choose ones which have the most characteristic features of the tissue we are segmenting and its surroundings. Seeds of the background ought to be placed not only near the boundary of the object, but also on different tissues surrounding it (even far away from the object). We must remember that we are dealing with a three-dimensional data set and tissues (organs) distant from the object on the current slice, can be in touch a few centimeters above. This fact should also be taken into consideration when selecting a place for seed points.

As we could see, on the effects of the two-dimensional segmentation, sometimes the segmented boundary is ragged (see Fig.2.c). When we need to capture the smallest detail, this is acceptable in most cases, but it may also be an unwanted artefact. To achieve smoother boundaries a slight alteration of the transition rule can be put forward. Let us call the cells of a different label than the examined one, the enemies of that cell. Now, cells having more enemies than, lets say, E1 are prohibited from attacking their neighbors, and cells that have more than E2 enemies are automatically conquered by the weakest of their neighbors. Values E1, E2 control boundary smoothness and should have a value from 6 to 9 (no smoothing) for the Moor neighboring system. This modification has not been tested for three-dimensional cases.

7 Tests and Conclusions

To objectively estimate quality of the cellular automaton segmentation, we have compared its results with segmentation done by an expert - physician. The test data was obtained from The Internet Brain Segmentation Repository (IBSR) [7]. This data consists of MRI brain scans of several patients. For each scan there is a second one with brain segmented by radiologist. Testing method is, therefore obvious. We have segmented brain from several dataset using the cellular automaton approach and compared them with model results by a simple image subtraction (see Fig. 5 and 6). Next, we have compared the number of over- and under-segmented pixels/voxels (real quantitative difference) with the exemplary pixels/voxels number. For a few dozens of two dimensional images we have obtained average difference of 3,3% (both over- and under-segmentation). For three dimensional datasets the difference ratio was slightly greater and was equal 4,7%. As we can see the dissimilarity is very small, and often not noticeable for a "naked eye".

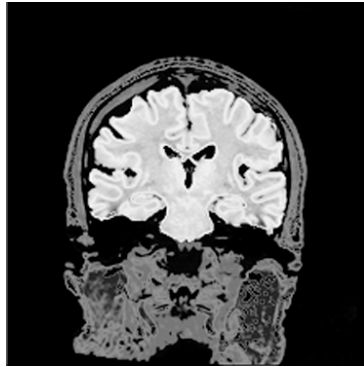


Fig. 5. Segmented brain (MRI)



Fig. 6. Comparison of result acquired by CA segmentation with one done by an expert (bright pixels mark over-segmentation - 434 pixels, darker under-segmentation - 155 pixels); relative difference to the model is 2,4%

To summarize, advantages of the presented algorithm are: accuracy (results comparable with ones provided by human expert), interactivity, possibility of segmenting multiple object at once, scalability. Its drawback is a long execution time, but as we could see this can be leveled by making some simple modification. In the three-dimensional case post processing is required to erase artifacts which could sometimes appear during segmentation. Nonetheless, the method itself is very promising and should be developed further to improve its performance and find appropriate modifications for specific purposes.

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Publishing Company, Reading (1992)
2. Von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana (1966), Ed. Completed by A. Burks

3. Hernandez, G., Herrmann, H.J.: Cellular automata for elementary image enhancement. *CVGIP: Graphical Model and Image Processing* 58(1), 82–89 (1996)
4. Serra, J.: *Image Analysis and Mathematical Morphology*. Academic Press, London (1982)
5. Horn, B.K.P.: *Robot Vision*. MIT Press, Cambridge (1986)
6. Heimann, T., Thorn, M., Kunert, T., Meinzer, H.-P.: New methods for leak detection and contour correction in seeded region growing segmentation. In: 20th ISPRS Congress, Istanbul 2004, *International Archives of Photogrammetry and Remote Sensing*, vol. XXXV, pp. 317–322 (2004)
7. <http://www.cma.mgh.harvard.edu/ibsr/>

Improved Context-Based Adaptive Binary Arithmetic Coding in MPEG-4 AVC/H.264 Video Codec

Damian Karwowski and Marek Domański

Poznań University of Technology, 60-965 Poznań, Poland
dkarwow@multimedia.edu.pl, domanski@et.put.poznan.pl

Abstract. An improved Context-based Adaptive Binary Arithmetic Coding (CABAC) is presented for application in high definition video coding. In comparison to standard CABAC, the improved CABAC works with proposed more sophisticated mechanism of data statistics estimation that is based on the Context-Tree Weighting (CTW) method. Compression performance of the improved CABAC was tested and confronted with efficiency of the state-of-the-art CABAC algorithm used in MPEG-4 AVC. Obtained results revealed that 1.5%-8% coding efficiency gain is possible after application of the improved CABAC in MPEG-4 AVC High Profile.

1 Introduction

An important part of video encoder is the entropy encoder that is used for removing correlation that exists within data. Numerous techniques of entropy coding were elaborated, many of them have been applied in video compression. The state-of-the-art entropy coding technique used in video compression is the Context-based Adaptive Binary Arithmetic Coding (CABAC) [3,5,6,14] that is used in the newest Advanced Video Coding (AVC) standard (ISO MPEG-4 AVC and ITU-T Rec. H.264) [1,5,6]. In comparison to other entropy coders used in video compression, CABAC uses efficient arithmetic coding and far more sophisticated mechanisms of data statistics modeling which are extremely important from the compression performance point of view. The goal of the paper is to show that it is possible to reasonably increase compression performance of adaptive entropy encoder by using more accurate techniques of conditional probabilities estimation. The paper presents the improved version of CABAC with more exact technique of data statistics modeling that is based on the well known from the literature Context-Tree Weighting (CTW) algorithm [7,8,9,14,15]. The older versions of improved CABAC codec have been already presented by authors in [10,11,12,13]. Nevertheless, the older versions of improved CABAC were working only with 4x4 transform for residual coding. In the High Profile of MPEG-4 AVC the 8x8 transform is additionally used. When using 8x8 transform, independent set of statistical models is used in standard CABAC. The 8x8 transform mode was not supported by the previous versions of improved CABAC. Coding tools of

High Profile significantly improve compression capability of encoder (especially for high definition (HD) video). This paper makes the continuation of previous research and presents the newest version of improved CABAC codec that works with 4x4 and 8x8 transform. In this paper, the improved CABAC was tested in application to compression of high definition video. The paper presents new not published yet results on efficiency of improved CABAC for 8x8 transform mode.

2 CABAC Algorithm

Detailed description of CABAC can be found in [13,6,14]. Only short review of CABAC is presented here to show some features of algorithm that are important from the point of view of this paper. In general, CABAC consists of three blocks: binarizer, context modeler and arithmetic encoder core (see Fig. 1).

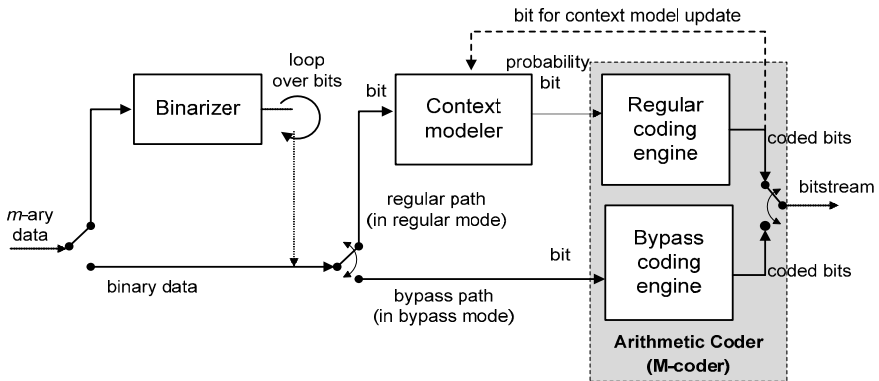


Fig. 1. Block diagram of standard CABAC encoder

In order to speed up processing of symbols, binary arithmetic codec (the so-called *M*-codec) was used in CABAC [13,4,5,6]. Due to application of binary arithmetic codec, all non-binary valued syntax elements must be mapped into string of binary symbols. This is realized by the binarizer. The binarizer has a huge impact on the number of bits that are put to arithmetic codec core. Therefore, CABAC exploits adaptive binarization by the application of several different binarization schemes for coded syntax elements. Results of binarization are put to arithmetic encoder core. The arithmetic encoder core allows to efficiently representing binary symbols by exploiting statistics of coded data estimated by the context modeler. The way of data statistics estimation to a large extent determines compression performance of adaptive arithmetic encoder. Therefore, in order to obtain an accurate adaptation to the current signal statistics, a total of 460 finite-state machines (FSM) was defined in CABAC for all syntax elements. Individual FSM is used to calculate the conditional probability of a binary symbol that appeared in a given context. By encoding a binary symbol, CABAC

chooses the right FSM depending on type of syntax element and its values in two neighboring blocks. That is the context encoding and independent statistics estimation for individual syntax elements make the power of CABAC algorithm. The algorithm of data statistics estimation used in CABAC belongs to the most advanced used in adaptive entropy encoders.

3 Research Problem

CABAC algorithm realizes advanced data statistics modeling, nevertheless some simplifications were done in the mechanism of data statistics estimation. First of all, only one transition rule was defined for 460 FSMs that calculates the conditional probabilities of symbols. Secondly, the limited set of only 128 pre-defined quantized values of probabilities is used by arithmetic codec core. Both simplifications of CABAC negatively affect its compression performance. An interesting research problem is to check the possibility of improving compression performance of CABAC by the use of even more exact method of data statistics estimation. This makes the subject of further part of the paper.

4 Improved CABAC Algorithm

Authors proposed the original improvement of CABAC algorithm that increases coding efficiency of entropy codec. Modifications concern the context modeler block only. Other parts of CABAC algorithm (binarization and arithmetic codec core) were left unchanged (see Fig. 2).

In the improved CABAC, the standard method of data statistics modeling was replaced with more sophisticated mechanism of the conditional probabilities estimation that exploits the CTW method. To do this, the 460 FSMs of CABAC were replaced with new 460 more accurate statistical models, each statistical model uses binary context tree of depth D (see Fig. 3). The binary context tree

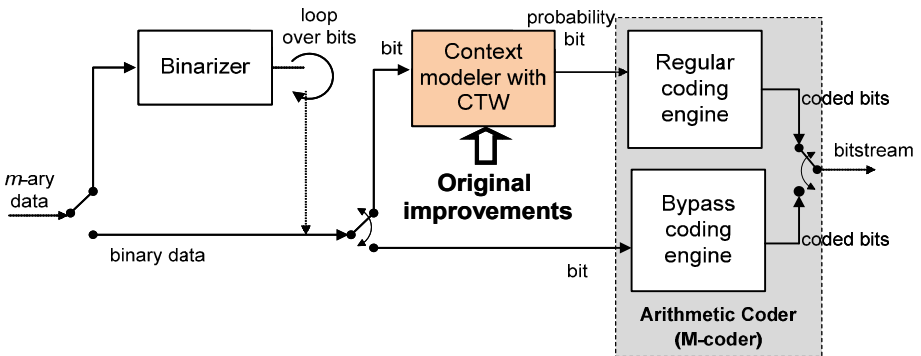


Fig. 2. Block diagram of improved CABAC encoder

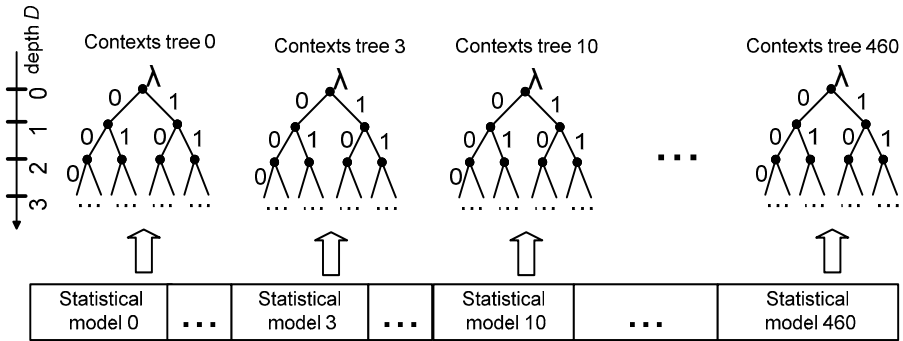


Fig. 3. Statistical models in improved CABAC

is a collection of nodes, each node keeps the information about the statistics of 0 and 1 symbols that appeared in a given context. It is difficult to predict which context will be the best for encoding the current symbol. Therefore, the CTW method appropriately weights the statistics collected in individual nodes of context tree to produce the weighted conditional probability for symbol. More details can be found in [10].

In the proposed mechanism of data statistics gathering the depth D of context trees strongly influences the precision of estimated conditional probabilities. In the previous works [10, 11, 14], authors tested the influence of this parameter on efficiency of entropy encoder. The improved CABAC was tested for depths D equal to 2, 4, 8, and 12. Obtained results proved, that the greater depth D of context trees the more accurate estimation of probabilities and better entropy encoder efficiency. The final conclusion of the research was that depth $D=8$ is a good compromise between the efficiency and complexity of entropy codec. In the newest version of improved CABAC depth $D=8$ is used for context trees. In comparison to the previous versions of improved CABAC, the current version was developed to work with both the 4x4 and the 8x8 transform mode.

5 Methodology of Experiments

The proposed improved mechanism of data statistics estimation was implemented and embedded into the structure of CABAC entropy codec. The modified CABAC algorithm was successfully activated in the reference software JM 10.2 [2] of MPEG-4 AVC video codec. It must be emphasized that both video encoder and video decoder were prepared to obtain reliable experimental results. This phase of works was very difficult and time-consuming for the reason of JM 10.2 reference software implementation complexity (about 90 thousands lines of program code in C). The compression performance of the modified MPEG-4 AVC encoder was tested and confronted with coding efficiency of the original MPEG-4 AVC video encoder. This paper focuses on efficiency of the improved MPEG-4

AVC working in High Profile, which distinguishes the paper from previous works [10,11,12,13]. Experiments were done according to the following scenario:

- Full HD test video sequences were used: *Pedestrian Area*, *River Bed*, *Rush Hour* and *Station* (1920x1080, 25 fps);
- Sequences were encoded with JM 10.2 encoder forcing different values of bitrate at the output of encoder. Different bitrates were obtained by setting different values of quantization parameter QP in encoder (from $QP=20$ to $QP=44$ with step 6) that is responsible for the quality of decoded video sequence. Experiments were done for I -, P - and B -frames (GOP: *IBBPBBP...*);
- The improved CABAC codec was tested for 8x8 transform mode;
- Encoding tools were used in both the modified and the original MPEG-4 AVC encoder that guarantee identical quality of reconstructed sequences regardless of entropy encoder applied;
- Correctness of encoded bitstreams was checked in each experiment.

6 Experimental Results

Obtained results revealed that the modified High Profile MPEG-4 AVC encoder with improved CABAC is characterized by higher compression performance relative to the original MPEG-4 AVC. In the first step of research, experiments were done for I , P , and B frame types. Detailed experimental results for this scenario were presented in Fig. 4. Results were averaged over 4 test sequences.

The improved CABAC outperforms the original CABAC for all frame types (I , P and B). Nevertheless, different values of bitrate reduction were observed for I -, P -, and B -frames. Additionally, different results were obtained for individual QP values that determine the size of bitstream at the output of video encoder. Greater bitrate reduction was observed for smaller bitrates (bigger values of QP parameter). Best results were noticed for B -frames (2.5%-7.5% reduction

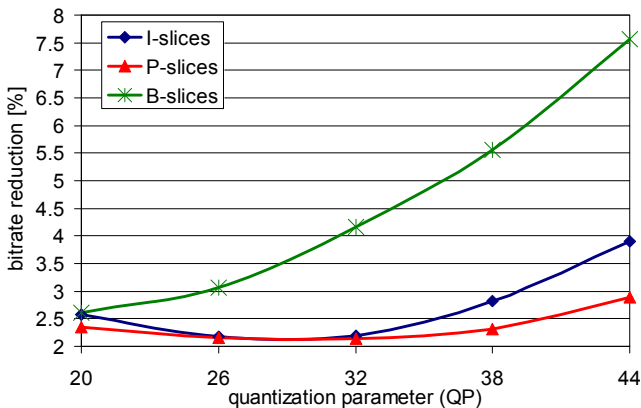


Fig. 4. Average bitrate reduction due to application of improved MPEG-4 AVC encoder instead of the original MPEG-4 AVC (for I -, P -, and B -frames, 8x8 transform mode)

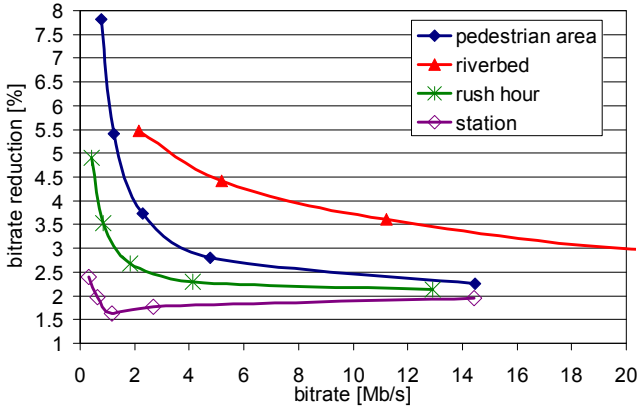


Fig. 5. Average bitrate reduction due to application of improved MPEG-4 AVC encoder instead of the original MPEG-4 AVC (for individual test video sequences, 8x8 transform mode)

of bitrate), the worst results were observed for *P*-frames (only 2.5%-3% bitrate reduction). In the case of *I*-frames 2.5%-4% bitrate reduction was observed.

In the second step of research, the efficiency of improved CABAC was tested for whole test sequences. Detailed experimental results are depicted in Fig. 5.

After the application of improved CABAC within MPEG-4 AVC bitrate reduction of 1.5%-8% was noticed in comparison to standard MPEG-4 AVC encoder. Different results were obtained for individual test video sequences. Best results were obtained for *Pedestrian Area* (2.5%-8%), the smaller bitrate reduction was observed for *Station* sequence (1.5%-2.5%). In the case of each video sequence, better results were achieved for bigger values of *QP* parameter (which corresponds to smaller bitrates).

Experimental results revealed that gap in compression performance between improved and original CABAC strongly depends on: value of bitrate, content of sequence and frame type. The three mentioned above parameters affect statistical properties of residual signal that is finally encoded with entropy encoder. The improved CABAC algorithm is able to track changing statistics of coded data. In the original CABAC codec, context modeler takes advantage of predefined probability distribution of data which reduces the ability of entropy codec to track data statistics efficiently. When the real statistics of data differs from the assumed one, coding efficiency of standard CABAC decreases. It results in greater bitrate reduction when using the improved CABAC.

Higher coding efficiency of improved CABAC was obtained at the cost of computational complexity increase. The application of improved CABAC in the MPEG-4 AVC reference decoder increased total decoding time up to 25% (for the useful in high definition television range of bitrates below 8 Mb/s). The authors are fully aware that this result may be different for another implementation of MPEG-4 AVC.

7 Conclusions

The improved adaptive arithmetic codec CABAC was presented in application to high performance video coding. The improved CABAC exploits authors' more accurate mechanism of the conditional probabilities estimation. Experiments revealed superior coding efficiency of improved CABAC relative to the state-of-the-art CABAC algorithm. When comparing to the original CABAC encoder, improved CABAC allows for 1.5%-8% bitrate reduction when applied in MPEG-4 AVC High Profile. The improved CABAC codec is more complex than the original CABAC. In comparison to original MPEG-4 AVC, improved MPEG-4 AVC video decoder is up to 25% more computationally complex for high definition sequences.

Acknowledgment

The work was supported by the public funds as a research project.

References

1. ISO/IEC 14496-10, Generic Coding of Audio-Visual Objects, Part 10: Advanced Video Coding (March 2006)
2. H.264/AVC software coordination site, <http://bs.hhi.de/~suehring/tml>
3. Marpe, D., Schwarz, H., Wiegand, T.: Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13(7), 620–636 (2003)
4. Marpe, D., Marten, G., Cycon, H.L.: A Fast Renormalization Technique for H.264/MPEG4-AVC Arithmetic Coding. In: 51st Internationales Wissenschaftliches Kolloquium Technische Universität Ilmenau (September 2006)
5. Special issue on H.264/AVC video coding standard. *IEEE Transactions on Circuit and Systems for Video Technology* 13 (July 2003)
6. Richardson, I.E.G.: H.264 and MPEG-4 Video Compression. In: *Video Coding for Next-generation Multimedia*. Wiley, Chichester (2003)
7. Willems, F.M.J., Shtarkov, Y.M., Tjalkens, Tj.J.: The Context-Tree Weighting Method: Basic Properties. *IEEE Transactions on Information Theory* 41(3), 653–664 (1995)
8. Willems, F.M.J.: The Context-Tree Weighting Method: Extensions. *IEEE Transactions on Information Theory* 44(2), 792–797 (1998)
9. Willems, F.M.J., Tjalkens, Tj. J.: Complexity Reduction of the Context-Tree Weighting Algorithm: A study for KPN research. Technical Report EIDMA-RS.97.01, Euler Institute for Discrete Mathematics and its Applications, Eindhoven University of Technology (1997)
10. Karwowski, D.: Improved Arithmetic Coding in H.264/AVC Using Context-Tree Weighting and Prediction by Partial Matching. In: *European Signal Processing Conf. EUSIPCO 2007*, Poznań, Poland, pp. 1270–1274 (September 2007)
11. Karwowski, D., Domański, M.: Improved Arithmetic Coding in H.264/AVC Using Context-Tree Weighting Method. In: *Picture Coding Symposium PCS 2007*, Lisboa, Portugal (November 2007)

12. Karwowski, D., Domański, M.: Improved Context-Adaptive Arithmetic Coding in H.264/AVC. In: European Signal Processing Conference EUSIPCO 2009, Glasgow, Scotland (August 2009)
13. Karwowski, D.: Ulepszone Adaptacyjne Kodowanie Arytmetyczne w Standardzie H.264/AVC. In: Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2009, Czerwiec 2009, Warszawa (2009)
14. Karwowski, D.: Advanced Adaptation Algorithms of Arithmetic Coding in Hybrid Video Compression. Doctoral Dissertation, Poznań University of Technology (2008)
15. Salomon, D.: Data Compression. The Complete Reference, 4th edn. Springer, Heidelberg (2006)

Local Polynomial Approximation for Unsupervised Segmentation of Endoscopic Images

Artur Klepaczko¹, Piotr Szczypiński¹, Piotr Daniel², and Marek Pazurek²

¹ Technical University of Łódź, Institute of Electronics
90-924 Łódź, ul. Wólczajska 211/215
aklepaczko@p.lodz.pl

² Medical University of Łódź, Department of Digestive Tract Disease
90-153 Łódź, ul. Kopcińskiego 22

Abstract. In this paper we present a novel technique for unsupervised texture segmentation of wireless capsule endoscopic images of the human gastrointestinal tract. Our approach integrates local polynomial approximation algorithm with the well-founded methods of color texture analysis and clustering (k-means) leading to a robust segmentation procedure which produces fine-grained segments well matched to the image contents.

1 Introduction

In this paper we present a novel technique for unsupervised texture segmentation applied to wireless capsule endoscopic (WCE) 2D images of the human gastrointestinal tract. We are primarily focused on improving automatic detection of ulcerations and lesions visible in the internal lumen of the small intestine and their precise delineation from normal or irrelevant regions. Our approach integrates local polynomial approximation (LPA) [1] algorithm with the well-founded color texture analysis and unsupervised classification (k-means) methods. As a first step, LPA performs pixel-wise analysis of the circular view given by a WCE camera (cf. Fig. 1a) and for each pixel it defines a corresponding region of interest (ROI) whose size and shape is adapted to this pixel local neighborhood. Then, using histogram information calculated separately for 8 different color channels, each ROI (and thus also its associated pixel) is described by a vector of color texture features. Eventually, these vectors are classified in the unsupervised manner by the k-means algorithm with the number of clusters set a priori. This combination of methods leads to a robust, three-phase fully automated segmentation procedure which produces fine-grained segments well matched to the image contents.

The motivation for our research stems from the persistent need for automation of WCE video interpretation process. It traditionally involves much effort from a human expert, is a monotonous and time-consuming task, requiring high level of concentration. Apparently, in the literature the problem of automatic segmentation of WCE images has attracted little attention. Majority of studies

(e.g. [2,3,4,5]) concentrate on WCE video sequence segmentation as a whole — the aim is to identify different sections of the gastrointestinal system. It is worth noticing that combined color and texture information is widely exploited in these works showing its potential in application to endoscopic image processing.

Another category of approaches utilize color and texture features to detect certain structures (like polyps, bleeding, ulcers and lesions) with or without performing image segmentation. In [6] WCE frames are first submitted to the smoothing procedure and then segmented by the region growing algorithm. Identified regions and their mean RGB color values are used to train a neural network classifier so that it recognizes a range of abnormal patterns in new images. Similar studies — although they involve different classifiers, color and texture models — are described in [7,8] and [9,10]. However, in these approaches images are divided uniformly into a predefined number of square or circular ROIs, disregarding particular shapes visible in a given frame. The focus is put on determination of characteristics of chosen patterns in endoscopic images. These characteristics can be used to build classifiers capable of distinguishing classes of ideally shaped image snippets. Although such classifiers may to some extent help indicate diagnostically valuable WCE frames, precise delineation, visualization and quantification of interesting regions remains an unsolved task.

In the following we present details of our approach to WCE image segmentation. Section 2 provides concise review of the materials and methods used in this research. In Sect. 3 we present preliminary results of segmentation obtained for sample WCE images taken from several different examinations. We compare these results with segmentation performed manually by two independent experts in WCE-based diagnosis. High level of correspondence between both types of segmentation techniques (manual and the automatic one) makes us believe that the proposed method can be successfully introduced to the clinical practice. Conclusions and possible improvements are outlined in Sect. 4.

2 Materials and Methods

2.1 Wireless Capsule Endoscopy

Wireless capsule endoscopy (WCE) [11,12], is a technique that facilitates the imaging of the human gastrointestinal system including small intestine. The WCE system consists of a pill-shaped capsule (cf. Fig. 1a) with built-in video camera, light-emitting diodes, video signal transmitter and battery, as well as a video signal receiver-recorder device. The wireless capsule endoscope used in this study produces color images of the internal lumen (cf. Fig. 1b). The images cover a circular 140° field of view. A patient under investigation ingests the capsule, which then passes through the gastrointestinal tract. When the capsule goes through the small bowel it is propelled by peristaltic movements. The capsule transmits video data at a rate of two frames per second for approximately 8 hours. Investigation of the recorded video, usually numbering several tens of thousands of frames, is performed by a trained clinician. It is a tedious task that usually takes more than an hour. The video interpretation involves viewing the

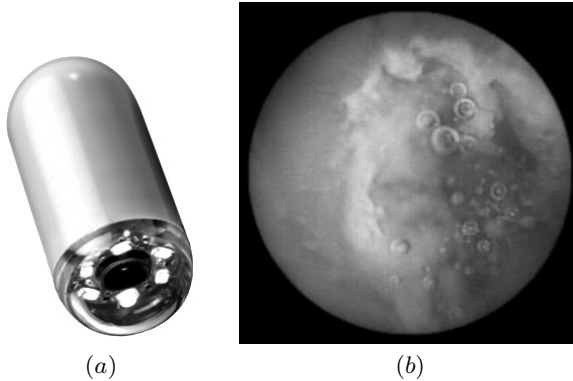


Fig. 1. Wireless capsule endoscope (a) and an example image (b)

video and searching for abnormal-looking entities like bleedings, erosions, ulcers, polyps and narrowed sections of the bowel.

In this study we examined four WCE video sequences among which we selected 20 sample images comprising of various forms of ulcerations, bleedings and erosions. Within these images, pathology regions were manually delineated by the cooperating gastroenterologists. Example images included in our experiments with their corresponding regions of interest are depicted in Fig. 2.

2.2 Local Polynomial Approximation

The task of the LPA algorithm invoked in the first step of the segmentation procedure is to define for each pixel a region that is adapted to the shape of this pixel local neighborhood. It is presumed that such a neighborhood exhibits homogenous color and texture characteristics. Consequently, it can be expected that in the subsequent texture analysis step pixels gain credible description, representative for a wider range of points. This in turn allows making the resulting texture description independent from local fluctuations in pixel values caused by noise and other imaging artifacts.

The LPA algorithm itself is a technique of non-parametric regression recently adopted in various image processing applications. Using low order polynomial function, LPA models a non-linear relationship between an independent variable X and a dependent variable Y . Data are fitted to a modeled polynomial function within a sliding window positioned at subsequent observations (X, Y) — e.g. measured values of a sampled signal. In a window, a signal is convolved with a kernel function of a known form. This enables estimating values of the Y signal in the neighborhood of a given data point X . Window size h is a key parameter of the method. It is defined as a number of data samples beyond which it becomes impossible to estimate signal Y basing on values measured in the proximal neighborhood of X .

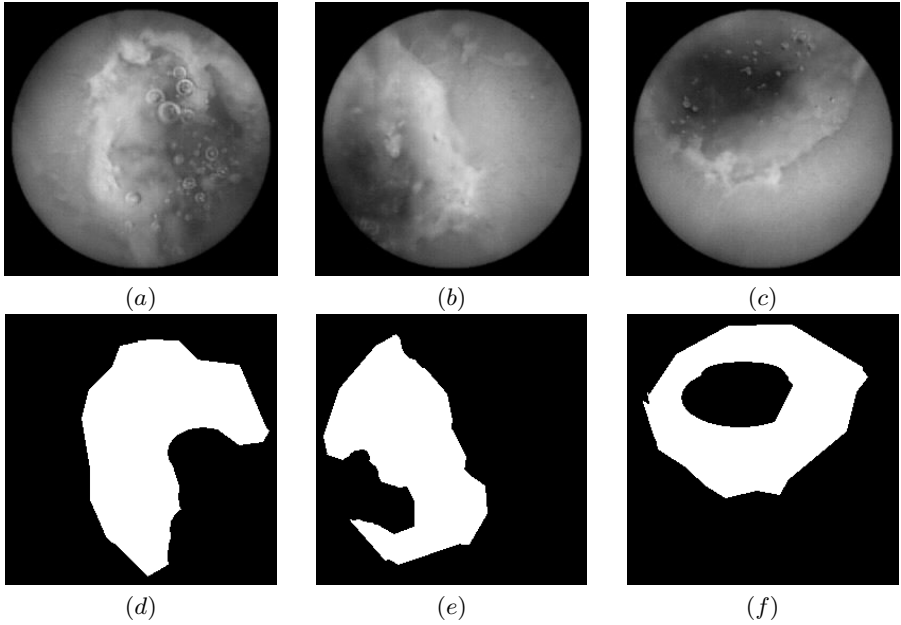


Fig. 2. Example WCE video frames (a-c) and their corresponding manually delineated regions of interest containing ulcerations with bleedings and narrowings (d-f)

In our study we apply the LPA algorithm following the approach presented in [13]. Firstly, a color WCE video frame is converted into a gray-scaled image. Next, we filter each pixel neighborhood in 8 distinct directions θ_i deviated from the horizontal East-oriented axis at angles $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$ and 315° . For a given pixel X we calculate

$$\mu^{(h)} = \sum_{j=1}^h g_j^{(h)} I(X + (j-1)\theta_i), \quad (1)$$

where $g^{(h)}$ is a discrete convolution kernel of scale h (window size), $g_j^{(h)}$ with $j = 1, \dots, h$ denote kernel weights which sum to unity and decrease with the increasing distance from a center pixel X . The exact procedure of weights generation is described in [1]. Eventually, I is a matrix of image intensity values.

Adjusting the window size to local image contents is performed using the *intersection of confidence intervals* (ICI) rule. The idea is to test several values of scale h , i.e. $h \in \{h_1, \dots, h_k\}$ and $h_1 < h_2 < \dots < h_k$ and for each of them evaluate (1) as well as local standard deviation value

$$\sigma_{\mu^{(h)}} = \sigma \|g^{(h)}\|, \quad (2)$$

where σ is the global standard deviation determined for the whole image. Then for each direction θ_i and scale h one calculates confidence intervals

$$\mathcal{D}_h = [\mu^{(h)} - \Gamma\sigma_{\mu^{(h)}}, \mu^{(h)} + \Gamma\sigma_{\mu^{(h)}}], \quad (3)$$

in which $\Gamma > 0$ denotes a global parameter that allows controlling noise tolerance. The lower Γ , the stronger requirement for local homogeneity is, and thus fewer pixels are included in the resulting neighborhood regions. The ICI rule states that for each direction one should choose a maximum value of h that ensures nonempty intersection of all previous confidence intervals, i.e.

$$h_{\max,i} = \max_{h \in \{h_1, \dots, h_k\}} \{h : (\mathcal{D}_1 \cap \mathcal{D}_2 \cap \dots \cap \mathcal{D}_h) \neq \emptyset\}. \quad (4)$$

In our experiments, we arbitrarily set $h \in \{1, 2, 3, 5, 7, 9, 12, 15, 18, 21\}$, hence the upper bound for the window size in any direction amounts to 21 pixels. On completion, pixels determined by relations $X + h_{\max,i}\theta_i$ constitute a set of polygon vertices whose interior determines a locally adapted ROI of a pixel X . Although we invoke LPA for a gray-scaled image, at the current stage of the research, for performance reasons, the resulting ROIs are used for each color channel in the texture analysis step.

2.3 Color and Texture Analysis

In order to exploit color information inherent in a WCE frame we convert it into 8 gray-scaled images using following color channels:

- brightness (according to the CCIR Recommendation 601-1),
- R, G, B, U and V color components,
- saturation and hue.

For each color channel and every pixel we then calculate its associated ROI first-order histogram (256 bins). The histogram is computed from the intensity of pixels, without taking into consideration any spatial relations between the pixels within the image. Features are simply statistical parameters of the histogram distribution such as: mean brightness, variance, skewness, kurtosis and percentiles [14]. However, distinct regions of WCE images differ mainly by color and intensity and second-order regularities appear less important. Moreover, determination of histogram features is time-efficient and thus reduces the computational load.

2.4 K-Means Clustering

As a last step in our segmentation technique we perform cluster analysis for unsupervised grouping of image pixels described by their corresponding ROI texture features. For that purpose we employ k-means algorithm [15] due to its low complexity and ease of implementation. An important drawback of k-means is its intrinsic tendency to get stuck in a local minimum of the optimization criterion (sum of squared distances between feature vectors and their corresponding cluster centers). Thus, we call a clustering procedure 5 times, each time with different initialization. Then, the result with the best score is chosen to represent final segmentation.

3 Results

In this preliminary research we have submitted all analyzed images into our segmentation procedure presuming arbitrarily that number of segments is equal to 6 in each case. By dividing an image into 6 segments we expect to encapsulate six possible patterns: ulcerations and accompanying lesions, bleedings, digestive residues, normal tissue and gastrointestinal lumen. This may of course lead to further subdivisions of some patterns if there are fewer distinguishable regions in a given frame.

In Fig. 3 we present example ROIs produced by the LPA algorithm for the image from Fig. 2a. Note, how regions A-C are well fitted to the surrounding tissue. On the other hand, region D placed in the interior of the intestinal canal embraces bubble-like structures whose borders emerge to be unimportant details in their local neighborhood. This observation shows that the algorithm — remaining local in nature — posses capability of reflecting general characteristics of a given pixel neighborhood.

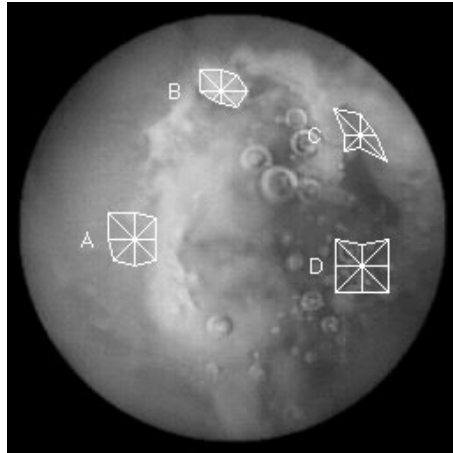


Fig. 3. Example ROIs defined by the LPA

Segmentation results obtained for three images viewed in Fig. 2a-c are depicted in Fig. 4a-c. Analysis of the segments borders reveals their fine matching to physical boundaries of ulcerations and bleedings. Moreover, it is apparent that identified image segments correspond well to the manually delineated regions of interest (cf. Fig. 4d-f). Although the latter occupy more than one automatically computed segments, those which are included to a large extent agree with the human expertise. In order to quantify a degree of this agreement, for each analyzed image we have calculated the Jaccard similarity coefficient between manually specified ROI and its respective automatically found and aggregated segments. The mean value obtained in these calculations amounted to 89% (with 4% of standard deviation).

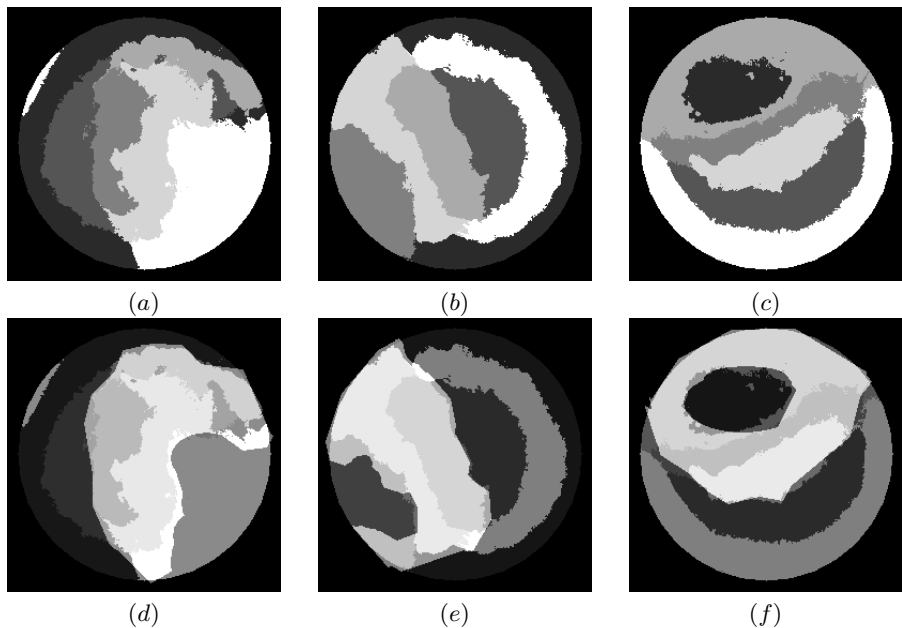


Fig. 4. Segmentation results (a-c) obtained for images from Fig. 2; d-f — identified segments overlaid on the manually delineated regions of interest

4 Summary

In this paper we presented a novel method for automatic segmentation of the WCE video frames. We demonstrated robustness of the proposed technique in application to several WCE images taken from real examinations. The obtained results exhibit high rate of agreement between automatically identified segments and regions manually delineated by gastroenterologists. Naturally, these results must be further validated with the use of larger set of sample images.

It must be noted that our study is still in the development phase. Further research is required to reduce time complexity associated with determination of pixels local neighborhoods. Number of LPA executions while processing a single WCE frame ranges the order of 10^5 . Fortunately, the LPA algorithm can be relatively easily parallelized and implemented efficiently e.g. on a multicore GPU. Moreover, postprocessing should be added after clustering step in order to cope with segmentation holes and islands, and also to identify clusters which include distal regions and thus should be subdivided.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education grant no. 3263/B/T02/2008/35.

References

1. Katkovnik, V., Egiazarian, K., Astola, J.: *Local Approximation Techniques in Signal and Image Processing*. SPIE Press (2006)
2. Coimbra, M., Cunha, J.: MPEG-7 visual descriptors—contributions for automated feature extraction in capsule endoscopy. *IEEE Transactions on Circuits and Systems for Video Technology* 16(5), 628–637 (2006)
3. Mackiewicz, M., Berens, J., Fisher, M., Bell, G.: Colour and texture based gastrointestinal tissue discrimination. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol. 2, pp. 597–600 (2006)
4. Mackiewicz, M., Berens, J., Fisher, M.: Wireless capsule endoscopy video segmentation using support vector classifiers and hidden markov models. In: *Proc. International Conference Medical Image Understanding and Analyses* (June 2006)
5. Mackiewicz, M., Berens, J., Fisher, M.: Wireless capsule endoscopy color video segmentation. *IEEE Transactions on Medical Imaging* 27(12), 1769–1781 (2008)
6. Bourbakis, N.: Detecting abnormal patterns in WCE images. In: *5th IEEE Symposium on Bioinformatics and Bioengineering (BIBE 2005)*, pp. 232–238 (2005)
7. Lau, P.Y., Correia, P.: Detection of bleeding patterns in WCE video using multiple features. In: *29th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society, EMBS 2007*, pp. 5601–5604 (August 2007)
8. Li, B., Meng, M.Q.H.: Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments. *Computers in Biology and Medicine* 39(2), 141–147 (2009)
9. Szczypinski, P., Klepaczko, A.: Selecting texture discriminative descriptors of capsule endoscopy images. In: *Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, ISPA 2009*, pp. 701–706 (2009)
10. Szczypinski, P., Klepaczko, A.: Convex hull-based feature selection in application to classification of wireless capsule endoscopic images. In: *Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS*, vol. 5807, pp. 664–675. Springer, Heidelberg (2009)
11. Iddan, G., Meron, G., Glukhowsky, A., Swain, P.: Wireless capsule endoscopy. *Nature* 405(6785), 417–418 (2000)
12. Swain, P., Fritscher-Ravens, A.: Role of video endoscopy in managing small bowel disease. *GUT* 53, 1866–1875 (2004)
13. Bergmann, Ø., Christiansen, O., Lie, J., Lundervold, A.: Shape-adaptive DCT for denoising of 3d scalar and tensor valued images. *Journal of Digital Imaging* 22(3), 297–308 (2009)
14. Szczypinski, P., Strzelecki, M., Materka, A., Klepaczko, A.: MaZda - a software package for image texture analysis. *Computer Methods and Programs in Biomedicine* 94, 66–76 (2009)
15. Duda, R., Hart, P., Stork, D.: *Pattern Classification*. John Wiley & Sons, Chichester (2001)

A Proper Choice of Vertices for Triangulation Representation of Digital Images

Ivana Kolingerová*, Josef Kohout, Michal Rulf, and Václav Uher

Department of Computer Science and Engineering
University of West Bohemia, Pilsen, Czech Republic
kolinger@kiv.zcu.cz

Abstract. Representation of the digital image by a triangulation does not bring a high compression but enables geometric transformations and is very simple. In this paper we will show that it is also possible to choose the triangulation vertices randomly, then their $[x, y]$ position does not need to be stored as it can be easily reconstructed during decoding. We show how such a choice behaves in comparison and in combination with the vertices selected from the edges of the digital image.

Keywords: Digital image, Triangulation, Representation, Compression.

1 Introduction

Digital images are often alternatively represented by triangulations. Such a representation brings a low compression in comparison with frequency-based methods but enables to apply geometric transformations and is very simple. Therefore, it may be a reasonable choice in some cases.

In order to represent a digital image by a triangulation, some pixels of the images must be chosen as vertices of the triangulation. On this set of vertices, a triangulation is constructed, using intensity values in the vertices as the z -coordinate. The intensity in other pixels is not stored but is approximated by an interpolation on the triangles. According to our experience, the kind of triangulation used has lower influence than the choice of vertices. It is clear that to represent the digital image properly, the most important pixels should be chosen but to decide which they are is difficult. Usually the points on edges are taken.

The idea presented in this paper is to choose pixels randomly, then their x and y coordinates do not have to be stored as they can be re-generated in the decoding process, so only intensity values in these pixels have to be kept. Our paper shows how this simple strategy behaves in comparison and in combination with the vertices selected from the edges in the digital image. We present the results on grey-scale images, however, application on colour images is also possible with similar results.

* This work was supported by the Grant Agency of Academy of Sciences of the Czech Republic (GA AV ČR), project No.KJB10470701.

Section 2 presents state of the art in the digital image representation by triangulations, section 3 describes a brief description of this approach, and the proposed method of vertex choice. Section 4 presents experiments and results, section 5 concludes the paper.

2 State of the Art

Many methods to convert digital images from the raster representation into triangulations exist. Three groups can be detected. The first group is oriented on enhancement of the quality of further image processing. Such representations usually use nearly all pixels as triangulation vertices and use the data dependent triangulation as its base. The methods differ in edge detection and in optimisation [1, 14, 15].

The second group tries to achieve as high compression as possible, even in the price of lower precision. They find its use in applications of non-photorealistic rendering where details are not desirable. Typical representatives of this class are: [12] which uses Canny operator to detect the edges with constrained Delaunay triangulation to include these edges and [11] where random choice of vertices is gradually improved by simulated annealing and used in Delaunay triangulation.

The last group tries to achieve the highest possible compression while preserving the highest possible image quality. To achieve this goal, an initial choice of vertices is done and an initial triangulation constructed, then either new vertices are added or, vice versa, existing vertices removed until the quality and compression are balanced. Two simple and not too efficient approaches of this type are [8] and [3]. Later methods of this group can be found in [13,7,6]. The best method of this type can be found in [4] where gradual decimations are used, removing in each step the least important vertex, importance is recognized according to the values of Mean Square Error (MSE). Image quality and compression of this method are comparative to JPEG. However, the method is very slow due to its slow progress of decimation.

More information can be found in [10].

3 Digital Image Representation by a Triangulation

As the first step, a proper number of vertices is assessed. It can be either computed from the required compression rate, or given by the user. Then the set of pixels is chosen. In the last step coordinates of these pixels are used as vertices of a triangulation. We recommend using Delaunay triangulation as its computation is rather easy and efficient. Our experiments did not acknowledge that data dependent or other types of triangulations would have behaved better.

Output representation is formed by triangulation vertices and intensities in these points. The triangulation topology does not need to be stored as it can be re-computed in decoding, if the triangulation algorithm is unambiguous. The already mentioned Delaunay triangulation has one ambiguity: if four points forming two neighbouring triangles lie on a circle, then there are two possible pairs

of triangles of these four points. This ambiguity may and should be removed by including some simple and unambiguous rule for such cases, e.g., to use in such a case the diagonal with lower intensity gradient.

In the process of decoding, values of intensity inside triangles are interpolated from the known vertex intensity values. Bilinear interpolation serves well for this purpose, although more sophisticated methods may bring some improvement [9].

In existing methods, pixels are chosen if they belong to edges of the digital image.

A pixel is considered to belong to the edge, if

$$O(x, y, I_{m,n}) > Th$$

where $[x, y]$ is the pixel on the image coordinates $[x, y]$, $I_{m,n}$ is a matrix of intensities of the digital image ($I[x, y]$ being the intensity value in the $[x, y]$ pixel), $O(x, y, I_{m,n})$ is the value of a particular operator in the $[x, y]$ pixel, Th is a threshold between edge and non-edge pixel intensity.

It is difficult to estimate the threshold for the given image in advance, if it is too high or too low, another round of evaluation with a different threshold is needed. To avoid this delay, some pre-computation on a small subset of the digital image may be done and the threshold value found before processing the whole image.

Let us briefly survey the most important edge detection operators used in this context. They are based on the gradient computation. The simplest is the Roberts's operator which computes gradient with the use of three neighbouring pixels (see Fig.1a):

$$\text{Roberts:} O(x, y, I_{m,n}) = \text{abs}(I[x, y] - I[x + 1, y + 1]) + \text{abs}(I[x + 1, y] - I[x, y + 1])$$

Laplace operator is based on convolution. It is used in two versions, using four or eight neighbouring pixels (see Fig.1b and 1c):

$$\begin{aligned} \text{Laplace4:} O(x, y, I_{m,n}) &= \text{abs}(I[x, y - 1] + I[x, y + 1] + I[x - 1, y] + \\ &+ I[x + 1, y] - 4 * I[x, y]) \end{aligned}$$

$$\begin{aligned} \text{Laplace8:} O(x, y, I_{m,n}) &= \text{abs}(I[x - 1, y - 1] + I[x - 1, y] + I[x - 1, y + 1] + \\ &+ I[x, y - 1] + I[x, y + 1] + I[x + 1, y - 1] + \\ &+ I[x + 1, y] + I[x + 1, y + 1] - 8 * I[x, y]) \end{aligned}$$

The Gaussian operator uses the Gaussian-weighted differences of intensity of the pixels to its neighbours (see Fig.1d):

$$\text{Gauss:} O(x, y, I_{m,n}) = \sum_{i=-v}^r \sum_{j=-v}^r |I[x, y] - I[x + i, y + j] * \exp(-(i^2 + j^2)/2/\sigma^2)|$$

where σ and r , respectively, define the vicinity area and influence factor of the point in this area, respectively; usually $\sigma = 24$ and $r = 4$ or 8 . The idea is that the closer points to the current pixel should have higher importance.

For colour images, the edge detection can be either used for each colour component, or the colour values can be recomputed to the intensity as follows:

$$I(x, y) = 0.3 * R[x, y] + 0.59 * G[x, y] + 0.11 * B[x, y]$$

where $R[x, y]$, $G[x, y]$, $B[x, y]$ are the colour components in $[x, y]$.

We suggest another possibility how to choose the points - to generate them randomly. Then it is not necessary to store the $[x, y]$ coordinates as they can be re-generated in the decoder - using the same seed for the random number generator or using our own random number generator. In the following section we will show how such a choice behaves in comparison with the edge points.



Fig. 1. The 9-10% pixels with the highest evaluation according to the presented operators, a) Roberts, b) Laplace4, c) Laplace8, d) Gauss

4 Experiments and Results

We implemented the whole method (including the Delaunay triangulation) in Delphi under MS Windows and tested on Intel Core Duo CPU 2.8 GHz with 4 GB memory. The testing images were Lena, Fruits, Boat, Goldhill, Barbara, Mountain, all 512×512 pixels (262 144 pixels), 256 levels of grey. These images

were downloaded from the USC-SIPI Image Database [15]. 1% of selected points corresponds to approximately 2620 pixels. The quality will be measured by MSE.

The image pixels chosen by an edge operator will be called 'edge points' and the pixels chosen randomly, with uniform distribution will be called 'random points'.

It should be pointed out that a maximum rate of selected edge points which has sense is $1/5$ to $1/3$ of the total number of pixels as for each point, 2 coordinates per 4B plus intensity value (2-4B) must be kept. The coordinates of random points do not need to be stored as they are re-generated from the seed value (1B, one value for the whole file).

First, we want to find out which of the edge detecting operators provides the highest fidelity of the resulting image while achieving at least some compression. We measured the Roberts, Laplace4, Laplace8 and Gauss operators. We took edge points and combined them with random points. 100% corresponds to all pixels of the original image. Typical results are in Figs.2,3. The worst results were achieved by the Gauss operator which is also the slowest one. The Laplacian operators are the best namely for a low number of edge points and a high number of random points, see Fig.2. This is a desirable combination as random points do not need to be stored. For higher number of edge points and a low number of random points, Roberts may bring better results, see Fig.3. With a higher number of random points, this behaviour disappears.

In the following test we want to find a proper rate of random and edge points. The results when only random points are used are not good, see Fig.4a - edge points which help to keep sharp edges are missing. Also the opposite extreme is not satisfactory - random points which help to recover flat areas are missing, see Fig.4b. Fig.5 shows the dependence of MSE on the total number of points, it can be seen that in comparison with Fig.2 and 3 the results are worse.

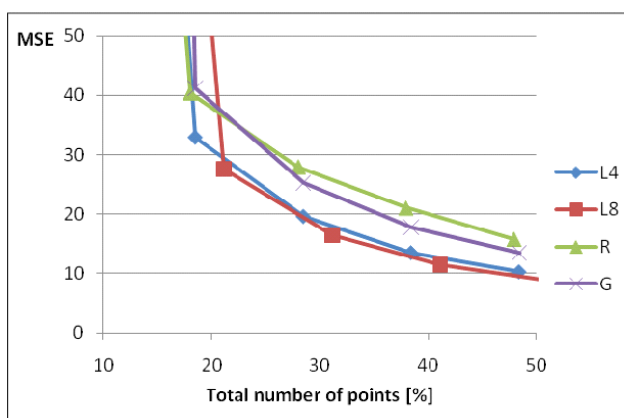


Fig. 2. The image Fruits Boat: Dependence of MSE on the total number of points of which 8-10% are edge points

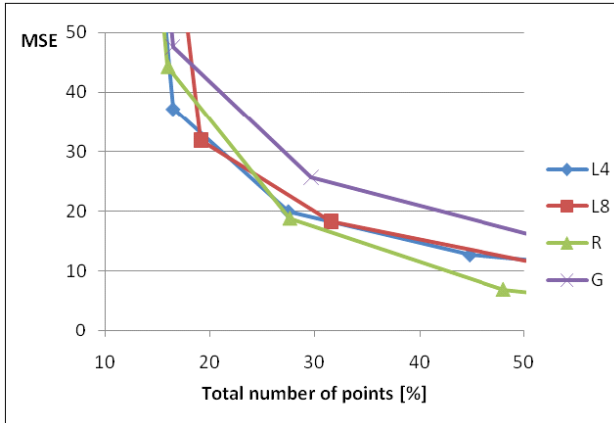


Fig. 3. The image Fruits: Dependence of MSE on the total number of points of which 8% are random points

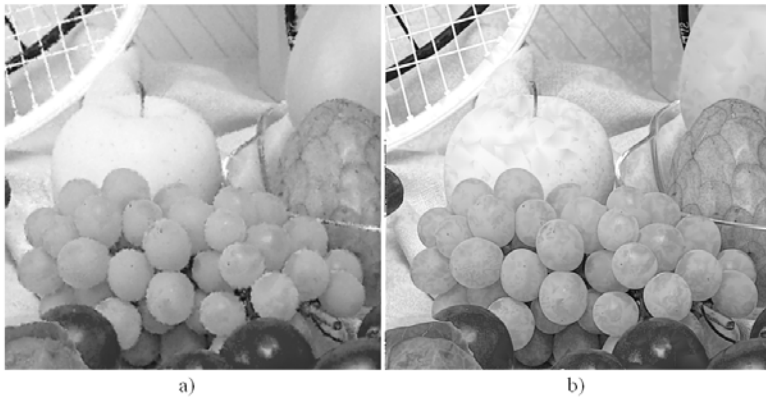


Fig. 4. The image Fruits with 20% of points: a) only random points (MSE 99.11), b) only edge points (MSE 131.94)

If we combine both types of points, two types of behaviour can be observed according to the image type: the „normal” images, which do not contain an extreme number of edges, are less sensitive to the type of points than to their total number. The images which are full of edges and of very small areas need also more edge points (Mountain, Barbara). For most images, the suitable number of random points to obtain an acceptable quality plus some compression is 10-15% of the image size. The suitable number of edge points depends on the image type. For common images about 5-10% of edge points can be enough, for images with many edges, 15-20% may be necessary. Fig.6 shows an example when approximately 26% of points are used.

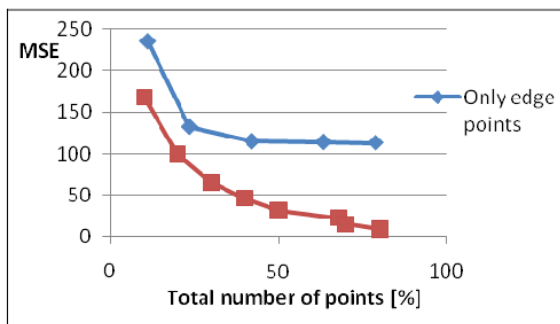


Fig. 5. The image Fruits: Only random and only edge points



Fig. 6. The image Fruits; a) input, b) result - 11% of edge points, 15% of random points, MSE=20.65

5 Conclusion

We showed that a random choice of representative points in the digital image to be used as vertices of a triangulation, in spite of simplicity of this approach, may be a viable alternative to edge points as only intensities in these points have to be stored, while their coordinates can be re-generated in decoding from a seed of a random generator.

References

1. Battiato, S., Gallo, G., Messina, G.S.: SVG Rendering of Real Images Using Data Dependent Triangulation. In: Proceedings of SCCG 2004, pp. 185–192 (2004)
2. De Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational geometry. In: Algorithms and applications. Springer, Heidelberg (1997)

3. Ciampalini, A., Cignoni, P., Montani, C., Scopigno, R.: Multiresolution Decimation Based on Global Error. In: *The Visual Computer*, vol. 13, pp. 228–246. Springer, Heidelberg (1997)
4. Demaret, L., Dyn, N., Iske, A.: Image compression by linear splines over adaptive triangulations. *Signal Processing* 86(7), 1604–1616 (2006)
5. Dyn, N., Levin, D., Rippa, S.: Data Dependent Triangulations for Piecewise Linear Interpolation. *IMA J. Numer. Anal.* 10, 137–154 (1992)
6. Galic, I., Weickert, J., Welk, M.: Towards PDE-based Image Compression. In: Paragios, N., Faugeras, O., Chan, T., Schnörr, C. (eds.) *VLSM 2005*. LNCS, vol. 3752, pp. 37–48. Springer, Heidelberg (2005)
7. García, M.A., Vintimilla, B.X., Sappa, A.D.: Efficient Approximation of Greyscale Images through Bounded Error Triangular Meshes. In: *Proceedings of IEEE International Conference on Image Processing*, Kobe, Japan, pp. 168–170. IEEE Computer Society, Los Alamitos (1999)
8. Gevers, T., Smeulders, A.W.: Combining Region Splitting and Edge Detection through Guided Delaunay Image Subdivision. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1021–1026 (1997)
9. Janák, T.: On Interpolation for Triangulation - represented Digital Image. In: *CESCG 2008*, Budmerice, Slovakia, April 24–26 (2008)
10. Kohout, J.: Alternative Representation of Image Information, Technical report DCSE/TR- 2009-11, UWB (2009)
11. Kreylos, O., Hamann, B.: On Simulated Annealing and the Construction of Linear Spline Approximations for Scattered Data. *IEEE Transactions on Visualization and Computer Graphics* 7, 17–31 (2001)
12. Prasad, L., Skourikhine, A.N.: Vectorized Image Segmentation via Trixel Agglomeration. *Pattern Recognition* 39, 501–514 (2006)
13. Rila, L.: Image Coding Using Irregular Subsampling and Delaunay Triangulation. In: *Proceedings of SIBGRAPI*, pp. 167–173 (1998)
14. Su, D., Willis, P.: Image Interpolation by Pixel Level Data-dependent Triangulation. *Computer Graphics Forum* 23, 189–201 (2004)
15. Weber, A. (ed.): *The USC-SIPI Image Database*. Signal & Image Processing Institution, University of Southern California (December 2007), <http://sipi.usc.edu/database/>
16. Yu, X., Morse, B.S., Sederberg, T.W.: Image reconstruction using data-dependent triangulation. *IEEE Computer Graphics and Applications* 21, 62–68 (2001); This work was supported by the Grant Agency of Academy of Sciences of the Czech Republic (GA AV R), project No.KJB10470701

A Method for Novel Face View Synthesis Using Stereo Vision

Jacek Komorowski¹ and Przemyslaw Rokita²

¹ Maria Curie-Skłodowska University, Institute of Computer Science, Lublin, Poland
jacek.komorowski@gmail.com

² Warsaw University of Technology, Institute of Computer Science, Warsaw, Poland
p.rokita@ii.pw.edu.pl

Abstract. This paper presents a method for synthesis of a novel view of a human face using a pair of images from a stereo camera set. Unlike other published methods our approach does not require an a priori knowledge (e.g. 3D head model) of the synthesized object shape. First 3D shape of a face is recovered using stereo vision technique and then an image from a novel viewpoint is synthesized. Proposed method is applicable in various image-based pattern recognition and biometric tasks as it allows to partially alleviate dependency on object orientation in space.

1 Introduction

Face recognition for biometric purposes is a field of a great practical importance and active development [9]. However classical image-based recognition methods, such as Principal Component Analysis (eigenfaces), Fisher's Linear Discriminant or Independent Component Analysis are very dependent on lighting and head orientation [8]. This is because 2D images of 3D face objects change significantly due to lighting and viewpoint variations.

Advanced face recognition techniques, such as fitting to a 3D morphable model [1], overcome this difficulty by using a 3D morphable model of a human face. The model is fit to the face image being recognized and fitting parameters are used as features in subsequent recognition phase. This method is very application specific, as it can be used only for objects of a known shape which 3D model is encoded in the recognition system. Another drawback is a necessity for very expensive equipment (e.g. laser scanners) to obtain a detailed 3D human face model.

The goal of our research was to build a system which allows to synthesize a novel view of a human face without requirement for a priori 3D head model. This would allow to build a face recognition system independent, to some extent, on head orientation. A required view of a human face could be synthesized and fed into subsequent recognition subsystem, using one of the classical, view dependent, image-based recognition methods (e.g. PCA).

Main benefit of our approach is flexibility, as it can be used to synthesize novel views of any object, making it useful in various recognition tasks. There's no need for an a priori 3D model of an object being analyzed.

2 Novel View Synthesis

Our solution for novel view synthesis uses a depth-image-based rendering concept [3]. Depth-image-based rendering is a process of synthesizing virtual views of a scene from one or more input images and associated per-pixel depth information. It can be described as a two step activity. First, original image points are reprojected into 3D space, using supplied per-pixel depth data. Then 3D points are reprojected into an image plane of a virtual camera, located at the required viewing position. The concept is visualized on Fig. 1.

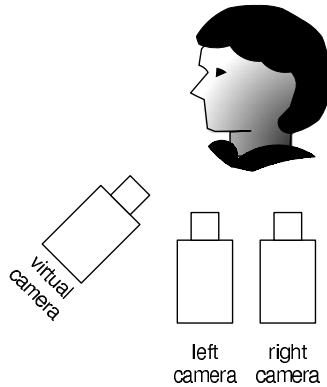


Fig. 1. Virtual view synthesis concept

To generate per-pixel depth data we use stereo vision technique. Two cameras capture simultaneously two images of the scene. These images are used to calculate depth information for each pixel of one of the images (called reference image).

Our stereo rig consist of 2 Chameleon digital cameras from Point Grey Research¹ (CCD sensor, 1296x964 resolution). Cameras are aligned horizontally with 8 cm distance between optical axis.

As an initial step the stereo rig is calibrated using OpenCV² library implementation of Zhang algorithm [10]. This is done by capturing a number of images of a chessboard calibration pattern placed at different positions behind the stereo rig. Knowing geometry of a calibration pattern it's possible to calculate intrinsic and extrinsic parameters of the stereo rig [2]. In the subsequent steps the following parameters computed by stereo cameras calibration function are used: left and right camera 3x3 intrinsic matrices (K_l^* , K_r^*), left and right camera distortion coefficient vectors (d_l , d_r), 3x3 rotation matrix (R_r) and 3x1 translation vector (T_r) relating left and right cameras.

After initial stereo rig calibration the following steps are implemented to synthesize a view of a human face from a novel viewpoint:

¹ See <http://www.ptgrey.com/> for technical specification.

² Available at <http://opencv.willowgarage.com/wiki/>

1. Stereo image acquisition
2. Images rectification
3. Segmentation
4. Depth map computation
5. Virtual view synthesis
6. Post processing

2.1 Image Acquisition

A subject is placed approximately 60 centimeters in front of the stereo rig on a neutral background (white wall). Grey level images are acquired simultaneously by a left and right camera at 1296x964 resolution and 256 intensity levels. An example of a captured image pair is presented on Fig. 2.



Fig. 2. Pair of stereo images from left and right camera

2.2 Rectification

The goal of a rectification procedure is twofold: first, to remove camera distortions from acquired pair of images and second, to project images onto a common image surface so that their epipolar lines are aligned horizontally. This simplifies subsequent step of a disparity function computation, as after rectification pixels from a left and right image corresponding to the same scene point have equal y-coordinate.

In our solution rectification is done using OpenCV library implementation of Bouguet's algorithm. See [2] for more information. Rectification procedure takes as an input a pair of stereo images and stereo rig intrinsic (K_1^*, K_r^*, d_l, d_r) and extrinsic (R_r, T_r) parameters calculated during an initial calibration. As the result we obtain a pair of rectified images and intrinsic matrices of a left and right rectified camera (K_l and K_r respectively). Example of rectified stereo images can be found on Fig. 3. Note, that pixels corresponding to the same point of a scene have identical y-coordinate. All further processing is done on rectified images using rectified cameras intrinsic matrices.

2.3 Segmentation

For simplification we choose a left image as a reference one, that is an image used as a basis for further view synthesis. In general case we should choose as a reference image the one which is closer to the required view.

A segmentation procedure produces a set of pixel coordinates $\mathcal{F} \subseteq \mathbb{N} \times \mathbb{N}$ constituting a region of the reference image containing a face. In our approach we use a simple segmentation technique based on thresholding. As subjects are placed on a neutral background (white wall) this technique gives satisfactory results.

2.4 Depth Map Computation

Let (X, Y, Z) be a scene point in a camera coordinate system and (x, y) coordinates of its projection onto a camera image plane. Depth map $depth(x, y)$ is defined for each pixel (x, y) as a Z-coordinate (depth) of a point (X, Y, Z) , that is $depth(x, y) = Z$.

Key step needed to recover 3D information about the scene from a pair of stereo images is a disparity function computation. Disparity is defined [5] as a difference between coordinates of the projections of the same scene point onto left and right image planes. Recall that in rectified images these projections have the same y-coordinate, thus only the difference between x-coordinates needs to be considered.

Semi-global matching using mutual information method [4] was used to compute a disparity map $disp(x, y)$ for each pixel (x, y) of a reference image contained in a face region \mathcal{F} . An example disparity map for a left-camera image from Fig. 3 is presented in Fig. 4.

Depth is inversely proportional to a disparity. Assuming left and right cameras have the same focal length f and distance between their principal points is T_r depth can be calculated from below formula:

$$depth(x, y) = \frac{f||T_r||}{disp(x, y)} . \quad (1)$$

2.5 Virtual View Synthesis

Having computed depth map $depth(x, y)$ for the face region \mathcal{F} in the reference image we can synthesize a virtual view from a novel viewpoint the following way.

Let's assume that the world coordinate system equals to a reference camera coordinate system. Let $\mathbf{M} = (X, Y, Z, 1)^T$ be 3D homogeneous coordinates of the scene point and $\mathbf{m}_1 = (x, y, 1)$ its projection onto reference camera image plane. \mathbf{M} and \mathbf{m}_1 are related by the camera perspective projection equation [7]:

$$z_1 \mathbf{m}_1 = K_1 [I|0] \mathbf{M} \quad (2)$$

where z_1 is the scene point depth, K_1 is a 3x3 rectified reference camera intrinsic matrix and $[I|0]$ is 4x3 matrix consisting of 3x3 identity matrix and 3x1 zero vector.



Fig. 3. Rectified images



Fig. 4. Disparity map of a left face from Fig. 3. Higher intensity corresponds to bigger disparity.

Rearranging (2) we can derive 3D coordinates of a scene point M :

$$M = z_1 (K_1)^{-1} m_1 . \quad (3)$$

Let R_v be a given 3x3 rotation matrix and T_v 3x1 translation vector defining the required position of a virtual camera relative to the reference camera. In order to calculate the projection m_v of a point M having $(X, Y, Z, 1)^T$ homogeneous coordinates in the reference camera coordinate system onto a virtual camera image plane the following perspective projection equation should be used:

$$z_v m_v = K_v [R_v | T_v] M . \quad (4)$$

Assuming that virtual camera intrinsic matrix K_v is the same as the reference camera intrinsic matrix K_1 and substituting (3) into (4) we obtain:

$$z_v m_v = z_1 K_1 R_v (K_1)^{-1} m_1 + K_v T_v . \quad (5)$$

Depth from a virtual camera z_v can be assumed arbitrary as homogeneous coordinates scaled by a constant factor identify the same point. As depth from a reference camera $z_1 = \text{depth}(x, y)$ we obtain:

$$z_v \mathbf{m}_v = \text{depth}(x, y) K_1 R_v (K_1)^{-1} (x, y, 1)^T + K_v T_v . \quad (6)$$

Applying (6) for a point $(x, y, 1)^T$ from reference image we can calculate a point \mathbf{m}_v on the synthesized image it's mapped to. Repeating this procedure for all points from a segmented face region \mathcal{F} on a reference image produces a required novel view.

Visibility problem must be addressed during rendering, as more than one pixel from a reference image may correspond to the same pixel in a synthesized image. This is solved using simple z-buffering technique. When 3D point is projected onto a novel image plane it's z coordinate is stored in a buffer, arranged as a two-dimensional array (x, y) with one element for each screen pixel. If another point of the scene is projected on the same pixel, it's z -coordinate is compared with the one stored in the buffer. If it's closer to the camera image plane it's rendered and it's z coordinate is saved to the z -buffer, replacing the old one. Example of a rendered image is shown on Fig. 5(a).

2.6 Post Processing

Apart from occlusion additional real-world problems arise during depth-based image rendering [3]:

- when mapping pixels from a reference image some pixels in synthesized image are not assigned any value (resampling problem),
- parts of a scene not visible in a reference image may become visible in a novel view (disocclusion problem).

A solution to resampling issues when standard, two dimensional, image warping is done, is reverse mapping. Each pixel on a generated image is mapped back into an original image. When mapped pixel has fractional coordinates and it's intensity value is calculated using interpolation (e.g. bilinear). This approach is not feasible for 3D image warping, as dependency between coordinates of pixels on a synthesized image and a reference image is not explicit. Two neighborhood pixels on a reference image may be mapped onto distant points on the synthesized image, depending on their depth values.

Occlusion problems seems to be even more difficult to tackle, as generally there's not possible to get intensity values for points not visible on a reference image and disoccluded on a synthesized image. Image from the second camera usually won't be helpful, as parts of the image e.g. occluded by a nose on a reference image are also occluded on the second image.

To deal with these problems we successfully use inpainting technique. Inpainting is a process of reconstructing lost or corrupted parts of images using values from neighborhood pixels. Opposite to simple interpolation, this technique attempts to continue the isophotes (lines of equal gray value) as smoothly

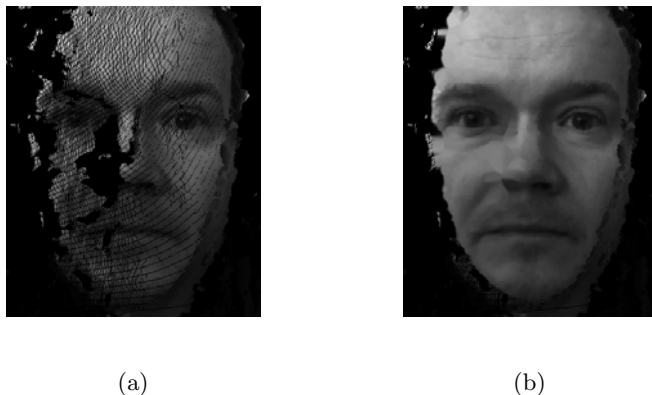


Fig. 5. Synthesized view from a novel viewpoint

as possible inside the reconstructed region, in order to produce a perceptually plausible reconstruction. Telea inpainting algorithm [6] is deployed and results are visible on Fig. 5(b).

3 Conclusions and Future Work

Proposed method renders visually satisfactory novel views, as can be seen on Fig. 5(b) where a face frontal view was generated based on a pair of stereo images from Fig. 2. Although there are some difficulties which still must be solved:

- There are significant holes in a created disparity map, especially near the high curvature regions (e.g. nose). This causes holes on a rendered view which are not accurately filled by inpainting technique. Improvement in a depth map calculation algorithm will be beneficial.
- Due to occlusion some areas of the face cannot be rendered properly. Estimation of the face orientation and using of symmetry assumption may allow to solve this issue.

Although visual results are encouraging the proposed method was not yet tested with image-based face recognition system so it was not proved that it's beneficial for face recognition purposes. Next step is a creation of an image-based face recognition system, using one of the standard methods (e.g. PCA - eigenfaces) to prove that proposed technique is advantageous for use in a biometric system as it can partially alleviate dependency of head orientation.

References

1. Blanz, W., Vetter, T.: Face Recognition Based on Fitting a 3D Morphable Model. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1063–1074 (2003)

2. Bradski, G., Kaehler, A.: *Learning OpenCV*. O'Reilly Media, Sebastopol (2008)
3. Fehn, C.: Depth-Image Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV. In: *Proceedings of the SPIE, Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291, pp. 93–104 (2004)
4. Hirschmuller, H.: Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 2, pp. 807–814 (2005)
5. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* 47(1-3) (April 2002)
6. Telea, A.: An image inpainting technique based on the fast marching method. *Journal of Graphics, Gpu, and Game Tools* 9(1), 23–34 (2004)
7. Schreer, O., Kauff, P., Sikora, T.: *3D Videocommunication Algorithms, concepts, and real-time systems in human centered communication*. John Wiley & Sons, Ltd., Chichester (2005)
8. Zhao, W., Chellappa, R.: Image-based Face Recognition: Issues and Methods. In: Javidi, B., Dekker, M. (eds.) *Image Recognition and Classification*, pp. 375–402 (2002)
9. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.: Face Recognition: A Literature Survey. *ACM Computing Surveys*, 399–458 (2003)
10. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(11), 1330–1334 (2000)

Outer Surface Reconstruction for 3D Fractured Objects

Anatoly Kornev¹, Laurent Babout¹, Marcin Janaszewski¹, and Hugues Talbot²

¹ Computer Engineering Department, Technical University of Łódź
18 Stefanowskiego Str., 90-924 Łódź, Poland
{lbabout, janasz, akornev}@kis.p.lodz.pl

² Departement Informatique, Université de Paris-East, 2
Bd Blaise Pascal 93162 Noisy-Le-Grand Cedex, France

Abstract. We study surface reconstruction using a combination of anisotropic Gaussian filter and image segmentation technique -the minimum surface method. Anisotropic Gaussian filtering allows to manage a contrast between intensities of the discontinuity and the object in a desired direction. The minimum surface method detects properly outer boundaries even affected by boundary leakage in the vicinity of blurred edges. The algorithm is tested on a set of real 3D images of large corrosion cracks in stainless steel that initiated at the surface of the tested samples. Results are presented and discussed.

Keywords: CT, Image Segmentation, Gaussian Filtering, Minimal Surface.

1 Introduction

The detection and extraction of cracks or discontinuities are among the common problems found in material science which utilize image analysis to obtain microstructural information. Nowadays, the results of 3D image processing assist in the design of new materials and calculation of mechanical, physical and chemical properties. Some of the problems that can be found in the analysis is an estimation of the damage of broken materials. That leads to the separation of the cracks from the environment. However, this can be a challenge if the cracks initiate at the surface of the sample and propagate in the bulk of the material. This is encountered frequently during service where degradation process is generated from external solicitation such as fatigue [1,2] or environmental induced cracking [3].

The problem definition is as follows (see Fig.1). On the scene we have a fractured object with a large number (e.g. > 20) of discontinuities or cracks surrounded by air. The intensity of air color is assumed constant everywhere and differ from the object color. The cracks together with the environment may be easily segmented by simple thresholding. The task is to extract the cracks from the environment or find the boundary Γ_h of fractured object.

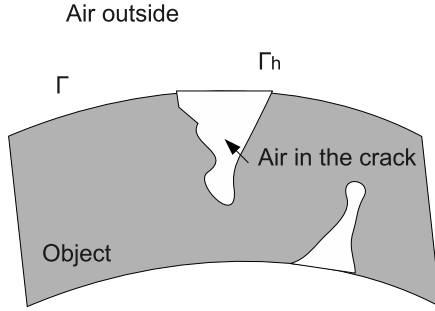


Fig. 1. The geometry of the problem

The existing methods of surface restoration for fractured objects handle it mainly by two steps. Firstly detect borders of all holes and discontinuities. In two dimensions (2D) it is a well-known procedure but its complexity and probable errors will increase when detection moves to 3D. Secondly interpolate or extrapolate the surface using border points and available algorithm from computational mathematics. Historically, in the computer aided design (CAD) applications the B-spline interpolation was utilized [4,5,6]. The marching cubes [7] also belongs to the interpolation technique. Both are based on the triangular mesh generation. The additional geometric partial differential equations were introduced in [8,9] to generate a mesh or to find an indicator function [10] and to restore a surface further. In [8,10] there are good reviews of related work. All above mentioned methods will require detection of crack boundaries and computing of either a mesh or additional geometrical parameters.

Deformable models [11,12], active surface [13] and active volume models [14] serve primarily for segmentation purposes but may be applied to surface reconstruction with substantial modifications. The recent technique [15] based on membrane potentials is best suited to recover smooth surfaces from noisy and sparse data sets. Therefore the input for the method is noisy point clouds on the existing surface.

2 Proposed Approach

For our 3D image samples the number of features to be extracted, i.e. cracks that initiate at the surface of the sample, is large and needs to be processed automatically. Therefore we proposed a method which avoids manual detection of crack borders. Firstly we process an image using anisotropic Gaussian filter in the direction parallel to the longitudinal axis of the object (assuming that the sample at best is straight and at worst, has a reasonable bending). The role of the Gaussian filter is to decrease the contrast between the intensities of the object and the crack, especially in places where the crack opening is large (e.g. > 20 voxels). Secondly we segment the processed image affected by boundary

leakage in the vicinity of blurred edges using minimum surface algorithm [16]. The algorithm is designed especially to detect degraded objects and to minimize their boundary length.

A Gaussian filtering is generated by convolving an image with a kernel of Gaussian values

$$I(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp^{-\frac{x^2+y^2+z^2}{2\sigma^2}} * I_o(\mathbf{x}), \quad (1)$$

where I_o and I are original and filtered images, $\mathbf{x} = (x, y, z)$ is a Cartesian coordinate system, and σ is the standard deviation.

For simplicity, let us obtain the formula for oriented Gaussian filter in 2D and consider coordinate system adapted to object boundary, e.g. u -axis and v -axis are parallel and perpendicular to the longitudinal axis of the object, respectively. The transformation from (x, y) to adapted coordinate system (u, v) will be

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad (2)$$

where ϕ is an angle between x and u axes. We want to use oriented anisotropic Gaussian filtering [18,19] along coordinate u only, therefore

$$I = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp^{-\frac{u^2}{2\sigma^2}} * I_o. \quad (3)$$

The filtered image is after processed using the minimal surface method that is described below. Note that a more detailed description of the method can be found in [16,17].

Let G be a Riemannian metric on Ω , a compact subset of \mathbb{R}^n . Let us consider two disjoint subsets of Ω : the source S_o (i.e. an input object that is used as starting object for the flow propagation. Usually, this input object is the segmented sample) and the sink S_i (e.g. an object that bounds the input object and in which the flow is contained). Let all closed hyper-surfaces s (not necessarily connected) be those that contain the source and do not contain the sink. The energy or cost E of s can be defined as follows:

$$E(s) = \oint_s dG. \quad (4)$$

The geodesic metric G is given by

$$dG = g(\mathbf{x})d\mathbf{x} = \frac{d\mathbf{x}}{1 + (\nabla I)^2}, \quad (5)$$

where ∇I is the gradient of the intensity of gray color. The minimum surface can be computed using the well-known Ford and Fulkerson (FF) maximum flow graph algorithm [20]. This corresponds to the well-known Graph-Cut (GC) problem. However, GC presents metrication artifacts, meaning that the computed minimal surfaces tend to follow the principal directions of the grid. Here we require an isotropic solution, which continuous maximum flows (CMF) can provide.

Therefore the CMF framework is applied. The set of points is replaced by a continuous scalar field - pressure P and the set of the edges is replaced by a continuous vector field -flow \mathbf{F} . The continuous maximal flow system is described by the following equations:

$$\begin{aligned} \frac{\partial P}{\partial \tau} &= -\nabla \cdot \mathbf{F}, \\ \frac{\partial \mathbf{F}}{\partial \tau} &= -\nabla P, \\ |\mathbf{F}| &\leq g. \end{aligned} \quad (6)$$

Here pressure P is forced to 1 on the source and 0 on the sink images. In the formulas the standard conventions are used. The vectors have three components, for example $\mathbf{F} = (F_1, F_2, F_3)$. The unit vectors along each axis are $\mathbf{i}_k, k = 1, 2, 3$. Also the standard operations with vectors are defined, for example $(x_1, x_2, x_3) + \mathbf{i}_2 = (x_1, x_2 + 1, x_3)$.

The equations (6) are approximated on a staggered grid using the explicit Euler scheme in time $t = \tau n, 0 < n < N$ and space $x_1 = hi, 0 < i < K; x_2 = hj, 0 < j < L; x_3 = hm, 0 < m < M; N, K, L, M = const$. Initial conditions may be chosen as $P = 0$ except at the source and the sink, and $\mathbf{F} = 0$ everywhere. The iteration begins with updating the pressure

$$P_{i,j,m}^{n+1} = P_{i,j,m}^n - \tau(\text{out}\mathbf{F}_{i,j,m}^n - \text{in}\mathbf{F}_{i,j,m}^n) \cdot \mathbf{i}. \quad (7)$$

Then, from (6) the flow \mathbf{H} without constraint will be

$$\begin{aligned} \text{out}\mathbf{H}_{i,j,m}^n \cdot \mathbf{i}_k - \text{out}\mathbf{F}_{i,j,m}^n \cdot \mathbf{i}_k &= \\ &= -\tau(P^{n+1}((i, j, m) + \mathbf{i}_k) - P^{n+1}(i, j, m)), \\ \text{in}\mathbf{H}_{i,j,m}^n \cdot \mathbf{i}_k - \text{in}\mathbf{F}_{i,j,m}^n \cdot \mathbf{i}_k &= \\ &= -\tau(P^{n+1}(i, j, m) - P^{n+1}((i, j, m) - \mathbf{i}_k)), \quad k = 1, 2, 3 \end{aligned} \quad (8)$$

Please note that for $P, \mathbf{F}, \mathbf{H}$: $P^n(i, j, m) \equiv P_{i,j,m}^n$ and $\text{out}\mathbf{F}^n(i, j, m) \cdot \mathbf{i}_k = \text{in}\mathbf{F}^n((i, j, m) + \mathbf{i}_k) \cdot \mathbf{i}_k, k = 1, 2, 3$.

3 Results and Conclusion

The proposed method (3), (6)-(8) was evaluated using a set of 3D CT images which corresponds to stress corrosion cracking in a stainless steel specimen [3] which presents very wide cracks that initiated at the surface of the sample. The specimen has an original cylindrical shape that has been deformed and bent due to the propagation of the cracks. However, the shape distortion is relatively small and the assumption that it is still close to a cylindrical shape is valid. Therefore, the orientation of anisotropic Gaussian filtering which coincides with the longitudinal axis of the cylinder, is acceptable in first approximation. A possible improvement would be to extract the local orientation of the cylinder from a curvilinear skeleton of the sample and uses it to orientate the unidirectional Gaussian filter. From Fig. 2 it is clear that the proposed technique is able to handle the objects broken up to parts - see, for example, the first column.

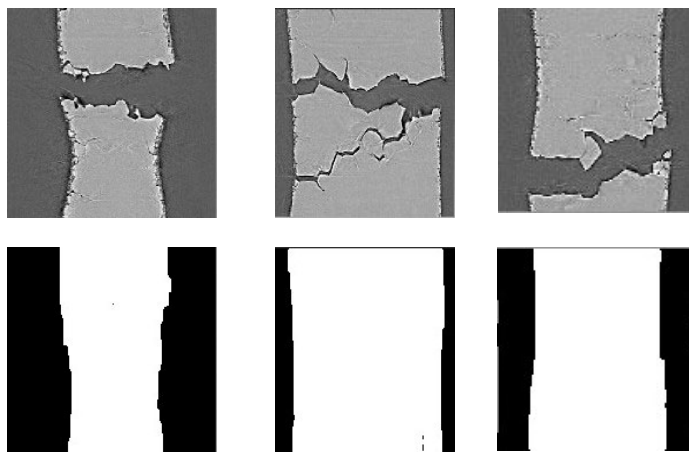


Fig. 2. The restoration of 3D fractured object. The first row is the original images and the second is the result of processing.

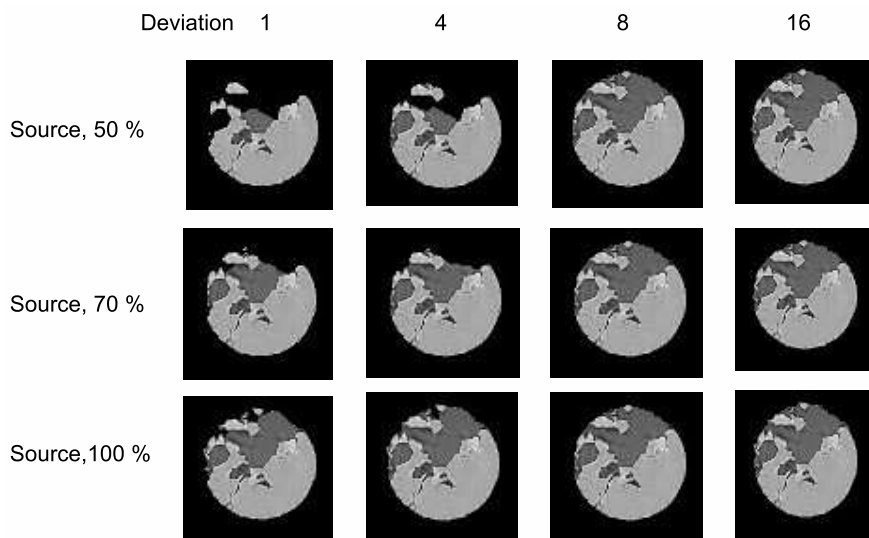


Fig. 3. The dependency of the reconstruction from the value of Gaussian deviation (columns) and the size of source (rows)

The reconstruction of the sample depends on a number of parameters. Our calculation shows that the most important influence is generated by the Gaussian deviation. The second one which also contributes to the desired result is the size of source which is used by the minimum surface method. Fig. 3 shows these dependencies. The original object is depicted in light gray, the environment is

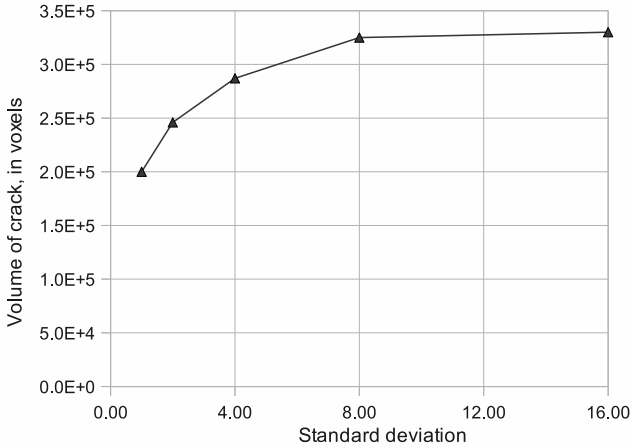


Fig. 4. The volume of crack vs. the standard Gaussian deviation

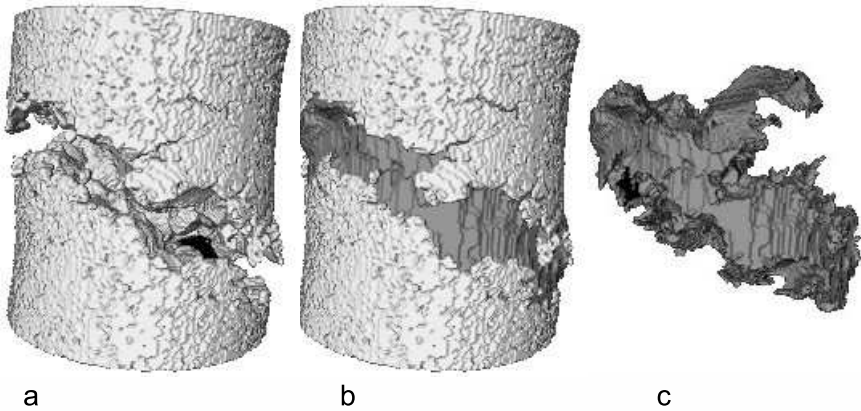


Fig. 5. The three-dimensional view of the restored object: a) the original object, b) the reconstructed object and c) the crack

in black and the results of restoration are in dark gray color. It can be seen that the accuracy of the restoration of the different fractured parts into the 3D object is increased when the deviation is changed from 1 to 8 and the size of source is changed from 50% to 100% of the radius of the original sample. The evolution of the crack volume, measured in voxels, as a function of the standard Gaussian deviation is presented in Fig. 4 for the source of 100% of the radius of the original sample. It can be seen that the calculated volume stabilizes for deviation larger than 8. Another quantitative analysis might be also performed. As the correct shape of the repaired sample is known, one can calculate deviation

of the approximated shape of outer surface from the correct one. Such analysis will be presented in the extended version of the paper.

Finally, the three-dimensional view of one of the restored objects is presented in Fig. 5; source is 100% and $\sigma = 8$. The complementary part of the sample that corresponds to the cracks is in dark gray color. Our method enables to reconstruct the outer surface of the object which presents a large number of discontinuities. The latter one can after be separated from the environment and the object with standard histogram based segmentation (see Fig. 5 c).

The standard approaches for surface restoration like B-spline approximation are clearly inefficient to process objects with many discontinuities. The proposed method doesn't depend on a number of cracks and demonstrates more flexibility. In this context it opens promising directions for research. Future work will address the processing of objects with a higher rate of damage and more complex bent shapes.

Acknowledgements. The work is funded by the European Community's 6th Framework Programme-Marie Curie Transfer of Knowledge Action (DENIDIA, contact No.: MTKD-CT-2006-039546). The work reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein.

References

1. Buffiere, J.Y., Ferrie, E., Proudhon, H., Ludwig, W.: Three-dimensional visualisation of fatigue cracks in metals using high resolution synchrotron X-ray microtomography. *Materials Science and Technology* 22(9), 1019–1024 (2006)
2. Biroscia, S., Buffiere, J.Y., Garcia-Pastor, F.A., Karadge, M., Babout, L., Preuss, M.: Three-dimensional characterization of fatigue cracks in Ti-6246 using X-ray tomography and electron backscatter diffraction. *Acta Materialia* 57(19), 5834–5847 (2009)
3. Babout, L., Marrow, T.J., Engelberg, D., Withers, P.J.: X-ray microtomographic observation of intergranular stress corrosion cracking in sensitised austenitic stainless steel. *Materials Sc. and Technology* 22(9), 1068–1075 (2006)
4. Farin, G.: *Curves and surfaces for CAGD. A practical guide*, 5th edn., 499 p. Academic Press, USA (2002)
5. Eck, M., Hoppe, H.: Automatic reconstruction of B-spline surfaces of arbitrary topological type. In: *Proc. 23rd Ann. Conf. on Comp. Graphics Inter. Tech.*, pp. 325–334 (1996)
6. Stoddart, A.J., Baker, M.S.: Surface reconstruction and compression using multiresolution arbitrary topology G1 continuous splines. In: *Proc. 14th Int. Conf. on Patt. Recogn.*, vol. 1, pp. 788–792 (1998)
7. Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: *Proc. 1st Int. Symp. on 3D Data Proc.*, Padua, Italy, pp. 1–11 (2002)
8. Bae, E., Weickert, J.: Partial differential equations for interpolation and compression of surfaces. In: Dæhlen, M., Floater, M., Lyche, T., Merrien, J.-L., Mørken, K., Schumaker, L.L. (eds.) *MMCS 2008. LNCS*, vol. 5862, pp. 1–14. Springer, Heidelberg (2010)

9. Verdera, J., Bertalmio, M., Sapiro, G.: Inpainting surface holes. In: Proc. Int. Conf. on Image Proc., pp. 903–906 (2003)
10. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proc. of the 4th Eurographics symp. on Geom., SGP 2006, pp. 61–70. Eurographics Assoc., Switzerland (2006)
11. McInerney, T., Terzopoulos, D.: Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis* 1(2), 91–108 (1996)
12. Fan, X., Bazin, P.-L., Bogovic, J., Bai, Y., Prince, J.L.: A multiple geometric deformable model framework for homeomorphic 3D medical image segmentation. In: Proc. Comp. Vis. and Patt. Recogn. Workshops, CVPRW 2008, pp. 1–7 (2008)
13. Keuper, M., Padeken, J., Heun, P., Burkhardt, H., Ronneberger, O.: A 3D active surface model for the accurate segmentation of drosophila Schneider cell nuclei and nucleoli. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.-X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5875, pp. 865–874. Springer, Heidelberg (2009)
14. Shen, T., Li, H., Qian, Z., Huang, X.: Active volume models for 3D medical image segmentation. In: Proc. of CVPR 2009, pp. 1–8 (2009)
15. Jalba, A.C., Roerdink, J.B.T.M.: Efficient surface reconstruction from noisy data using regularized membrane potentials. *IEEE Transaction on image processing* 18(5), 1119–1134 (2009)
16. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. *IEEE Trans. Patt. Anal. and Mach. Intell.* 28(1), 106–118 (2006)
17. Marak, L.: Segmentation with minimum surface. MS thesis, ESIEE, Paris, pp. 1–48 (2008)
18. Geusebroek, J.M., Smeulders, A.W.M., van de Weijer, J.: Fast anisotropic Gauss filtering. *IEEE Transaction on image processing* 12(8), 938–943 (2003)
19. Lampert, C.H., Wirjadi, O.: An optimal nonorthogonal separation of the anisotropic Gaussian convolution filter. *IEEE Transaction on image processing* 15(11), 3502–3514 (2006)
20. Ford, J.L.R., Fulkerson, D.R.: Flows in networks. Princeton University Press, NJ (1962)

Application of the FraDIA Vision Framework for Robotic Purposes

Tomasz Kornuta

Warsaw University of Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19 00-665 Warsaw, Poland
tkornuta@ia.pw.edu.pl
<http://robotics.ia.pw.edu.pl>

Abstract. With computer vision playing more and more important role in robotics, it became clear that a new approach, standardizing the method of vision subsystems development and testing, is required. This paper presents **FraDIA**, a vision framework, that can work as a stand-alone application, as well as a vision subsystem of the **MRROC++** based robotic controllers. The framework description emphasizes its architecture and methods of vision tasks implementation. The diversity of presented robotic tasks utilizing the framework prove its usability.

1 Introduction

Robotic controllers utilizing vision require special submodules, able to aggregate visual data in real-time. The article focuses on the problem of construction of such submodules. The first paper part contains a description of few real-time vision systems and presents an existing trend in robotics for vision subsystems development. The second describes the **FraDIA** vision framework and elaborated approach for vision tasks creation and management. In the last part selected vision-based robotic applications confirming framework usability are presented, without going deep into details of their implementation.

2 Related Work

Machine vision has gained much attention through last few decades, thus multiple computer vision libraries and frameworks were developed. The selection focuses only on frameworks that are presented as versatile enough for real-time vision applications. The Machine Vision Toolkit (**MVT**) [4] is a MATLAB toolkit, providing many functions useful in machine vision and vision-based control, from image acquisition, through preprocessing, to visual Jacobians and camera calibration. The authors claim that utilizing MVT, with input from a camera and output to a robot (not provided), it would be possible to implement

a visual servo system entirely in MATLAB. The **TRIPOD** (Template for Real-time Image Processing Development) [7] is a software framework designed for didactic purposes. It includes interfaces to the video hardware and a graphical user interface to simplify code development. The software provides a pointer to the frame pixel data to enable developer to focus on image processing.

The **ViSP** (Visual Servoing Platform) [3] is a modular software environment featuring independence with respect to the hardware, simplicity, extensibility and portability. Besides algorithms for real-time tracking and vision-based control, ViSP provides also an interface with various classical frame grabbers.

One of the common features of the above are low-level hardware drivers for image acquisition. Their main disadvantage is that the claimed real-time depends mostly on the skills of programmer implementing given vision task and the experience shows that most of self-made implementations of e.g. preprocessing or image segmentation are very inefficient. This is the reason why roboticists started to search and encapsulate in their vision subsystems existing, efficient libraries for computer vision – and **OpenCV** (Open Source Computer Vision Library) [1] is one of the best known, offering enormous number of high-efficient image processing, analysis and recognition algorithms.

Mentioning only few projects utilizing the **OpenCV**, the RoboTIS group is developing a package which provides a toolkit and a set of dedicated to computer vision, OpenCV-based components for the **OROCOS** (Open ROBOT CONTROL Software) [2]. The toolkit includes an image class, defined as a shared pointer on standard OpenCV *IplImage* structures and set of components – drivers for frame grabbers, image and video writers and readers. The **YARP** (Yet Another Robot Platform) [6] contains a *libYARP_sig* module for signal (speech and vision) processing tasks and in case of vision there exists an interface between the *yarpImage* structure and *IplImage*, which enables the usage of OpenCV vision subroutines. The **JAMF** Attention Modeling Framework [8] is a distributed, component-based framework, in which not only images are stored as *IplImage*, but also the whole communication between components is carried out exclusively by passing pointers to instances of OpenCV matrix type *CvMat*, a flexible structure that can represent matrices using any basic numeric type.

The **MRROC++** (Multi-Robot Research Oriented Controller) [9][10] is a distributed robotic framework, facilitating the implementation of multi-robot system controllers. Its usage was successfully verified in multiple diverse robotic tasks, utilizing e.g. visual servoing, force-control or two-handed manipulation. Most of the historical **MRROC++** vision-based applications were written each time from the scratch, thus there was a gap for a specialized vision subsystem, filled by **FraDIA** (Framework for Digital Image Analysis). Besides subroutines for image acquisition and methods for communication with rest of the **MRROC++** processes, the **FraDIA** also imposes a method of vision tasks implementation. It relies on the **OpenCV** and adopts the *IplImage* structure as main image representation. In order to create a scalable and easy to expand graphical user interface (GUI), the **FLTK** (Fast, Light Toolkit) library was used.

3 The Framework Structure

FraDIA is a framework, written mostly in C++ (in an objective manner) and composing of a components library with patterns for their usage. Those patterns impose general implementation method of image processing and analysis algorithms as well as construction of user interface, which might be required for algorithms management and testing. It was important to develop mechanisms which will simplify the process of vision tasks creation, enable switching between different tasks during the application work and, moreover, will offer clear method of their separation. In order to create a general management mechanism dealing with any number of kernels, each of them is encapsulated in a separate dynamic library – shared object (SO) in case of the Linux OS.

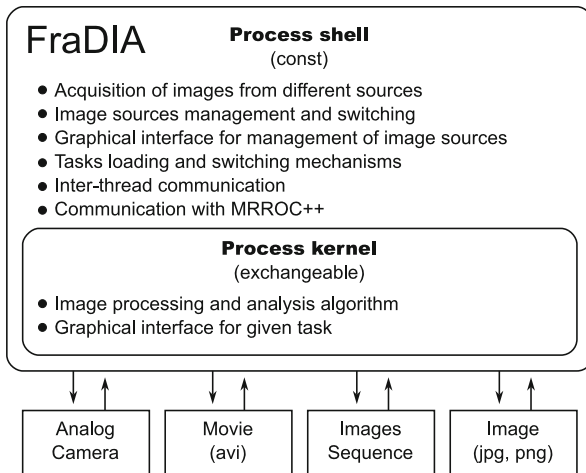


Fig. 1. FraDIA structure

On the other hand, **FraDIA** is an application, executed under the control of the Linux OS, divided into (fig. 1):

- frozen shell, responsible for image acquisition from different sources, tasks switching, communication, etc.
- exchangeable kernel, called **task**, which consists of a set of classes and functions utilized for solution of given vision task (image processing and analysis).

3.1 Kernels

Each kernel (task) consists of two major elements:

- **image processor**, containing image processing and analysis algorithm,

- **task panel**, a part of the graphical interface, through which user can change algorithm parameters, its working modes, etc.

The **tasks manager** is responsible for kernels management – their loading from SOs and switching. All shared libraries, located in the same directory as the **FraDIA** application, are automatically loaded and verified during the application start. The verification process validates the presence of a required interface in the given library – a task interface, consisting of two functions returning image processor and task panel objects. The task manager also creates a menu with successfully loaded tasks names, through which user can switch tasks (currently running task is stopped and the one selected is activated). Tasks can also be switched during the application work if a special command from the **MRROC++** is received. State diagram of a single kernel is presented in fig. 2. Only one task can be in the Running state at given moment and **FraDIA** must successfully load at least one kernel. If multiple kernels were loaded, the first loaded is activated by default. However, there is also an option to pass the default kernel name as a parameter from the command line.

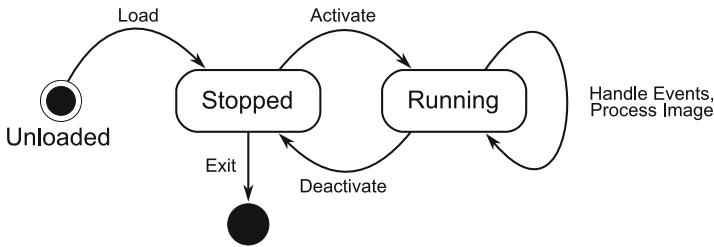


Fig. 2. State diagram of a single kernel

3.2 Application Threads

FraDIA is a multi-threaded application. Major threads elements, as well as the elements belonging to the active task, are presented in fig. 3. Mechanisms of internal (inter-thread) communication and data synchronization are implemented in the process shell, thus are task independent.

The main thread is the Image Acquisition and Processing Thread, responsible for a cyclic retrieving of an image from the source and passing it to the active image processor in order to execute the vision task.

The Interface Thread is responsible for displaying of the graphical user interface and handling user events (e.g. button pressing).

The External Communication Thread is responsible for communication with external processes. From the **MRROC++** point of view **FraDIA** plays a role of a process specialized for aggregation of vision data. The framework is retrieving and processing images independently of the **MRROC++**, simultaneously listening for its commands – if such will come, it will send back the newest results of task execution, e.g. the location of a detected object.

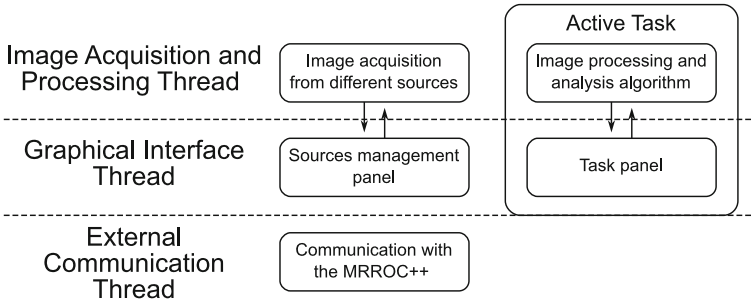


Fig. 3. Diagram presenting **FraDIA** threads and their major elements – arrows represent events (commands) passed between adequate objects in different threads

3.3 Image Sources

It was important to create a possibility to implement and test vision tasks off-line (that is without the access to the vision hardware) and afterwards, when the algorithm will be working properly, to run it (without any additional lines of code) on cameras. Currently there are four different sources of images: V4L-based source responsible for acquisition of images from analog cameras, source retrieving frames from movies, source retrieving images from image sequence and from a single image. The framework also enables recording frames retrieved from camera (as single images or whole movies). All those functions, as well as sources switching, are implemented in the **FraDIA** shell.

4 Selected Applications

4.1 Hand–Eye Calibration

In order to use a gripper-mounted camera (EIH from Eye–In–Hand) for a robotic task, the camera position and orientation with respect to the gripper frame must be known. The problem of determining the relation is known as the hand–eye calibration and it can be resolved by moving the manipulator and observing the resulting camera motion in reference to static object – in this case a chessboard was used (fig. 4a). The whole process was divided into two phases: data acquisition phase and calibration computations phase.

The goal of the first phase was to gather calibration data, consisting of pairs of end-effector poses in two different reference frames (first related to the manipulator base and second related to the chessboard location) from multiple different positions in operational space. The gathered data was later used in the computations phase, in which the relation was determined in a non-linear optimization. In this application **FraDIA** was utilized to determine the location of the camera in the reference frame related to the chessboard.

4.2 Robot–World Calibration

When using a camera statically mounted above the manipulator operational space (SAC from Stand–Alone Camera) transformations from the world frame to the robot base as well as camera frames must be known. This calibration task was also divided into data acquisition and calibration computations phases. The gathered data consisted of end-effector pose in relation to its base and pose of specially marked calibration tile held by manipulator (fig. 4c) in the camera frame. In this case **FraDIA** was utilized for localization of the calibration tile.

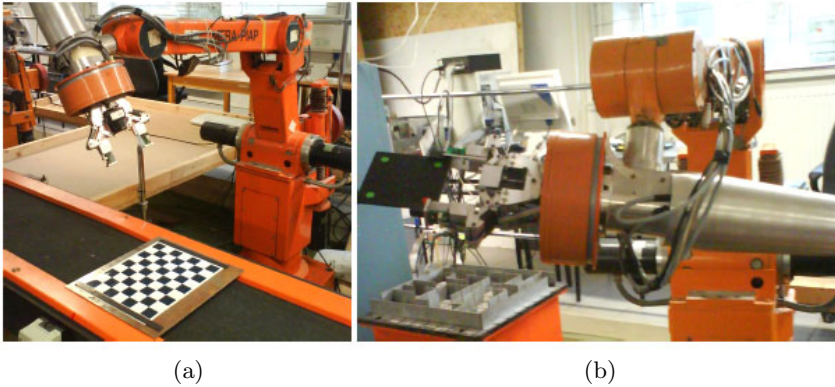


Fig. 4. (a) The Hand–Eye calibration: manipulator over the calibration pattern, (b) The Robot–World calibration: manipulator holding calibration tile in its gripper

4.3 Utilization of the Haar Classifier for Object Grasping

In this task **FraDIA** was used for two purposes: training of a hierarchical Haar classifier and afterwards utilization of the trained classifier for object localization and grasping. Training required preparation of a set of hundreds of pictures with both positive and negative samples, which was partially automated – a movie with moving object of interest was recorded and set as a **FraDIA** source. User had to set regions of interest (ROIs) on few movie frames (e.g. the first and last one). Remaining frames were automatically approximated (ROIs position and size). When user confirmed the correctness of ROIs on every frame, a set of pictures were generated, which was later used as an input for Haar classifier training process. During the object grasping **FraDIA** was switched to task which utilized the trained classifier to locate the objects position and pass it to the **MRROC++**.

4.4 Robot Playing Checkers

In case of robot playing checkers the **FraDIA** was utilized in two different modes and in both the EIH camera was used (fig. 6a). In first mode it had to analyze the

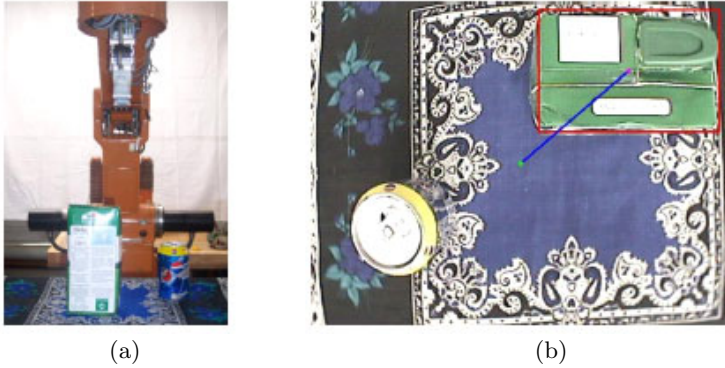


Fig. 5. Utilization of the Haar Classifier for object grasping: (a) The IRp-6 manipulator above the object of interest, (b) – view from the Eye-In-Hand camera

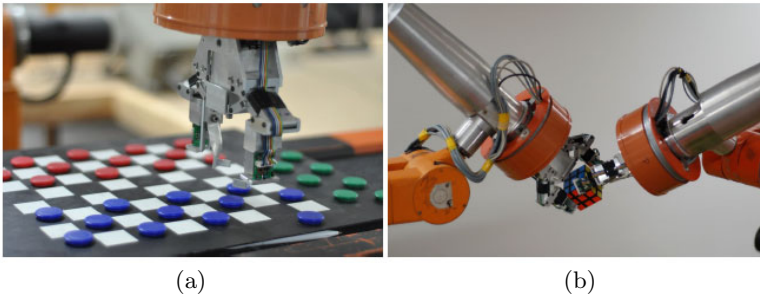


Fig. 6. (a) The IRp-6 playing checkers, (b) Robotic system solving the Rubik's Cube

checkers board in order to monitor game state. If opponent's move was detected, **FraDIA** passed the checkers board state to the **MRROC++**, which searched for the next sub-optimal move ($\alpha - \beta$ pruning). During the pawn grasping (a stage of the move execution) the **FraDIA** task goal was to return the position of selected pawn (utilization of visual servoing).

4.5 Localization and Identification of the Rubik's Cube

The Rubik's Cube solving robotic system consisted of two IRp-6 manipulators, each equipped with an EIH camera and a force sensor [9]. In the primal solution the Cube state was identified by a dedicated subsystem, thus later it was replaced by the algorithm implemented in **FraDIA**. For details please refer to the [5][9].

5 Summary and Future Plans

The paper presented the **FraDIA** vision framework, which strength is the facilitation and standardization of vision subsystems development. The encapsulation

of the **OpenCV** enabled users to utilize its high-efficient subroutines, which resulted in the real-time vision-loop closure in various robotic applications.

Currently **FraDIA** is moving towards the component-based architecture, enabling the creation of processing streams from combination of multiple predefined, lightweight components. Besides others, the goal requires to refactor the existing sources in a component manner and to break the strongly coupled data flow between sources and image processors.

Fast, digital cameras are highly demanded from the robotic point of view (e.g. for visual servoing), thus a new source (an acquisition card driver for digital cameras with Camera Link interface) is being developed. In order to process such amount of data we are also working on the utilization of the GPU (Graphics Processing Units) for low level image processing and feature extraction.

Acknowledgments. Author acknowledges the support of the Polish Ministry of Science and Higher Education grant N514 1287 33.

References

1. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Sebastopol (2008)
2. Bruyninckx, H.: The real-time motion control core of the OROCOS project. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2766–2771 (2003)
3. Marchand, E., Spindler, F., Chaumette, F.: ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine* 12(4), 40–52 (2005)
4. Corke, P.: The Machine Vision Toolbox: a MATLAB toolbox for vision and vision-based control. *IEEE Robotics & Automation Magazine* 12(4), 16–25 (2005)
5. Kasprzak, W., Szykiewicz, W., Czajka, L.: Rubik's cube reconstruction from single view for service robots. *Machine Graphics & Vision* 25, 451–459 (2006)
6. Metta, G., Fitzpatrick, P., Natale, L.: YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems* 3(1), 43–48 (2006)
7. Oh, P.Y.: A Template for Real-time Image Processing Development: Windows-based PCs and webcams for computer vision research and education. *IEEE Robotics & Automation Magazine* 12(4), 65–74 (2005)
8. Steger, J., Wilming, N., Wolfsteller, F., Höning, N., König, P.: The JAMF Attention Modelling Framework. In: Paletta, L., Tsotsos, J.K. (eds.) *Attention in Cognitive Systems*. LNCS, vol. 5395, pp. 153–165. Springer, Heidelberg (2009)
9. Zieliński, C., Szykiewicz, W., Winiarski, T., Staniak, M., Czajewski, W., Kornuta, T.: Rubik's cube as a benchmark validating MRROC++ as an implementation tool for service robot control systems. *Industrial Robot: An International Journal* 34(5), 368–375 (2007)
10. Zieliński, C., Winiarski, T.: Motion Generation in the MRROC++ Robot Programming Framework. *The International Journal of Robotics Research* 29(4), 386–413 (2010)

Video Analysis Based on Mutual Information

Lenka Krulikovská, Michal Mardiak, Juraĵ Pavlovic, and Jaroslav Polec

Department of Telecommunications, Slovak University of Technology
Ilkovicova 3, 812 19 Bratislava, Slovakia
{krulikovska,mardiak,pavlovicj,polec}@ktl.elf.stuba.sk

Abstract. In this paper we present the methods used for the analysis of video based on mutual information. We propose a novel method of abrupt cut detection and a novel objective method for measuring the quality of video. In the field of abrupt cut detection we improve the existing method based on mutual information. The novelty of our method is in combining the motion prediction and the mutual information. Our approach provides higher robustness to object and camera motion. According to the objective method for measuring the quality of video, it is based on calculation the mutual information between the frame from the original sequence and the corresponding frame from the test sequence. We compare results of the proposed method with commonly used objective methods for measuring the video quality. Results show that our method correlates with the standardized method and the distance metric, so it is possible to replace a more complex method with our simpler method.

Keywords: mutual information, shot detection, abrupt cut, video quality, objective method, SSIM, VQM, PSNR, Minkowski-form distance.

1 Introduction

The twentieth century brought many innovations and inventions. One of the most widespread and popular innovations is video in all its variations like cinema, television, videoconference etc. As the number of users, asking for transfer of video increases, the quality of video becomes more important. Providers of video services try to reduce the bandwidth and memory storage required for their services to lower the costs affiliated with video providing. Due to this fact it is important to provide a sufficient quality to satisfy user's requirements for the charge of the video streaming services. Thus the measuring of video quality plays an important role. Recently, the evaluation of video quality has been mostly focused on the quality of service. The quality measurements were often performed by subjective assessment methods which anticipate more directly the viewer's reactions [1]. But subjective methods are time-consuming and administratively difficult (to achieve the standard viewing condition, assessment procedure etc.). Therefore the objective methods are used to correlate with the results of subjective methods [1,2,3,4].

We present a novel objective method for evaluation quality of video based on calculating the mutual information and compare this method with commonly used objective methods.

Progress in the multimedia compression technology and computer performance has led to the widespread availability of digital video. The way how to find an exact video is a very important field of the research. The basic element of video sequences is a shot. The shot is defined as a sequence of consecutive frames caught by one camera in single time and action. Video shot boundary detection is the first step toward the high level video content analysis. Two shots can be combined by an abrupt cut or a gradual transition. The abrupt cut is an instantaneous transition of the content from one shot to another without any special effects. It is a direct change of shots. The gradual transition is a transition where special effects are used to combine two shots artificially.

Different approaches have been proposed to extract shots. The major techniques used for the shot boundary detection are pixel differences, statistical differences, histogram comparisons [5], edge differences, compression differences and motion vectors [6,7,8]. In general, abrupt transitions are much more common than gradual transitions, accounting for over 99% of all transitions found in a video [9]. Therefore, the correct detection of abrupt shot transitions is a very important task in the video segmentation and this paper is only focused on the detection of an abrupt cut. Our aim was to improve the existing method proposed by Cernekova [10]. This method uses the mutual information, but it is very sensitive to object or camera motion. We propose a method for the abrupt cut detection, which combines the mutual information and the motion prediction. The results were verified with RGB and YUV format and compared with the results of the original method.

2 Mutual Information

The mutual information measures the amount of information about random variable X conveyed to random variable Y . For a video it is calculated for two successive frames for each RGB or YUV component separately [10]. For the calculation we need matrices $C_{t,t+1}^K$, c_t^K and c_{t+1}^K where K indicates RGB or YUV component. We consider the video sequence with pixel intensity in the range from 0 to $N - 1$. $N \times N$ matrix $C_{t,t+1}^K$ carries the transition of the pixel intensity between the frames f_t and f_{t+1} . Each matrix element $C_{t,t+1}^K(i, j)$ shows a number of pixels with the intensity i in the frame f_t , which changes to the intensity level j in the frame f_{t+1} divided by the number of pixels in the frame. Matrices c_t^K and c_{t+1}^K have dimension $1 \times N$ and their elements $c_t^K(i)$ or $c_{t+1}^K(j)$ show the total number of pixels with the intensity level i in the frame f_t , respectively with the level j in the frame f_{t+1} divided by number of pixels in the frame. The mutual information $I_{t,t+1}^K$ for each RGB or YUV component is expressed by [10]:

$$I_{t,t+1}^K = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{t,t+1}^K(i, j) \ln \frac{C_{t,t+1}^K(i, j)}{c_t^K(i) c_{t+1}^K(j)}. \quad (1)$$

And the total mutual information for RGB format is given by:

$$I_{t,t+1}^{RGB} = I_{t,t+1}^R + I_{t,t+1}^G + I_{t,t+1}^B . \quad (2)$$

And for YUV format:

$$I_{t,t+1}^{YUV} = I_{t,t+1}^Y + I_{t,t+1}^U + I_{t,t+1}^V . \quad (3)$$

3 Abrupt Cut Detection

We used the approach proposed by Cernekova [10] based on the mutual information. The novelty of our method is in combining the mutual information and the motion prediction. At first, we calculated the motion prediction for each frame. We used the standard hierarchical motion estimation for frame-based compression with the value of maximum displacement 15. Then the mutual information is calculated. Small value of the mutual information $I_{t,t+1}$ indicates an abrupt cut. We confirmed the effectiveness of our method through a test experiment. For test purposes we created a video sequence at QCIF resolution (176 x 144 pixels) with 3 abrupt cuts. The test video sequence consists of parts of four standard test sequences coastguard_qcif, hall_qcif, miss-america_qcif and suzie_qcif. Fig. 1 displays one pair of frames of the test sequence without (for the original method computation) and with the motion prediction (for computation of the proposed novel method).

The values of the mutual information are determined for each component for both format Y, U, V, R, G and B and subsequently total value for YUV and RGB. The result of the abrupt cut detection for RGB and YUV format is shown on Fig. 2. The first graph is for RGB, the second one presents YUV



Fig. 1. Example of frames in Suzie sequence: (a) without motion prediction and (b) with motion prediction

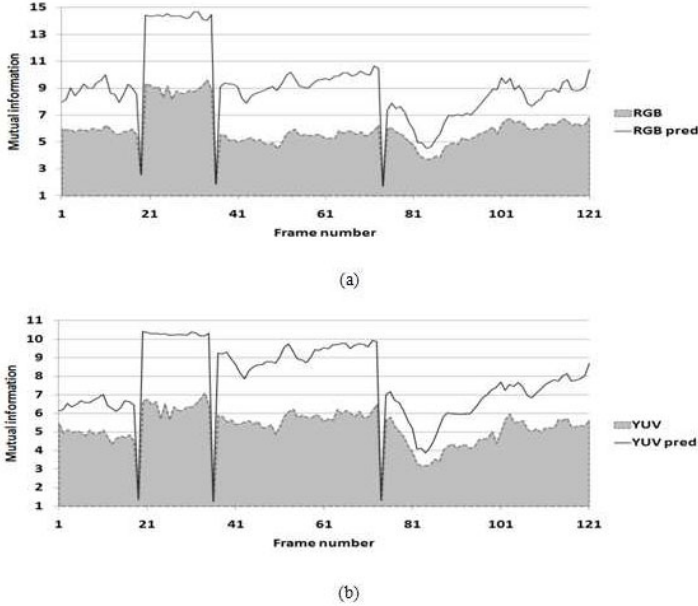


Fig. 2. Plot of total mutual information for: (a) RGB format and (b) YUV format

format. Each graph displays results for the original method and for our improved method (marked with the word pred).

Three shots were detected by both methods; they are represented by a significant decrease in the value of the mutual information. The values of the mutual information for abrupt cuts are the same for the original and our method, but our method with the motion prediction provides a higher rate for all other frames. If we use threshold for the abrupt cut detection, we can set higher values for threshold for the proposed method without risking a false detection. In comparison of the results for RGB and YUV format, YUV gives a smoother graph. It is seen for example at the intervals between the frames 21 and 41 or between the frames 75 and 80.

The next step how to compare obtained results and find another method for the cut detection was to evaluate the difference between the successive pairs of the calculated mutual information. Similar approach was proposed in [11] for the technique of the gradual transition detection. The difference between the pair of the mutual information of successive frames $I_{t,t+i+1}^K$ and $I_{t,t+i+2}^K$ is expressed by:

$$diff_{t+i,t+i+1}^K = | I_{t,t+i+1}^K - I_{t,t+i+2}^K | . \quad (4)$$

With effort to provide as fair comparison of the original and the proposed method as possible, we normalized all values of the computed difference to the maximum value 1. In other words, we found the value of the global maximum for each difference function and divided every its value by the found maximum.

After this step the functions of difference are normalized and can obtain values only in the range from 0 to 1. An example of the normalized difference is presented on Fig. 3 for both tested formats (also an example of threshold is displayed). We can see that the graph for RGB is wavier than the graph for YUV format. According to the method comparison, the proposed method gives better results - it is noticeable at frame number 25, where the value of the normalized difference is much higher for the original method. If we set the threshold value to 0,14 for YUV format or 0,12 for RGB format, it would cause a false detection of the fourth abrupt cut (between frames 25 and 30) in the video sequence for the original method. The proposed method would detect only three existing cuts and avoid false detection for this value of threshold.

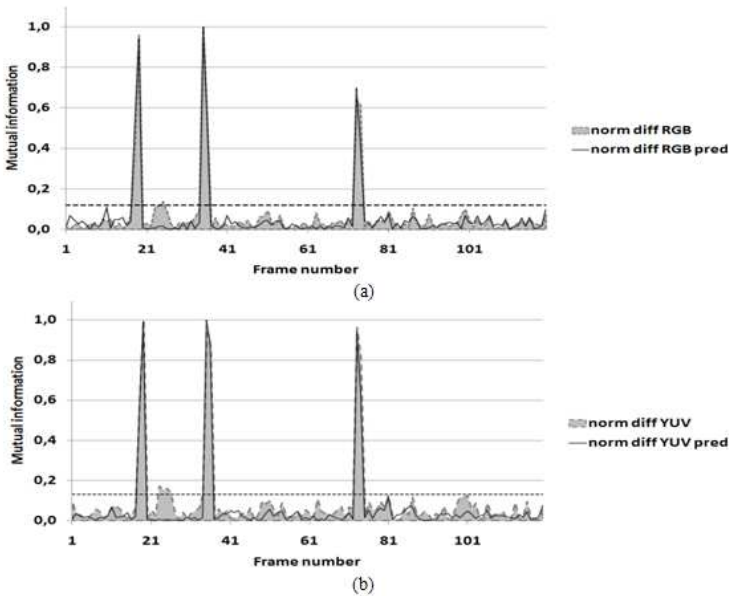


Fig. 3. Plot of normalized difference for: (a) RGB format and (b) YUV format

4 Evaluation of Video Quality

Another way how to use mutual information for video analyses is to evaluate the quality of video. In our method we calculated the mutual information between two frames, a frame from the original sequence f_x and the corresponding frame from the test sequence f_y , where x and y is an ordinal number of the actual frame. Thus we meditate only cases where $x = y$. To verify the relevance, we compared the proposed method with *Peak Signal-to-Noise Ratio* [1], *Video Quality Metric* [12], *Structural Similarity Index* [13,14] and *Minkowski-form distance* with $r=3$ [15] as representatives of commonly used objective methods.

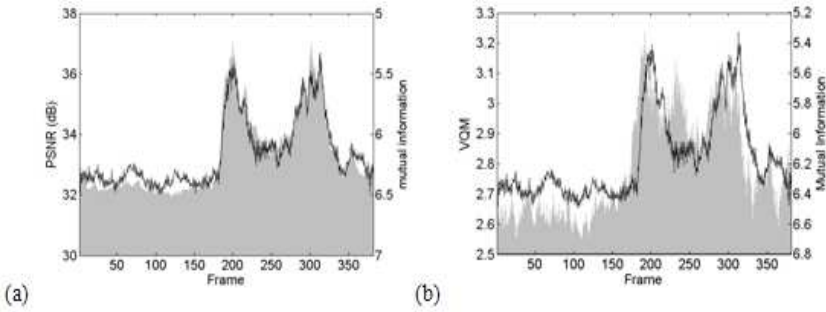


Fig. 4. Comparison between: (a) PSNR in dB and mutual information, left Y-axis represents PSNR in decibels (gray curve), right Y-axis represents mutual information (black curve) and (b) VQM and mutual information, left Y-axis represents VQM (gray curve), right Y-axis represents mutual information (black curve)

Fig. 4 shows amount of the mutual information for each frame in comparison of PSNR. Note that Y-axis for the mutual information has descending order, thus the run of both metrics are similar but turned vice versa. It means that the increasing of the PSNR means the fall of the mutual information or in other words, when two frames become more different, they are more independent, PSNR is growing and the mutual information is decreasing.

The comparison of the standardized metric VQM and the mutual information between the original and the test sequence is shown on Fig. 4. The trend of both curves is very similar but turned vice versa (Y-axis for the mutual information has opposite order) same as for PSNR. When the frame from the original and the test sequence differ in more pixels (due to noise, etc.), the quality is worse, VQM is rising and the mutual information is falling down. Mutual information peaks occur slightly behind the VQM peaks. Just one of the VQM peaks which appear in the frame 236 does not correspond with the peak of mutual information.

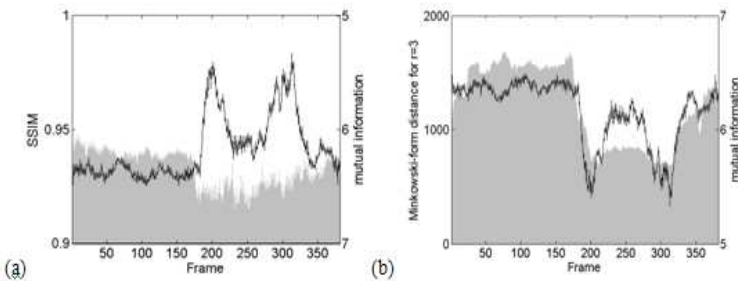


Fig. 5. Comparison between: (a) SSIM and mutual information, left Y-axis represents SSIM (gray curve), right Y-axis represents mutual information (black curve) and (b) Mikowski-form distance and mutual information, left Y-axis represents Mikowski-form distance (gray curve), right Y-axis represents mutual information (black curve)

Fig. 5 shows the runs of SSIM metric in gray and the mutual information in black. Values of both curves differ in many frames, rising of the mutual information does not correspond with an apparent equivalent change of SSIM run (rising or falling down). So the trend of both metrics is not similar as in cases of PSNR and VQM metrics. Results for comparing the mutual information and the Mikowski-form distance are shown on Fig. 5. Trends of both metrics are similar. Rising of the Minkowski-form distance, means rising of the mutual information and its fallings means falling of the Minkowski-form distance. Mutual information peaks are sharper thus the values are higher but they occur almost at the same frames as peaks for Minkowski-form distance (mutual information peaks appear few frames later).

5 Conclusion

In this paper we have proposed novel methods for video analysis: a method of the abrupt cut detection and an objective method for measuring the video quality. The novelty of this abrupt cut detection method is in combining the motion prediction and the mutual information. We confirmed the effectiveness of our method by a test experiment and compared the obtain result with a method based only on the mutual information. The proposed method gives better results. It provides higher robustness to object and camera motion and avoids false cut detection.

The advantage of our method for measuring the video quality is in its easy computation and that it does not have any special demanding. We compared this method with four metrics: Peak Signal-to-Noise Ratio, VQM, Structural Similarity Index and Minkowski-form distance metric. Results show that the mutual information is suitable for measuring the quality of video and can replace VQM or PSNR metric.

For future work, we would like to propose a method of the abrupt cut detection from the view of coder logic, where the predicted frame will be compared with the previous frame from the original sequence without the motion prediction. According to the method for measuring the quality of video, we want to run a more complex set of experiments with different video sequences to prove the relevance of the proposed method.

Acknowledgments. Research described in the paper was financially supported by the Slovak Research Grant Agency (VEGA) under grant No. 1/0883/08.

References

1. ITU-R BT.500-11, Recommendation ITU-R (2002)
2. Winkler, S.: Digital video quality vision model and metrics. John Wiley & Sons Ltd., Chichester (2005)
3. Kawayoke, Y., Horita, Y.: NR objective continuous video quality assessment model based on frame quality measure. University of Toyama, Japan

4. Ries, M., Nemethova, O., Rupp, M.: Video quality estimation for mobile H.264/AVC video streaming. *Journal of communication* 3, 41–50 (2008)
5. Amiri, A., Fathy, M.: Video shot boundary detection using QR-decomposition and Gaussian transition detection. *EURASIP Journal on Advances in Signal Processing* 2009, Article ID 509438
6. Hanjalic, A.: Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology* 12(2), 90–105 (2002)
7. Boreczky, J.S., Rowe, L.A.: Comparison of video shot boundary detection techniques. In: *Proc. SPIE in Storage and Retrieval for Still Image and Video Databases IV*, vol. 2664, pp. 170–179 (January 1996)
8. Lienhart, R.: Comparison of automatic shot boundary detection algorithms. In: *Proceedings of SPIE in Storage and Retrieval for Image and Video Databases VII*, San Jose, Ca, USA, vol. 3656, pp. 290–301 (January 1999)
9. Paschalakis, S., Simmons, D.: Detection of gradual transitions in video sequences (April 24, 2008), <http://www.wipo.int/pctdb/en/wo.jsp?W0=2008046748>
10. Cernekova, Z.: Temporal video segmentation and video summarization, Ph.D. dissertation, Dept. App. Inf., Comenius Univ., Bratislava, SK (2009)
11. Zhao, H., Li, X., Yu, L.: Shot boundary detection based on mutual information and canny edge detector. In: *Proc. 2008 International Conference on Computer Science and Software Engineering* (2008)
12. Xiao, F.: DCT-based Video Quality Evaluation, MSU Graphics and Media Lab. (Video Group) (Winter 2000)
13. Wang, Y.: Survey of objective video quality measurements, EMC Corp., Hopkinton, MA, Tech. Rep. WPI-CS-TR-06-02 (2006)
14. Dosselmann, R., Yang, D.X.: A Formal Assessment of the Structural Similarity Index, Technical Report TR-CS 2008-2. University of Regina, Regina (2008)
15. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2), 99–121 (2000)

Speeding Up Powerful State-of-the-Art Restoration Methods with Modern Graphics Processors

Tim Kubertschak¹, Rico Nestler²,
Torsten Machleidt², and Karl-Heinz Franke²

¹ Centre of Image and Signal Processing (ZBS) Ilmenau
Gustav-Kirchhoff-Str. 5, 98693 Ilmenau, Germany
tim.kubertschak@zbs-ilmenau.de

² Ilmenau University of Technology, Computer Graphics Group

Abstract. One important aspect of digital image processing is the removal of glitches from the measured data, particularly from observations of physical phenomena. We propose an approach which realises valid results that have nearly no restoration artefacts. It is based on a further developed state of the art regularisation principle - the restriction of the degrees of freedom of the solution-describing model. The key idea is to use parameterised image elements instead of single pixels which are determined jointly in a bayesian estimation. However, the long duration of a restoration using such an approach is a problem in many applications. This article presents a technique how to speed up this method and reduce the runtime using the example of restoration of kelvin probe force microscopy-data.

Keywords: restoration, regularisation, deconvolution, degree of freedom, GPU.

1 Introduction

Generally, a goal of digital image processing is to gain information from two-dimensional measured data. It turns out to be a problem that usually all data is blurry and noisy due to the measuring-process. In order to retrieve meaningful information from the data the corruptions must be removed as well as a possible lack of information must be compensated.

For this purpose a variety of methods have been developed. These methods reach from simple linear deconvolution techniques through bayesian approaches. The latter try to find the real data in an iterative, well-posed process. By estimating the cause of the data, these methods try to maximise the likelihood between measured and real data using a predefined imaging model. In addition, these methods allow to incorporate as many a-priori knowledge of a suitable solution into the restoration process as can be expressed mathematically.

However, the majority of the known methods carry out the restoration of the measured data using a given pixelgrid. Every pixel corresponds to one degree-of-freedom that has to be determined. In real situations the distribution of information

doesn't have to fit the given grid. There may be cells that contain too few information as well as cells that contain a lot of information. As a result the estimation may overfit or underfit the data locally, which leads to undesired artefacts.

Hence, an improved, modern approach is a method that restricts the possible dof's to a minimum required number. The dof-restricting approach tries to find the underlying cause of the data by estimating both, the undisturbed data as well as a model that describes the appearance of the data. Unfortunately the dof-restricting method suffers from two problems. First, it is very hard to find parameter settings that lead to good results. Second, this approach needs a very long time until the restoration is done and a result is available. It is therefore not possible to easily retrieve the best parameter settings in a heuristic way.

To speedup the method it is executed in parallel. With today's graphics adapters (GPU) and their highly parallel architecture it is possible to swap out expensive calculations and process them in parallel on the GPU. This article presents the prerequisites for using the described acceleration as well as the results of the restoration of nanoscaled data obtained via kelvin probe force microscopy (KPFM) using a dof-restricting approach.

2 Regularised Approaches for KPFM-Measured Data Deconvolution

As shown in [1] deterministic (ideal) measured KPFM-data is linearly connected to the real potential by a convolution. Like in most real world systems the data is corrupted by mostly stochastic measurement uncertainties. Besides, some naturally deterministic factors, e.g. random line offsets, are also put in the same class. In our case it could be shown that each measured KPFM data point contains the real unknown signal value and an additive signal-independent random value. Related to the two-dimensional data domain of the measured data the overall noise process may be considered as pointwise stationary, normal distributed and without any lateral dependency [1]. Thus, our data formation model will be completed by an appropriate noise term N (see figure [1]).

The estimation of the measured data underlying unknown physical cause $\tilde{\phi}$ can be found in our case by deconvolution. Thereby, formally the following operation takes place.

$$\tilde{\phi} = \mathcal{KPFM}^{-1}\{\phi_{\text{Mess},n}\} \in \Phi \rightarrow \Phi = \{\tilde{\phi} \mid \mathcal{KPFM}\{\phi\} = \phi_{\text{Mess}}\}. \quad (1)$$

For that reason a reliable knowledge of the deterministic parts of the data formation model, the system operator $\mathcal{KPFM}\{\}$, is absolutely necessary.

Due to the incomplete information within the data and the corruption of the measured data the resulting solution set of the deconvolution operation Φ comprises a big amount of possible solutions. Vica versa, each element of this solution set acts from the view of the system operator as an equivalent indiscernible single solution.

Because of the loss of information within the measured data the selection of a solution could not be done solely from the data. The discrepancy between

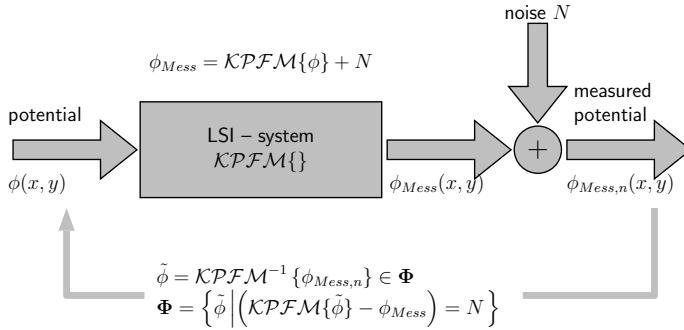


Fig. 1. Linear model for the data degradation in KPFM-mode and its use for the data restoration process

the number of degrees-of-freedom used in the restoration and the corrupted and incomplete information in the data is usually denoted as an *ill-posed* problem. If disregarded, the solution will be arbitrary and the result which is fully covered with restoration artefacts is not suitable for further data processing or evaluation. A common approach to overcome these inherent problems is to put some well-posed elements into the procedures to solve this known ill-posed inverse problem. The so called regularisation introduce knowledge about a plausible solution into the restoration process. So a selection from the solution set must be done in the sense of both, the needs of the data **and** the needs of the available knowledge. Besides, the regularisation give us more stability within the restoration progress.

In our case, the restoration of recorded physical phenomena, we have to highly stress the selection of the used method. A universal and minimal restrictive approach for the problem regularisation will be given by a adaptively restriction of the degrees-of-freedom (dof) which form a possible solution.

2.1 Regularisation by a Dof-restricted Approach

The regularisation by a dof-restricted solution belongs to the class of modern und most powerful approaches in the field of image data restoration. The origins of this method and the ideas of using multiscaled and multiformed image elements (pixons) for the composition of a solution are given in [2]. This approach was already adapted and applied sucessfully by the authors in the field of astronomical image restoration in the past [3].

Basically, the key idea is to restrict the dof's of the solution composition in that kind to realise a solution which shows both a minimal complexity (minimal dofs) **and** a sufficient ability to reproduce all deterministic parts of the measured data (the so called minimal goodness-of-fit). Because the need for restriction is different in various regions of the data, a pixon-based image model will be used which consists of elements of diffent shape and size (according to [2] so called macro pixels or pixons). The selection of the appropriate image elements take

place locally, adapted to the needs of both the measured data and the deducible structure of the solution (see below). This kind of a data-adaptive solution composition is particularly suitable for inhomogeneous structured data underlying signals. Restoring artefacts as a result of a missing restriction of the dofs will be suppressed automatically because the complexity of the needed composition must not be increased without any further benefit in the goodness-of-fit. Despite of its potential, this approach is minimal restrictive in comparison to other known approaches which use primitives to be combined in solution as well.

A wider application of this advanced restoration method in the field of the scientific data analysis is opposed to the typical runtime to realise such solutions so far. A practicable answer could be to restrict the amount and variety of image elements, e.g. use only isotropic elements with rough increments as used in [3]. On the other side, doing this restricts the applicability and the full potential of the method could not be tapped. The application of this method starts usually with a specific parameterisation. Until today this is a heuristic process which is done based on test runs on preselected data parts. This process is also affected negatively by the significant runtime.

Our current application could vastly benefit if a more comprehensive image element basis with isotropic and anisotropic elements is available [1]. This comes to multivariate image element models which bring severe requirements to the formulation and implementation of an adapted pixon-based restoration procedure. If we realise a procedure which is easy to handle and results in solutions with acceptable runtimes a new quality will be achieved.

2.2 Some More Details of the Proposed Dof-restricted Restoration with Multivariate Image Elements

Our proposed method / algorithm is characterised by a solution composition in a two-dimensional discrete domain based on two separate parts, the so called pixelwise *signal-correlation*-pairs. In the so called fuzzy correlation approach adjacent image elements share some of each other's signal instead of having hard boundaries. The first component, the signal part \mathbf{S}^ϕ , is a two-dimensional scalar field. It contains the signals that are related to each image element. The second component, the correlation part \mathbf{C}^ϕ , is generally a multidimensional field with as many dimensions as the amount of parameters of the multivariate image element basis. A solution $\tilde{\phi}$ is realised by local convolution based on the signal-correlation-pairs. In the practical case of a shape-description by an image element feature vector $\underline{\delta}_i$ at position i follows

$$\tilde{\phi}(x, y) \rightarrow \tilde{\phi}_{i_x \cdot \Delta x, i_y \cdot \Delta y} = \tilde{\phi}_i \quad (2)$$

$$\tilde{\phi}_i = (\tilde{\mathbf{S}}^\phi * \tilde{c}(\underline{\delta}_i))_i = \sum_j \tilde{s}_j \cdot \tilde{c}(\underline{\delta}_i)_{i-j}. \quad (3)$$

We propose a feature vector which is composed of the scale, the eccentricity and the rotation angle. The solution component \mathbf{S}^ϕ is in principle most widely

uncorrelated. Hence, it is possible to use additional regularisation techniques which require such a signal feature, e.g. maximum entropy.

To find an estimation of both solution components we use a bayesian estimation procedure. There the joint probability $p(\tilde{\mathbf{S}}^\phi, \tilde{\mathbf{C}}^\phi \mid \phi_{\text{Mess},n})$ will be maximised in a cyclic way until some criterion is reached. Attending to our bayesian view onto the restoration problem the role and interpretation of the components of the resulting cost functions as posterior, likelihood and prior are defined. To restrict the amount of solution parameters that have to be estimated and to decrease the complexity of the cost functions we do an optimisation based on marginal distributions, meaning in each cycle each solution component is found separately one after another whereby the other is held as a constant. This procedure is also advantageous because the stationary solution component can be used to control an additional regularisation of the others. Hence, one cycle t consists of two steps:

Step 1: Image model (parameter) estimation $\tilde{\mathbf{C}}^{\phi_t}$ - local correlation part of the solution ($\mathbf{S}^{\phi_0} = \tilde{\mathbf{S}}^{\phi_{t-1}}$)

Step 2: Estimation of the signal part $\tilde{\mathbf{S}}^{\phi_t}$ of the solution ($\mathbf{C}^{\phi_0} = \tilde{\mathbf{C}}^{\phi_t}$)

A particular importance belongs to the prior of step 1. To determine this prior the effective degree-of-freedom of a single image element $\mathbf{eDOF}(\underline{\delta})$ is proposed [1]. This definition is valid for any case of a general multivariate parameterised image element $\mathbf{c}(\underline{\delta})$ as shown in formula [3]. The \mathbf{eDOF} can be calculated easily from its parameter vector $\underline{\delta}$. The independent single pixel is the upper bound which realises an \mathbf{eDOF} -value of $\mathbf{eDOF}(\underline{\delta}_1) = 1.0$. The amount of all effective dof's builds the overall degree-of-freedom of a so called dof-restricted solution \mathbf{DOF} . Its maximum value, the amount of image pixels, marks the \mathbf{DOF} of the measured data. This resulting heuristic prior increases if the overall degree-of-freedom of the solution decreases. The regularisation constant which weights the role of the prior related to the data driven likelihood can be determined adaptively based on the local goodness-of-fit at the current point $\tilde{\phi}$. To estimate this constant an evidence estimation scheme has to be installed. At the moment we prefer the first option instead.

$$p(\tilde{\mathbf{C}}^\phi(\underline{\delta}), S^{\phi_0}) = \prod_i e^{-\alpha(\tilde{\phi}) \cdot \mathbf{eDOF}(\underline{\delta}_i)} = e^{-\sum_i \alpha(\tilde{\phi}) \cdot \mathbf{eDOF}(\underline{\delta}_i)}. \quad (4)$$

The estimation of the parameters that optimise the different cost functions at each sub-step is done by using a conjugated gradient method. In the case of the proposed image elements all parts to make the optimisation possible can be calculated by local convolution with precalculated kernels of image elements or its partial derivatives. These local convolutions are efficiently done by using a fast fourier transform (FFT). Former investigations show that a main part of the overall runtime is required to do a lot, meaning normally several 100000's, of these FFT-operations. This overall runtime can be essential affected by the

kind of how the fast fourier transform is done. In our view this will be a crucial point of the further actions.

3 Speeding Up with CUDA

The compute unified device architecture (CUDA) is a concept designed by NVIDIA to use the graphics adapter for general purpose calculations. The advantages of using the GPU are an increased performance compared to the CPU and the capability to work in a highly parallel manner. It is therefore possible to greatly speed up problems or parts of them, if they can be parallelised in a SIMD-fashion. That is, the same instruction can be carried out on different parts of data.

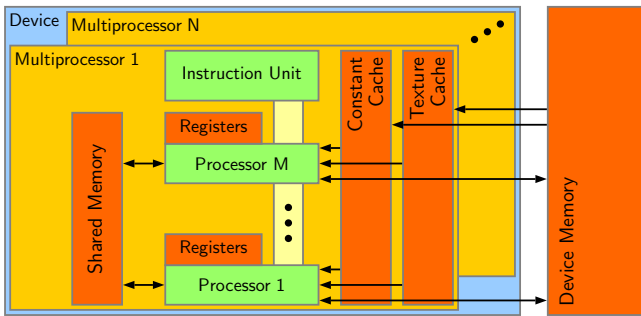


Fig. 2. Physical structure of a CUDA-capable device (according to [4])

The enormous acceleration is possible due to a special architecture of CUDA-capable devices ([4]). As depicted in figure 2 the graphics adapter consists of a certain amount of multiprocessors that consists of a set of processors. Every processor is able to execute a single instruction on a different part of the data. The resulting speedup of an algorithm is of the same order of magnitude as the number of simultaneously processible data.

Another advantage of CUDA is the ease of programming. The term CUDA does not only describe a way how to build hardware. Moreover it is the name of an interface to easily write programs that run on the GPU. Therefore, NVIDIA provides the programmer within its CUDA API with a package of valueable tools. Besides two useful libraries for calculation of the FFT and matrix-operations a simple extension to the C-programming language is provided.

4 Results

For accelerating an algorithm with CUDA it has to fulfill certain conditions. The algorithm or at least parts of it must be capable of being executed in parallel.

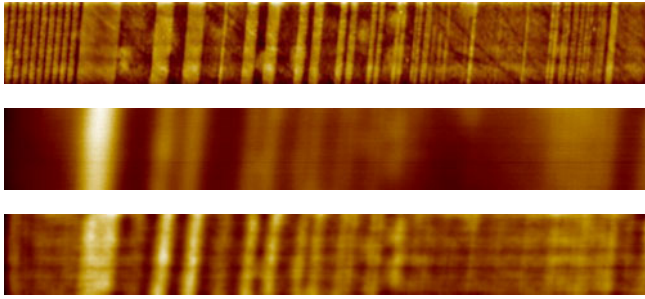


Fig. 3. Topography measured with AFM (top), measured potential (center) and reconstructed potential (bottom) of the BAM-L200 standard

That means there have to be parts in the algorithm where single instructions are applied to different portions of the data. Fortunately, the restoration algorithm include parts that can easily be parallelised.

As noted in section 2, one part of the estimation of the real image is the calculation of various convolutions and thus utilising the FFT algorithm. For instance, the reconstruction of the KPFM-data depicted in figure 3 requires more than 55.000 FFTs to be computed. Executing the FFT in parallel would yield in a neat acceleration of the algorithm. Fortunately, the FFT is very suitable for parallelisation through its recursive structure. With this knowledge, the restoration algorithm was modified to use CUDA for its expensive FFT calculations.

First results of the modified algorithm are illustrated in figure 3. It shows the topography, the measured potential and the reconstructed data of the nanoscale sample BAM-L200 with a size of $1200\text{nm} \times 150\text{nm}$ (1024×128 pixel). The results are quite good, although just a simple set of image elements was used. As depicted, the restored data has an improved resolution over the measured one. As the comparison with the topography shows, there is still room for further improvement, even though one must take into account that both images are of different underlying physical cause.

However, the main reason for an improvement with CUDA was the long runtime of the algorithm on the CPU. This goal could be reached. The restoration shown in figure 3 required only 12 minutes and 30 seconds on a NVIDIA Tesla C870 GPU. Compared to a similar implementation for the CPU, this is an acceleration of approximately factor four. Similar results could be achieved with other samples and correspond to those in 5.

Additionally, it has to be mentioned, that the factor of acceleration depends on the data itself. With a growing image-size, the speedup increases until the size reaches a certain boundary. With a larger amount of data more pixels can be dealt with in the same period of time. A similar behavior was observed in 5 and 6 for the parallel execution of the FFT. Thus, with a larger amount of data a better speedup may be expected although there is a hardware-side limitation through transferrates of the pci-express bus.

5 Conclusion

This article presented a method how to accelerate the lengthy runtime of a state-of-the-art dof-restricting restoration approach along with its possible speedup. As demonstrated, the utilisation of the graphics adapter with CUDA is a feasible way for accelerating many algorithms in digital image processing as long as they are capable of being executed in parallel.

The accelerated algorithm will be the starting point for further improvements. For instance, it is now possible to work with an enlarged set of data in a passable amount of time. Furthermore it is possible to ease the heuristic determination of optimal parameter settings of the presented restoration algorithm and thus improve the quality of the reconstruction.

For further acceleration of the algorithm, more effort will be made. The algorithm will be examined to find other parts that are adequate for parallelisation. It is planned to fully implement the algorithm for the graphics adapter, hence utilising its full potential.

Our next task in the reconstruction of kelvin probe force microscopy-data will be the enhancement of the reconstruction-quality. We will consider two things to achieve this: First, the parameterisation will be improved. Second, we will examine image elements that are more complex and thus are more suitable for the expected deterministic nanoscaled structures. Both are greatly simplified through the revised algorithm.

Acknowledgement. This work was supported by the German Science Foundation (DFG, SFB 622). The authors wish to thank all those colleagues at the Ilmenau University of Technology and the Centre of Image and Signal Processing Ilmenau, who have contributed to these developments.

References

1. Nestler, R., Machleidt, T., Franke, K.H., Sparrer, E.: Dof-restricted deconvolution of measured data from afm special modes. *Technisches Messen* 75(10), 547–554 (2008)
2. Puetter, R.C.: Information, pixon-based bayesian image reconstruction. In: *Digital Image Recovery and Synthesis III*. Proceedings of S.P.I.E, vol. 2827, pp. 12–31 (1996)
3. Nestler, R., Franke, K.H., Hiltner, P., Kroll, P.: Restoration of digitized astronomical plates with the pixon method. In: Payne, H.E., Jedrzejewski, R.I., Hook, R.N. (eds.) *Astronomical Data Analysis Software and Systems XII*. ASP Conference, vol. 295, p. 407 (2003)
4. NVIDIA Corporation: *CUDA Programming Guide*. 2.1 edn. (2008)
5. Kubertschak, T.: *Anwendung moderner Bildverarbeitungsmethoden (Pixonenmethode) zur Restauration von Nanomessdaten*, Thesis (2008)
6. Heist, T., Schmid, U., Osten, W.: Schnelle Berechnung zweidimensionaler Fouri-ertransformationen mittels Grafikkarten. In: *VDI-Berichte*, vol. 1981, pp. 217–223 (2007)

Multi-layered Framebuffer Condensation: The l-buffer Concept

Jarosław Konrad Lipowski

Institute of Computer Science, Computer Graphics Laboratory
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
J.Lipowski@ii.pw.edu.pl

Abstract. In this paper we present a method that can be used with algorithms requiring the presence of multiple data elements at the last step of z-buffer based rasterization pipeline. The work presented here-in gives a ground for solutions to problems of order independent transparency; semi-transparent light occluders; non-uniform, light scattering media and visualization of voxel based data. It can also enable the interactive graphics researchers to adapt some algorithms previously only found in off-line solutions. Its general form also allows it to work seamlessly with hardware-accelerated multi-sampling techniques. The contributions of this work include:

- general description of the data structures used in sparse to dense buffer condensation process;
- mapping of the general structures into actual hardware resources;
- rendering process for multi-layered sparse framebuffer;
- condensing the sparse buffer into dense fragment list buffer;
- mapping the list buffer into displayable framebuffer.

1 Introduction

Traditional, hardware-accelerated rendering architectures such as OpenGL [1] or Direct3D [2] work by converting three-dimensional objects represented as sets of polygons to a flat, two-dimensional framebuffers. This process is commonly known as hardware-accelerated rasterization. Throughout years each new generation of hardware was similar to the last one in terms of its basic functionality. Although we gained restricted programmable access to the rasterizing pipeline, its underlying structure remained the same. Accepting restrictions of data elements independence and processing homogeneity, allows many data elements to be processed at the same time in similar fashion (data parallelism) and to process independent objects at different stages of pipeline (task parallelism). If we consider all pipeline stages in different hardware generations, one of them remains notably unchanged. It is the final picture element composition engine. Only recently, we obtained less strict access to data gathering parts of these so-called raster operations [3] [4]. But sometimes we need to know more about the output data than simply the color and window space depth. The best example here can be visualization of semi-transparent surfaces. It is also present in basic polygon-based rasterization [5] as well as in volume element (voxel) methods [6].

2 Related Work

Bavoil et al. [7] generalized the concept of so called k-buffer, first proposed by Callahan et al. [8] for the purpose of direct rendering of volume data sets. They present their algorithm as requiring only single geometry pass. K-buffer can be used to perform final fragment composition for three-dimensional scenes with a limited ordering of its geometrical primitives. They describe k-buffer as a data structure holding some limited, constant number of k data elements for each pixel. This data structure is operated upon by using Read-Modify-Write (RMW) operations, distinguishing given usage by using custom Modify operation. They note that implementing k-buffer on current hardware could lead to possible race conditions, due to undefined geometry rasterization order. In their work they divide previous algorithms into two groups according to the number of geometry passes that they require.

The first group lists single-pass algorithms:

- Catmull’s [10] Z-buffer – for storing single data element for each pixel;
- Carpenter’s [8] A-buffer – storing unbound number of data elements per-pixel;
- Wittenbrink’s [11] R-buffer – using FIFO queue for storage;
- Z³-buffer – using element merging in case of an overflow.

The second group lists multi-pass algorithms, that is:

- F-buffer – similar to R-buffer, but unsorted and requiring the geometric primitives to be rendered in an intended order;
- Depth peeling - dividing primitives into sorted subsets;
- Vis-Sort - using occlusion queries for sorting convex objects;

The usage patterns of a k-buffer are divided into two groups:

- data driven – storing fixed number of data elements for each pixel;
- process driven – defined mainly by the Modify operation.

The first of them practically uses k-buffer’s data storage as a cache for multi-argument operation. They also propose two modifications to hardware-accelerators that would enable the direct implementation of a k-buffer. Both of them can potentially lead to serious hampering of performance due to invading into processing order or cache policies. Bavoil also describes a sample implementation using hardware-accelerated OpenGL [12]. But his solution can lead to artifacts connected to undefined processing order in current hardware. To avoid memory RMW hazards, they propose to group geometric objects or their primitives into non-screen-space-overlapping sets. They summarize their findings by listing timing values that result from using k-buffer for depth peeling. The listed values can be described as being relatively interactive and strongly shaped by used RMW artifacts reduction method.

Other most popular way of storing multiple data elements per-pixel is the so-called a-buffer first proposed by Carpenter [9]. It works by storing values in a

list for each frame-buffer pixel. In practice its use is limited by hardware memory constraints or method used for its construction. For instance using multi-sampled-anti-aliased (MSAA) framebuffer to store the consecutive list values [13], we are effectively limited by the number of color samples. In [13] Myers and Bavoil proposed using stencil test to route separate buffer values into separate pixel samples in a MSAA framebuffer. They used the configurable multi-sampling mask [14] to change particular color sample. They also proposed using hardware occlusion queries [15] to find cases of buffer overflow. Using MSAA to route values also prevents them from using it in its intended way.

We consider our method to be most similar to the original concept of an a-buffer [9], that is we do not pose any harsh restriction through rendering method used like in [13]. Our method is also similar to Myers a-buffer [9] – it uses stencil- and alpha-tests to properly store the incoming values, but by not using MSAA to route values, we can use it in traditional multi-sampled rendering. Additionally by using MRT [19] in place of multi-layer geometry shader [16] [17] sparse framebuffer rendering, we allow our solution to work even on Direct3D 9 level hardware.

3 The l-buffer

In this section we present our concept of so called l-buffer (abbreviated from layered and list buffers). We describe it in terms of a general three-dimensional visualization API such as OpenGL [1] or Direct3D [2]. We intended to keep our solution as abstract as possible. This paper presents a general concept of our solution and leaves the implementation details to future publications.

Similarly as an a-buffer, the l-buffer can potentially store unlimited number of data elements per-pixel. Structure-wise it is most similar to a deque container, storing references to multiple fragment lists of varying length in a pixel indexed array. We present it as a storage mechanism, not restricting its usage pattern by forcing the method of content modification. This paper describes only operations that are interesting from the data container point of view. In the next couple of subsections we outline the process of building l-buffer from image data. To keep the description high-level enough we introduce some general terms (for instance: count buffer), that can be easily mapped into hardware components (such as stencil buffer). When there is a straight functionality mapping of given term to hardware resource, we rather use the hardware derived name. Our algorithm can be conceptually divided into a series of steps:

1. Render into two-dimensional count-buffer;
2. Reduce the two-dimensional count-buffer;
3. Allocate sparse two-dimensional layered buffer;
 - (a) Allocate sparse attribute buffer;
 - (b) Allocate sparse count buffer;
 - (c) Initialize the layers of the count-buffer;
4. Render into the two-dimensional layered buffer;
5. Calculate the one-dimensional map buffer;

- (a) Allocate address buffer;
- (b) Allocate offset buffer;
- (c) Allocate count buffer;
- (d) Clear the offset buffer;
- (e) Initialize the count buffer;
- 6. Allocate and fill the dense one-dimensional list buffer;
 - (a) Allocate dense attribute buffer;
 - (b) Gather sparse buffer into dense list buffer using map buffer;
- 7. Process the contents of list buffer;
- 8. Scatter the dense buffer into displayable framebuffer using map buffer.

3.1 Step One: Render into Two-Dimensional Count-Buffer

We start by rendering geometry into so called two-dimensional count-buffer (see fig. 1). It can be viewed as functionally equivalent to a stencil buffer:

- it contents can be cleared to 0;
- it contents can be incremented by polygon’s fragments;
- it can be read or written, but not read-written at the same time.

To ease the access to its contents in further processing and assure that its values can be represented in broad enough range, we propose the alpha part of a color buffer to be used for its implementation. The rendering into this buffer should be enclosed into occlusion query, that can count the number of all rendered fragments. After this step, each of the pixels of a count buffer holds the number of covering fragments. Having the final contents of the count-buffer we can perform one additional occlusion query, drawing a full screen quad with count test set to pass on not equal to reference value that should be set to zero. The count buffer should be left unchanged. Alternatively we could find the number of non-empty pixels during count-buffer reduction (see next subsection).

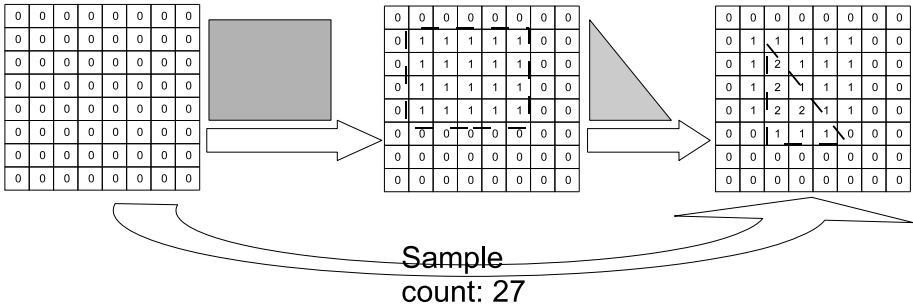


Fig. 1. Rendering of two overlapping primitives into a count buffer

3.2 Step Two: Reduce the Count-Buffer

In this step we get the maximum number of fragments per-pixel. It can be performed for instance as two-dimensional folding of the buffer while choosing the maximum value from each of the four pixels of the folded count-buffer (see fig. 2). This value will be used as the layer count in the layered framebuffer.

3.3 Step Three: Allocate Sparse Two-Dimensional Layered Buffer and Initialize the Layers of Its Count-Buffer

Next we allocate sparse two-dimensional layered buffer, capable of holding maximum number of fragments per-pixel (see fig. 2). Each layer of the buffer should contain both attribute- and count-buffer. The count-buffer will be used to route rendered fragments into separate layers. Next the count portion of the two-dimensional sparse layered buffer is initialized with numbers – starting at two up to the number of layers plus one. The shift in smallest count index should allow us to easily detect overflow cases.

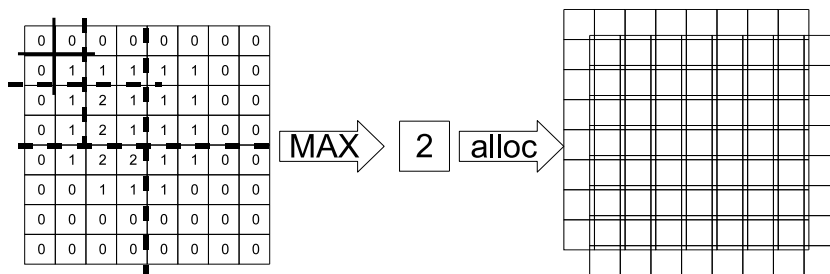


Fig. 2. Reducing the count buffer and allocating the sparse two-dimensional buffer

3.4 Step Four: Render into the Two-Dimensional Layered Buffer

Next we render all geometry into all layers of the sparse buffer (see fig. 3). The count-buffer operation should be set to decrement with saturation to zero. The count-buffer test should be set to pass for pixels with count-buffer values equal to two. We therefore use the count-buffer as a routing mechanism, writing-out incoming fragments into next available layer. After this step, the fragments are arranged into dense per-pixel lists. But the two-dimensional buffer is sparse, because it has some unused pixels due to the need of independence between parallel pipelines. It may need to be performed couple of times for different object groups. In particular when the maximal fragment count, found earlier, exceeds (due to memory constraints) the maximal dimensions restriction for two-dimensional texture array object [17]. In this case we should repeat this and subsequent steps until all geometric primitives have been accounted for. To detect the case of an overflow, we can draw a full screen quad into the last layer of the count-buffer with count test set to pass on equal to zero. We can use boolean occlusion queries [18] to detect the overflow cases.

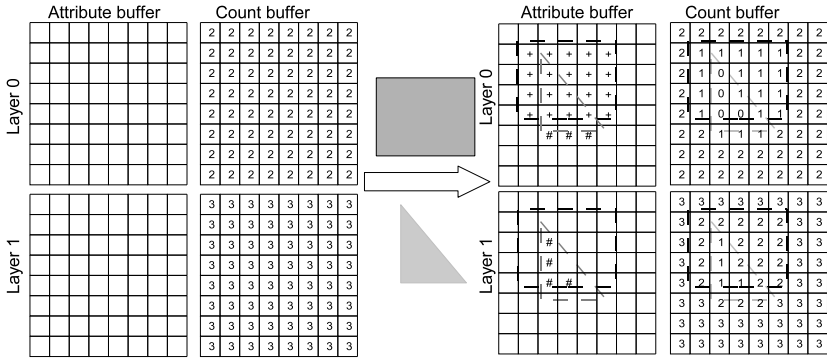


Fig. 3. The count and attribute parts of the sparse buffer

3.5 Step Five: Calculate the One-Dimensional Map Buffer

In this step we create a buffer holding the address and offset of the pixel in a sparse layered-buffer (see fig. 4). This portion of the map buffer should be able to store two-component integer vectors. The second portion should be able to store integer scalars and to increment its entries by a given integer value. Just like in the two-dimensional sparse buffer’s case we will also need a count buffer to route memory writes into subsequent map buffer entries. Summing up, the map buffer should consist of three one-dimensional sub-buffers, that is:

- a two-component integer address buffer;
- a one-component integer offset buffer;
- a one-component integer count buffer.

The size in terms of the number of entries for each sub-buffer of the map buffer should be equal to the number of non-empty sparse buffer pixels that we have found in the first or second step. We should start by initializing the contents of the count sub-buffer with element indexes going from one up to the number of non-empty pixels in the sparse buffer. The offset sub-buffer should be initialized to zero for each entry. The filling process for this buffer should go as follows:

1. For each non-empty entry in the sparse two-dimensional layered buffer.
 - (a) Write its address into all unmasked entries in the address buffer.
 - (b) Add the number of fragments to all unmasked entries in the offset buffer.
 - (c) Decrement by one with saturation to zero all entries in the count buffer.
2. Shift the contents of the offset buffer to the right by one, filling with zero.

Given entry in buffer is unmasked, if the corresponding entry in count buffer is greater then zero. After this step the map buffer should contain for the first of its entries the address of the first processed non-empty pixel of the sparse buffer and zero in the offset and count buffer. The last entry in the map buffer will hold the address of the last processed non-empty pixel and the offset equal to

the number of all fragments in the sparse buffer minus the number of fragments for the last non-empty pixel. To process only the non-empty pixels in the sparse buffer we can either use vertex shader, outputting line primitives covering whole map buffer and invalid (clipped) vertexes for the empty ones. We could also use geometry shader to construct such line primitives on the fly. If due to memory constraints we were unable to create sufficiently large two-dimensional sparse buffer and thus were forced to resort to the multi-pass approach, we should properly increment the contents of the offset buffer after each pass.

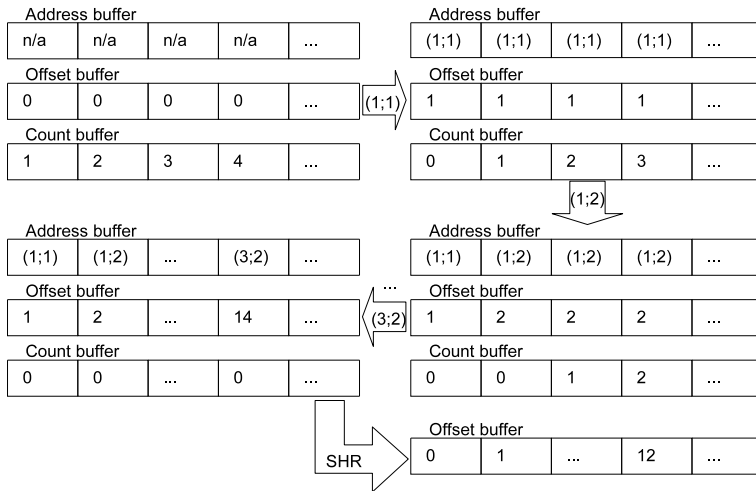


Fig. 4. The map buffer after processing of three non-empty pixels. Note that the pixels may be processed in unspecified order and thus the map buffer may be unordered.

3.6 Step Six: Allocate and Fill the Dense One-Dimensional List Buffer

We should then allocate one-dimensional dense buffer able to hold all non-empty attribute entries from the two-dimensional sparse buffer. Therefore its size should be equal to the fragment count obtained in the first step. Having the map buffer ready, we can condense the contents of the sparse attribute buffer into one-dimensional dense fragment list (see fig. 5):

1. For each element of the map buffer.
 - (a) Find entries in the sparse buffer corresponding to given address.
 - (b) Draw a line with first vertex position proportional to the value found in offset buffer and the last vertex shifted by the value proportional to the number of entries (in the two-dimensional count buffer, see step one).

- (c) Sample the two-dimensional sparse buffer with pair of first texture coordinate components equal to the address found earlier and the third component proportional to the range from zero up to the number of fragments for given pixel minus one.

The second sub-step of the above algorithm can be realized through vertex or geometry shader analogously as in step five. The third sub-step should optimally be done with a fragment shader. Just as before, if we use a multi-pass approach we should properly adjust the input and output coordinates.

After this step we have access to the dense one-dimensional list buffer representation of the sparse buffer. We can now perform a processing algorithm of choice for its data and when done remap the results back into two-dimensional displayable framebuffer using the one-dimensional map buffer and a vertex or geometry shader’s scatter capabilities.

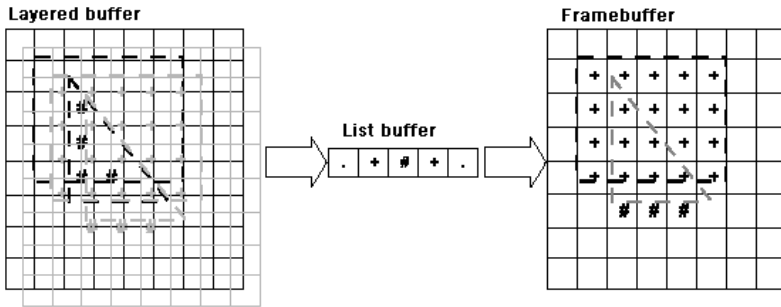


Fig. 5. Data condensation and final composition

4 Conclusion and Future Work

In this paper we have presented the l-buffer concept as a method of condensing two-dimensional sparse data sets into one-dimensional list. Our algorithm was designed with parallel implementation on modern graphics hardware in mind. Therefore it can bring a real breakthrough in interactive graphics by allowing the researchers to use algorithms up until now found only in offline renderers. We intentionally decoupled the data condensation from data processing so that our solution would be general enough to satisfy the needs of all developers.

We look forward to seeing new effects, not applicable to interactive graphics before, due to its reliance on the aging z-buffer algorithm. We also work on a reference implementation with multi-pass condensation, to boost the work of other researchers wanting to apply new algorithms to interactive graphics.

References

1. OpenGL 4.0 Core Profile Specification (updated March 11, 2010),
<http://www.opengl.org/registry/doc/glspec40.core.20100311.pdf>
2. Microsoft DirectX 11,
<http://www.microsoft.com/games/en-US/aboutGFW/pages/directx.aspx>
3. GL_ARB_multisample OpenGL extension specification,
<http://www.opengl.org/registry/specs/ARB/multisample.txt>
4. Microsoft DirectX 10.1,
<http://www.microsoft.com/games/en-US/aboutGFW/pages/directx.aspx>
5. Everitt, C.: Interactive order-independent transparency. Tech. rep., NVIDIA Corporation (2001)
6. Eisemann, E., Coret, D.: Fast scene voxelization and applications. In: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 71–78 (2006)
7. Bavoil, L., Callahan, S.P., Lefohn, A., Comba, J.L.D., Silva, C.T.: Multi-Fragment Effects on the GPU using the k-Buffer. In: Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp. 97–104 (2007)
8. Callahan, S.P., Ikits, M., Comba, J.L.D., Silva, C.T.: Hardwareassisted visibility sorting for unstructured volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 11(3), 285–295 (2005)
9. Carpenter, L.: The A-buffer, an antialiased hidden surface method. In: *Computer Graphics (Proceedings of SIGGRAPH 1984)*, vol. 18, pp. 103–108 (1984)
10. Catmull, E.: A Subdivision Algorithm for Computer Display of Curved Surfaces. PhD thesis, Dept. of Computer Science, University of Utah (1974)
11. Wittenbrink, C.: R-buffer: A pointerless A-buffer hardware architecture. In: *Graphics Hardware 2001*, pp. 73–80 (2001)
12. OpenGL 2.1 Specification (December 1, 2006),
<http://www.opengl.org/registry/doc/glspec21.20061201.pdf>
13. Myers, K., Bavoil, L.: Stencil Routed A-Buffer. In: *ACM SIGGRAPH 2007: Sketches* (2007)
14. GL_NV_explicit_multisample OpenGL extension specification,
http://www.opengl.org/registry/specs/NV/explicit_multisample.txt
15. GL_ARB_occlusion_query OpenGL extension specification,
http://www.opengl.org/registry/specs/ARB/occlusion_query.txt
16. GL_ARB_geometry_shader4 OpenGL extension specification,
http://www.opengl.org/registry/specs/ARB/geometry_shader4.txt
17. GL_EXT_texture_array OpenGL extension specification,
http://www.opengl.org/registry/specs/EXT/texture_array.txt
18. GL_ARB_occlusion_query2 OpenGL extension specification,
http://www.opengl.org/registry/specs/ARB/occlusion_query2.txt
19. GL_ARB_draw_buffers OpenGL extension specification,
http://www.opengl.org/registry/specs/ARB/draw_buffers.txt

Accurate Overlap Area Detection Using a Histogram and Multiple Closest Points

Yonghuai Liu¹, Ralph R. Martin², Longzhuang Li³, and Baogang Wei⁴

¹ Department of Computer Science, Aberystwyth University
Ceredigion SY23 3DB, UK

² School of Computer Science & Informatics, Cardiff University
Cardiff, CF24 3AA, UK

³ Department of Computing Science, Texas A and M University
Corpus Christi, TX 78412-5824, USA

⁴ College of Computer Science, Zhejiang University
Hangzhou 710072, P.R. China

Abstract. In this paper, we propose a novel ICP variant that uses a histogram in conjunction with multiple closest points to detect the overlap area between range images being registered. Tentative correspondences sharing similar distances are normally all within, or all outside, the overlap area. Thus, the overlap area can be detected in a bin by bin batch manner using a histogram. Using multiple closest points is likely to enlarge the distance difference for tentative correspondences in the histogram, and pull together the images being registered, facilitating the overlap area detection. Our experimental results based on real range images show that the performance of our proposed algorithm enhances the state of the art.

1 Introduction

Range image registration finds numerous applications in areas such as 3D object modelling and recognition, computer graphics, virtual reality, reverse engineering, and industrial inspection. It has thus attracted considerable attention in the 3D vision community. Research into range image registration techniques has been driven by the development of 3D laser scanning technologies which provide easy 3D data acquisition, and the computational power of modern computers (see Figure [1](#)).

Registration has two goals (i) to establish point correspondences between overlapping range images, and (ii) to estimate the rigid transformation that brings one range image into the best possible alignment with the other. The fact that these two problems must be solved simultaneously complicates the range image registration process.

1.1 Previous Work

Due to the challenging nature of automatic range image registration, a large number of algorithms have been developed. Many are based on the iterative

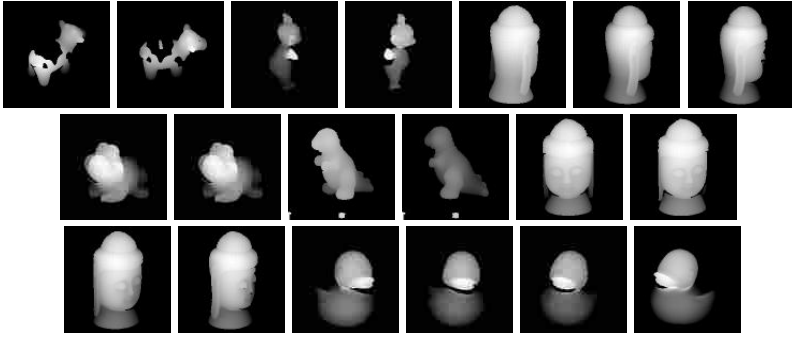


Fig. 1. Real range images used. Top, left to right: cow60 and 57, tubby100 and 60, buddha140, 120, and 100; Middle: frog20 and 0, dinosaur36 and 0, buddha0 and 20; Bottom: buddha40 and 60, duck120, 100, 80 and 60.

closest point (ICP) approach [3], and improved variants [1,9,11,15,16], as well as other approaches such as feature extraction and matching [6,14], salient point detection and matching [4], density functions [21], genetic search [12,20], scatter search [17], visibility classification and modeling [19], M-estimators [2], exhaustive search with limited points [13], graduated assignment [7], and expectation maximisation [5,8]; many others also exist.

These algorithms belong to three main categories, or a combination of them, depending on how the correspondences are established: (i) feature extraction and matching [6,14]; (ii) coarse matching of points in one image to those in another [5,7], and (iii) transformation search and evaluation [17,20]. Each category has advantages and disadvantages:

- The first class of algorithms can establish correspondences between any two overlapping range images subject with either small or large transformations, while the second and third require that the transformation is approximately known;
- The first class of algorithms has to extract and match geometric and/or optical features from a structured, a non-structured, or an analytic representation of the surface of interest, while the second and third do not have to perform feature extraction and matching;
- Extraction of features is typically sensitive to noise caused by sampling and properties of the scanning process, and furthermore, matching of features is not straightforward because: (a) features should ideally be viewpoint invariant, (b) the similarity metric must allow discrimination of different features; and (c) a feature in one image may match multiple candidate features in another, leading to a combinatorial correspondence problem. The second class of algorithms heavily depends on both optimization and explicit outlier treatment. In contrast, while the third class typically has the advantage of finding the globally optimal solution, they are usually time consuming and it is difficult to determine appropriate termination conditions.

In summary, automatic registration of overlapping range images is still not a fully solved problem.

1.2 Our Approach

The ICP algorithm is a de facto standard technique for registration of overlapping free form shapes for three main reasons: (i) it is not just an algorithm, but also a methodology, which is widely used for object recognition and data clustering; (ii) it is theoretically guaranteed to establish high quality tentative correspondences [10]; and (iii) it is usually employed to refine the registration results obtained [12]. However, it introduces false correspondences in almost every iteration caused by inaccurate transformation parameters, occlusion, and appearance and disappearance of point, and so it has attracted intensive attention for improvement from the 3D vision community.

The FICP algorithm [15] is mathematically elegant, since it simultaneously estimates both the overlapping region and the registration parameters by optimizing the root mean squared distance (RMSD) of the tentative correspondences established using the traditional closest point criterion (CPC). Inspired by the FICP algorithm, in this paper, we propose a novel ICP variant that uses a histogram in conjunction with multiple closest points to determine the overlap area between the range images being registered. If certain correspondences have similar distances, typically, they will *all* simultaneously fall either inside or outside the overlap area. Thus, detection of the overlap area may be performed by considering batches. To implement this idea, we employ a histogram of the squared distances (SDs) between tentative correspondences: all tentative correspondences with similar distances fall into the same bin. The objective function that minimizes the Euclidean distances between the tentative correspondences is evaluated for each bin, allowing detection of the overlap area to be done in a batch manner: if *any* correspondence from a bin lies in the overlap area, then all the others in the same bin also lie in the overlap area and vice versa.

The SDs of the tentative correspondences determine their distribution in the histogram. To enlarge the difference between distances belonging to overlap and non-overlap areas, we replace the estimate the SD of a tentative correspondence by the sum of the SDs between a point in one image and a number of closest points in the other. In the overlap area, such multiple closest points (MCPs) will not make much difference in the sense of distinguishing between different correspondences. However, in the non-overlap area, they may cause tentative correspondences to differ significantly in their SDs and thus fall into different bins of the histogram, facilitating the differentiation of real correspondences from false ones. Real correspondences will typically have lower distances.

To evaluate our proposed algorithm, we have also implemented the FICP algorithm [15] and one of the latest ICP variants, Geometric Primitive ICP (GP-ICP) [1]. A comparative study was carried out using real range images from a Minolta Vivid 700 range camera.

The rest of this paper is structured as follows: Section 2 details our novel algorithm, Section 3 presents experimental results, and Section 4 draws some conclusions.

2 Algorithm

Assume that two range images to be registered are represented as two sets of unorganised points $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_{n_1}\}$ and $\mathbf{P}' = \{\mathbf{p}'_1, \dots, \mathbf{p}'_{n_2}\}$, representing the same free form shape viewed under a small transformation, such that \mathbf{P} and \mathbf{P}' have a relatively large overlap in 3D space and the pure translation motion derived from the difference of the centroids of \mathbf{P} and \mathbf{P}' provides a good initial estimate for the transformation, which comprises a rotation matrix \mathbf{R} and translation vector \mathbf{t} . Using the traditional ICP criterion [3], a set of tentative correspondences $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ is obtained between \mathbf{P} and \mathbf{P}' :

$$\mathbf{p}'_{c(i)} = \operatorname{argmin}_{\mathbf{p}' \in \mathbf{P}'} \|\mathbf{p}' - \mathbf{R}\mathbf{p}_i - \mathbf{t}\| \quad (1)$$

where $c(i) \in [1, n_2]$ is the label of a point in \mathbf{P}' : this is a mapping associating $\mathbf{p}'_{c(i)}$ in \mathbf{P}' with \mathbf{p}_i in \mathbf{P} . The search for a tentative correspondent is determined by the size of \mathbf{P}' . In order to speed up the search for the closest points $\mathbf{p}'_{c(i)}$, an optimised k -D tree data structure [9], and squared Euclidean distance, rather than Euclidean distance itself, are employed in our implementation.

2.1 Using a Histogram to Determine Overlap

Having obtained a set of tentative correspondences $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ between \mathbf{P} and \mathbf{P}' , the following objective function is used [15] to estimate the rotation matrix \mathbf{R} and translation vector \mathbf{t} , and the fractional overlap size f (meaning that fn_1 points belong to the overlap region):

$$\operatorname{FRMSD}(\mathbf{P}, \mathbf{P}', \mathbf{R}, \mathbf{t}, f) = \frac{1}{f^\lambda} \sqrt{\frac{1}{|\mathbf{P}_f|} \sum_{\mathbf{p}_i \in \mathbf{P}_f} \|\mathbf{p}'_{c(i)} - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2} \quad (2)$$

where \mathbf{P}_f denotes the fn_1 points $\mathbf{p}_i \in \mathbf{P}$ with smallest distances $\|\mathbf{p}'_{c(i)} - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|$. We set $\lambda = 3$ during the iterative process and $\lambda = 0.95$ in a final iteration.

Suppose that certain correspondences have similar distances. Typically, they will either *all* be inside or *all* be outside the overlap area (Figure 2). Thus, deciding whether particular correspondences belong to the overlap area can be more effectively determined in batches, which we implement using a histogram.

All tentative correspondences falling in the same bin j have similar SDs d_i and are presumed to *all* be inside or *all* outside the overlap area. The SDs d_i can be approximated as $(j+1)h_n^*$ and $j = 0, \dots, s-1$ where s is the number of bins and h_n^* is the bin width. The larger the parameter s , the better the approximation. Using this approximation, the objective function in Equation 2 can be rewritten as:

$$\text{FRMSD}(\mathbf{P}, \mathbf{P}', \mathbf{R}, \mathbf{t}, f) = \frac{1}{f^\lambda} \sqrt{\frac{1}{|\mathbf{P}_f|} \sum_{\mathbf{p} \in \mathbf{P}_f} (j+1)h_n^*}$$

which counts the number h_j of tentative correspondences whose SDs can be approximated by $(j+1)h_n^*$. This is exactly the idea of a histogram. As h_n^* is a constant, the above objective function can be further rewritten as:

$$\text{FRMSD}(\mathbf{P}, \mathbf{P}', \mathbf{R}, \mathbf{t}, f) = \frac{1}{(\sum_{j=0}^f h_j/n_1)^\lambda} \sqrt{\frac{\sum_{j=0}^f h_j(j+1)}{\sum_{j=0}^f h_j}} \quad (3)$$

which calculates the square root of a weighted average of the bin indexes with weights defined as the frequencies h_j of the corresponding bins j .

In practice, we implement the histogram based detection of the overlap area as follows:

- Use Scott’s method [18] to determine the bin width h_n^* of the histogram:

$$h_n^* = \alpha d_\sigma / n_1^{1/3} \quad (4)$$

where α is a positive number that reflects the characteristics of the actual data and d_σ is the standard deviation of $d_i = \|\mathbf{p}'_{c(i)} - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2$.

- Determine the number s of bins in the histogram: $s = (d_{\max} - d_{\min})/h_n^*$ where $d_{\max} = \max_k d_k$ and $d_{\min} = \min_k d_k$.
- Construct the histogram $\mathbf{H} = \{h_0, h_1, \dots, h_{s-1}\}$ of SDs d_i of all tentative correspondences where h_j is the frequency of $(j+1)h_n^*$: $h_j \leftarrow h_j + 1, j = \lfloor (d_i - d_{\min})/h_n^* \rfloor$ and $i = 1, \dots, n_1$.
- Compute values J_f of the objective function in Equation 3 by changing the number f of the bins in the overlap area: $J_f = \text{FRMSD}(\mathbf{P}, \mathbf{P}', \mathbf{R}, \mathbf{t}, f)$, $f = 0, \dots, s-1$.
- Select the minimum J_f to determine the points $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ in the overlap area.

Note that the computation of J_f does not involve the actual SDs d_i of the tentative correspondences. This is because they are approximated by $(j+1)h_n^*$ where h_n^* is a constant. This property provides a novel perspective for detection of overlap area.

2.2 Using Multiple Closest Points to Reduce Error

Earlier, the SD d_i of a tentative correspondence $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ was defined as $d_i = \|\mathbf{p}'_{c(i)} - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2$. Since d_i plays a crucial role in determining whether or not $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ lies in the overlap area and hence will be used for the transformation re-estimation, its exact form is important. Here, we define it as the sum of the SDs between $\mathbf{R}\mathbf{p}_i + \mathbf{t}$ and a number m of the closest points in \mathbf{P}' (Figure 2).

Computation of d_i can be easily implemented using a k -D tree without requiring significantly more computational time. Multiple closest points (MCPs) have a potential to play two roles: (i) using MCPs will not make much difference from using the single closest point in the overlap area between \mathbf{P} and \mathbf{P}' . However, in the non-overlap area, it is likely to enlarge the SDs of incorrect correspondences, thus making detection of the overlap area more robust. Figure 2 shows that the maximum of the SDs of the tentative correspondences defined over 16 MCPs has been increased by as much as $(1400 - 60 \times 16)/(60 \times 16) \times 100\% = 45.83\%$, manifested by the fact that the histogram is more representative of the distribution of the SDs of the tentative correspondences, and the bins of the histogram are more scattered; and (ii) they tend to pull together the images being registered and thus maximize the overlap area.

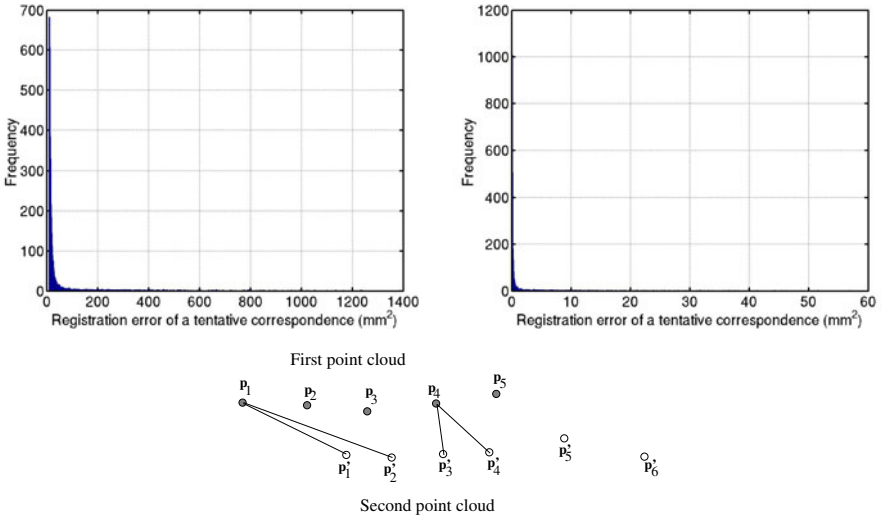


Fig. 2. Top: The histogram of SDs of tentative correspondences defined using 16 closest points (left) and a single closest point (right) respectively for the duck80 and duck60 images. Bottom: \mathbf{p}'_1 and \mathbf{p}'_2 , \mathbf{p}'_3 and \mathbf{p}'_4 are two closest points to $\mathbf{R}\mathbf{p}_1 + \mathbf{t}$ and $\mathbf{R}\mathbf{p}_4 + \mathbf{t}$ respectively. The difference between $\|\mathbf{p}'_1 - \mathbf{R}\mathbf{p}_1 - \mathbf{t}\|^2 + \|\mathbf{p}'_2 - \mathbf{R}\mathbf{p}_1 - \mathbf{t}\|^2$ and $\|\mathbf{p}'_3 - \mathbf{R}\mathbf{p}_4 - \mathbf{t}\|^2 + \|\mathbf{p}'_4 - \mathbf{R}\mathbf{p}_4 - \mathbf{t}\|^2$ can be as large as twice that between $\|\mathbf{p}'_1 - \mathbf{R}\mathbf{p}_1 - \mathbf{t}\|^2$ and $\|\mathbf{p}'_3 - \mathbf{R}\mathbf{p}_4 - \mathbf{t}\|^2$.

After the overlap area has been detected, the transformation parameters are estimated using the quaternion least square method [3]. When the difference of the average SD of the overlapping correspondences between two successive iterations falls below a threshold (0.000001) or the number of iterations has exceeded 300, iteration terminates. Since the proposed algorithm is a novel ICP variant based on the Histogram and Multiple closest points, it is denoted HM-ICP in the rest of this paper. It has a computational complexity of $O(n \log n)$.

3 Experimental Results

To validate the performance of the proposed HM-ICP algorithm, the FICP algorithm [15] and a recent ICP variant, geometric primitive ICP (GP-ICP) [1] were also implemented and compared using a Pentium IV, 2.80GHz computer. These three algorithms employ different strategies for the detection of the overlap area: (i) sorting d_i in the FICP algorithm takes time $O(n \log n)$, whereas constructing a histogram has a computational complexity of $O(n)$; (ii) the FICP algorithm detects the overlap area by considering each correspondence in turn, whereas the histogram determines the overlap area by considering bins, and there are far fewer bins than points; (iii) the FICP algorithm uses the same objective function for the detection of the overlap area and the estimation of the transformation, while our HM-ICP algorithm separates them; and (iv) while both HM-ICP and FICP algorithms simultaneously optimize the registration parameters and the overlap area, the GP-ICP algorithm employs thresholds for the rejection of false correspondences and minimizes the point to point distance at early stages of registration and the point to plane distance at later stages.

The comparative study used real range images (Figure 1) downloaded from a publicly available database hosted at the Signal Analysis and Machine Perception laboratory at Ohio State University. They were captured using a Minolta Vivid 700 range camera, and are all of size of 200×200 pixels. The purpose of these experiments is threefold: (i) to determine how to set parameters such as the histogram bin width and the number m of closest points in our HM-ICP algorithm, (ii) to test whether the histogram and MCPs can effectively detect the overlap area, and (iii) whether our HM-ICP algorithm advances the state of the art.

To compare the algorithms, we used the average and standard deviation in millimetres of registration errors of reciprocal correspondences (RCs), the estimated rotation angle in degrees of the transformation relative to the ground truth value, and the time taken in seconds for registration. A reciprocal correspondence $(\mathbf{p}_i, \mathbf{p}'_{c(i)})$ is one for which $i = c(c(i))$, implying that if \mathbf{p}_i in \mathbf{P} finds $\mathbf{p}'_{c(i)}$ in \mathbf{P}' as a correspondent, $\mathbf{p}'_{c(i)}$ in \mathbf{P}' also finds \mathbf{p}_i in \mathbf{P} as a correspondent. RCs are often used to measure the performance of registration algorithms as their determination does not involve any threshold [1,16] and hence unwanted bias. In Figures 3-5, yellow represents the transformed first image \mathbf{P} , and green represents the second image \mathbf{P}' .

3.1 Histogram Bin Width

The SD histogram bin width plays a key role in determining registration errors and speed. We investigated choice of the bin width by letting $\alpha = 0.01$, $\alpha = 0.1$, and $\alpha = 1$ in Equation 4; we used the cow60, cow57 and tubby100, tubby60 images from Figure 1 for the experiments. Results are presented in Figure 3 and Table 1, and show that using a large $\alpha = 1$ and hence a large bin width provides only a coarse approximation of the SDs of the tentative correspondences. While this leads to efficient registration, it gives poor registration results: e.g. the two

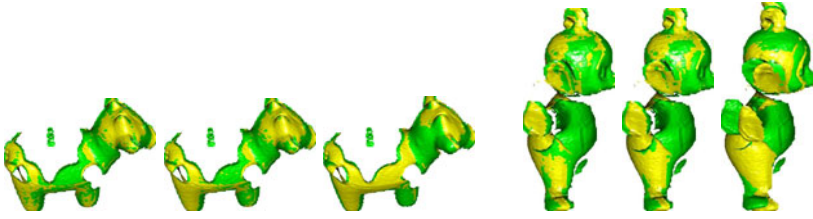


Fig. 3. Registration results for the proposed HM-ICP algorithm using two different pairs of range images and varying α . Left three: cow60-57; Right three: tubby100-60. In each case: $\alpha = 0.01$, $\alpha = 0.1$, $\alpha = 1$.



Fig. 4. Registration results for our HM-ICP algorithm applied to different overlapping range images using different values of m . Left three: buddha140-120; Right three: buddha120-100. In each case: $m = 1$, $m = 16$, $m = 25$.

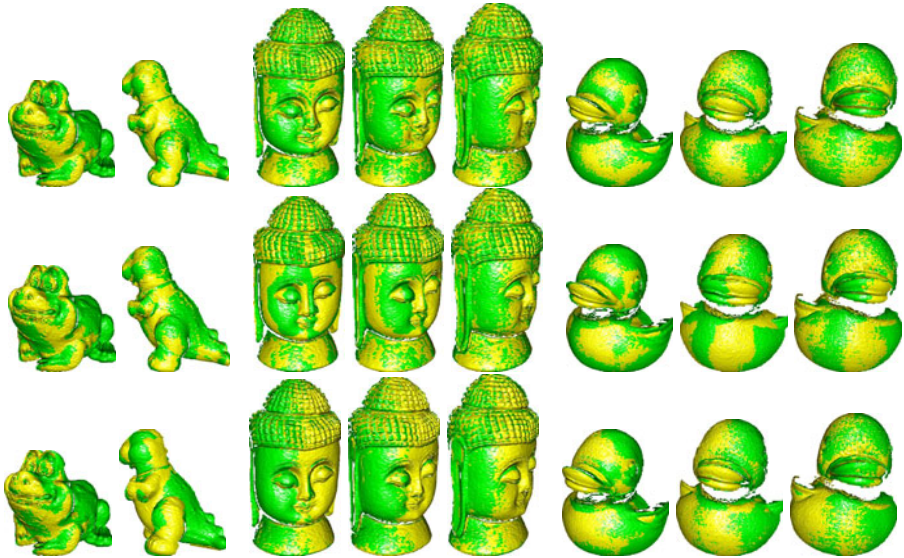


Fig. 5. Registration results for different algorithms and different range images. Top: HM-ICP; Middle: FICP; Bottom: GP-ICP. Columns from left to right: frog20-0, dinosaur36-0, buddha0-20, 20-40, and 40-60; duck80-60, 100-80, and 120-100.

Table 1. The mean e_μ and standard deviation e_δ of registration errors in millimetres for reciprocal correspondences (RCs), known and estimated rotation angles θ and $\hat{\theta}$ in degrees of the transformation, the number N of RCs, and registration time t in seconds for our HM-ICP algorithm with varying α

Images	α	e_μ (mm)	e_δ (mm)	θ ($^\circ$)	$\hat{\theta}$ ($^\circ$)	N	t (sec.)
cow60-57	0.01	0.49	0.19	30	30.39	3249	9
	0.10	0.50	0.19		30.17	3255	7
	1.00	0.71	0.29		28.80	3215	5
tubby100-60	0.01	0.29	0.19	40	39.13	2056	20
	0.10	0.29	0.19		37.96	2053	17
	1.00	0.52	0.33		27.31	1649	10

hands of the tubby in the two images are separated. Using a small bin width, setting $\alpha = 0.01$, provides a good approximation of the SDs, leading to more accurate registration results at a cost of increased computational time. $\alpha = 0.1$ provides a good compromise between accuracy and speed. Thus, in later tests, we set $\alpha = 0.1$.

3.2 Number of Closest Points

The number of the closest points has a subtle effect on determination of SDs for the tentative correspondences. We experimentally investigated the effect of varying the number m of closest points, setting $m = 1$, $m = 16$, and $m = 25$. We used the buddha140, buddha120, and buddha100 image pairs in Figure 1 for the experiments. Results are presented in Figure 4 and Table 2, which show that as expected, a large number of closest points, $m = 25$, leads to most accurate registration results but at a cost of greater computational time, while setting $m = 1$ increases computational efficiency but produces poorer results: e.g. the two ears of the buddha in the two images are misplaced. In practice $m = 16$ seems a good compromise between accuracy and speed. Thus, henceforth, we set $m = 16$.

Table 2. Average e_μ and standard deviation e_δ of registration errors in millimetres for RCs, known and estimated rotation angles θ and $\hat{\theta}$ in degrees of the transformation, the number N of RCs, and registration time t in seconds for our HM-ICP algorithm, using different numbers m of closest points

Images	m	e_μ (mm)	e_δ (mm)	θ ($^\circ$)	$\hat{\theta}$ ($^\circ$)	N	t (sec.)
buddha140-120	1	0.82	0.39	20	4.42	8153	58
	16	0.59	0.25		20.08	10070	163
	25	0.59	0.26		20.21	10097	199
buddha120-100	1	0.84	0.45	20	2.75	8061	30
	16	0.58	0.26		20.46	9760	140
	25	0.58	0.26		20.46	9755	152

Table 3. Average e_μ and standard deviation e_δ registration errors in millimetres for RCs, known and estimated rotation angles θ and $\hat{\theta}$ in degrees of the transformation, the number N of RCs, and registration time t in seconds for different algorithms

Images	Algorithm	e_μ (mm)	e_δ (mm)	θ ($^\circ$)	$\hat{\theta}$ ($^\circ$)	N	t (sec.)
frog20-0	HM-ICP	0.30	0.15	20	18.84	5806	38
	FICP	0.30	0.15		18.83	5798	16
	GP-ICP	0.31	0.15		18.58	5726	16
dinosaur36-0	HM-ICP	0.60	0.54	36	35.11	5302	29
	FICP	0.56	0.54		35.66	5309	23
	GP-ICP	1.38	1.41		23.96	3321	31
buddha0-20	HM-ICP	0.58	0.24	20	20.10	11011	137
	FICP	0.81	0.49		13.77	9353	64
	GP-ICP	0.73	0.32		18.18	10531	29
buddha20-40	HM-ICP	0.58	0.25	20	20.11	10786	130
	FICP	0.71	0.35		17.16	9859	58
	GP-ICP	0.67	0.27		18.78	10576	29
buddha40-60	HM-ICP	0.58	0.25	20	19.81	10551	115
	FICP	0.58	0.25		19.41	10535	51
	GP-ICP	0.61	0.25		19.01	10472	26
duck80-60	HM-ICP	0.30	0.17	20	19.08	7529	87
	FICP	0.31	0.18		17.77	7517	38
	GP-ICP	0.32	0.17		19.39	7454	23
duck100-80	HM-ICP	0.27	0.15	20	18.06	7229	103
	FICP	0.41	0.27		6.03	6700	22
	GP-ICP	0.28	0.14		18.23	7144	25
duck120-100	HM-ICP	0.27	0.12	20	19.19	7567	84
	FICP	0.30	0.15		16.47	7446	38
	GP-ICP	0.28	0.12		18.39	7551	27

3.3 Comparative Study

In this section, we compare our new HM-ICP algorithm with FICP and GP-ICP algorithms using 8 pairs of real overlapping range images shown in Figure 1: frog20-0, dinosaur36-0, buddha0-20, 20-40, and 40-60, duck80-60, 100-80, and 120-100. The buddha head is of a generally cylindrical shape. The duck head and belly are of a generally spherical shape. Such shapes are challenging to register, since simplicity leads to ambiguity in transformation parameters. Results are presented in Figure 5 and Table 3. Both the HM-ICP and FICP algorithms produced similar results for the registration of geometrically complex frog20-0 and dinosaur36-0 images. The FICP algorithm had difficulties registering the buddha and duck images, giving poor average registration errors for reciprocal correspondences and rotation angle, because (i) a point difference of the objective function in Equation 2 does not always reflect a difference in the overlap area, especially when the tentative correspondences have similar distances; and (ii) an objective function suitable for the transformation estimation is not necessarily also suitable for detection of the overlap area. Our HM-ICP algorithm produces

much better results, decreasing the average registration error by up to 17%. As the FICP algorithm is a refinement algorithm, this error decrease is valuable and represents a significant achievement. Accurate registration results are manifested as a large amount of interpenetration [20] between the superimposed images.

The reasons why multiple points in the same bin of the histogram are a powerful tool for the detection of the overlap area can be explained as follows. MCPs render the data in the bins in the histogram to become more scattered, as seen in Figure 2. This facilitates the discrimination of correct correspondences from false ones. Tentative correspondences in the same scattered bins also play a role in perturbing the existing registration parameters and thus preventing our HM-ICP algorithm from converging prematurely. This perturbation arises due to use of MCPs about the local geometry in the images and relies on accuracy of the existing registration parameters. This is desirable and in sharp contrast with the ad hoc random approach [11] that is usually not reliable. Our HM-ICP algorithm requires an additional computational time of 133% compared to the FICP algorithm. This is because the latter usually converges prematurely.

Even though the dinosaur36-0 images are geometrically complex, the GP-ICP algorithm still produced much worse results, with average errors compared to our HM-ICP algorithm as much as 130% higher. It exhibits a similar behaviour for registering the geometrically simple buddha and duck images. In this case, it increases the average error by as much as 12%. The GP-ICP algorithm produces worse results because it is difficult to set up thresholds for the rejection of false correspondences. In the later stages of registration, it also has to invert a matrix to estimate the registration parameters. When this matrix is close to singular, the estimated registration parameters are unreliable and may lead the GP-ICP algorithm to fail to find any true correspondence in the next iteration. On the other hand, it minimizes the point-to-plane distance, but not the point-to-point distance.

4 Conclusions

This paper has proposed a novel ICP variant that uses both a histogram and multiple closest points to detect the overlap area during range image registration. Our experimental results show that accurate detection of the overlap area is obtained for various real range images even with relatively simple geometry that is typically more challenging to register. This is because the role of multiple points in the same scattered bin of the histogram is more pronounced than a single point in the FICP algorithm [15] in perturbing the existing registration parameters and helping our HM-ICP algorithm to traverse the local minimum of the objective function during registration of overlapping range images.

Our HM-ICP algorithm also outperforms the latest ICP variant, Geometric Primitive ICP (GP-ICP) [1], because our approach simultaneously optimizes both the overlap area and the registration parameters, while the latter has difficulty in choosing thresholds for rejection of false correspondences and in minimizing the point-to-plane distance. Thus, our HM-ICP algorithm advances the state of the art for registration of overlapping range images. Future research will consider how to reduce the computational cost of the HM-ICP algorithm.

References

1. Bae, K.-H.: Evaluation of the convergence region of an automated registration method for 3D laser scanner point clouds. *Sensors* 9, 355–375 (2009)
2. Banno, A., Masuda, T., Oishi, T., Ikeuchi, K.: Flying laser range sensor for large-scale site-modeling and its application in Bayon digital archival project. *IJCV* 78, 207–222 (2008)
3. Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. *IEEE Trans. PAMI* 14, 239–256 (1992)
4. Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum* 27, 643–652 (2008)
5. Dewaele, G., Devernay, F., Horaud, H.: Hand motion from 3D point trajectories and a smooth surface model. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 495–507. Springer, Heidelberg (2004)
6. Dinh, H.Q., Kropac, S.: Multi-resolution spin images. In: *Proc. CVPR*, pp. 863–870 (2006)
7. Gold, S., Rangarajan, A., et al.: New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. *Pattern Recognition* 31, 1019–1031 (1998)
8. Granger, S., Pennec, X.: Multi-scale EMICP: a fast and robust approach for surface registration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2353, pp. 418–432. Springer, Heidelberg (2002)
9. Liu, Y.: Replicator dynamics in the iterative process for accurate range image matching. *IJCV* 83, 30–56 (2009)
10. Liu, Y.: Constraints for closest point finding. *Pattern Recognition Letters* 29, 841–851 (2008)
11. Liu, Y.: Improving ICP with easy implementation for free form surface matching. *Pattern Recognition* 37, 211–226 (2004)
12. Lomonosov, E., Chetverikov, D., Ekart, A.: Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm. *Pattern Recognition Letters* 27, 1201–1208 (2006)
13. Marquez, M.R.G., Wu, S.-T.: Using simplified meshes for crude registration of two partially overlapping range images. In: *WSCG*, pp. 183–190 (2007)
14. Masuda, T.: Log-polar height maps for multiple range image registration. *Computer Vision and Image Understanding* 113, 1158–1169 (2009)
15. Phillips, J.M., Liu, R., Tomasi, C.: Outlier robust ICP for minimizing fractional RMSD. In: *Proc. 3DIM*, pp. 427–434 (2007)
16. Pulli, K.: Multiview registration for large data sets. In: *Proc. 3DIM*, pp. 160–168 (1999)
17. Santamaria, J., Cordon, O., Damas, S., Aleman, I., Botella, M.: A scatter search-based technique for pair-wise 3D range image registration in forensic anthropology. *Soft Computing* 11, 819–828 (2007)
18. Scott, D.W.: On optimal and data-based histograms. *Biometrika* 66, 605–610 (1979)
19. Sharp, G.C., Lee, S.W., Wehe, D.K.: Maximum-likelihood registration of range images with missing data. *IEEE Trans. PAMI* 30, 120–130 (2008)
20. Silva, L., Bellon, O.R.P., Boyer, K.L.: Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. PAMI* 27, 762–776 (2005)
21. Wang, F., et al.: Groupwise point pattern registration using a novel CDF-based Jensen-Shannon divergence. In: *Proc. CVPR*, pp. 1283–1288 (2006)

Automatic Extraction of the Lower Boundary of the Mandibular Bone in Dental Panoramic Radiographs

Przemysław Maćkowiak

Poznań University of Technology

Chair of the Multimedia Telecommunications and Microelectronics, Poland
przemyslaw.mackowiak@gmail.com

Abstract. The paper presents a new technique for extraction of the lower mandibular bone boundary on dental panoramic radiographs. The proposed method can be the first step in examination of the morphology of the mandibular bone or in the computation of the cortical width. The described algorithm is robust against presence of the undesired structures that in an image. The technique was tested on 34 images. The results obtained are equivalent with the differences between boundaries hand drawn by radiologists.

1 Introduction

The main goal of the article is to propose an original technique for extraction of the lower boundary of the mandibular bone on dental panoramic radiographs. The upper and lower boundaries are important features that can be exploited to support osteoporosis investigation.

Osteoporosis is characterized by low bone mineral density and is associated with deterioration of the structure of bone tissues. Osteoporosis is detected mainly with the double-energy X-ray absorptiometry (DEXA). Unfortunately this method is very expensive. Therefore, cheaper and more accessible methods become the area of medical research. Some reports indicate that the use of dental panoramic radiograph (DPR) is helpful in osteoporosis investigation [1, 2]. Horner K. et al. [1] demonstrate that individuals with the cortical width (Fig. 1(a)) below 3mm should be advised on to do further osteoporosis investigation. One of the last work concludes that measurement of the above mentioned parameter is the alternative method of risk assessment for people who have undergone radiography for the usual dental reasons [2].

Both the cortical width measurement and the morphology examination of the mandibular bone can be supported by the computer aided system. There are only few works that present methods for analysis of the mandibular bone on DPR. One of them [3] uses the active shape models in order to find the upper and lower boundaries of the mandibular bone. Once the boundaries are located, the thickness of the bone is measured in different sections. Unfortunately, this method requires a training data set for which a specialist manually indicates

the landmark points. Other technique [4] is based on the morphological operations and the nonlinear filtering. This semi-automatic technique was designed to compute the cortical width and requires selecting a region of interest and some landmark points. Traditional edge detection algorithms (Canny detector [5], LoG (Laplacian of Gaussian) [6] and looking for zero crossing, linear filtering with Sobel/Prewitt kernel associated with local or global thresholding) are not appropriate for the extraction of the lower boundary of the mandibular bone. In contrast to the above mentioned algorithms, the presented method does not require any training data like in [3] and is automatic once the region of interest is located.

2 Lower Boundary Extraction

The method of extracting the lower boundary should generate an edge that is single, continuous and has specific shape. The last condition means that the shape of the lower boundary can be approximated by the third order polynomial. Traditional edge detection techniques do not meet the above conditions. Therefore, a new five-step method is developed and described in this section:

1. Region of interest selection,
2. Gradient computation and classification,
3. Initial boundary extraction and approximation,
4. Curve correction,
5. Edge pixels detection.

2.1 Region of Interest Selection

As a first step a specialist (e.g. a radiologist) marks the regions of interest (ROI) with respect to the points A, B, C (Fig. 1). The first one is the mental foramen (point A). Points B and C correspond to places where the upper and lower boundary point lie respectively. ROI cut in the presented way represents the mandibular bone sufficiently.

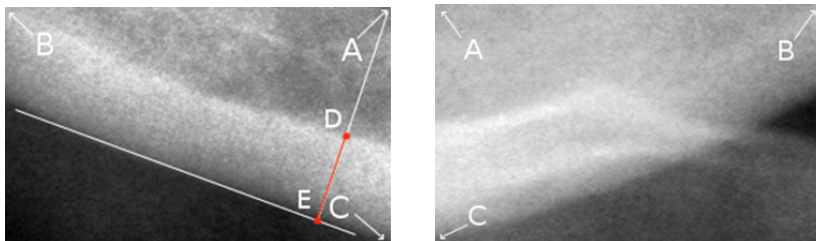


Fig. 1. The correctly (a) and incorrectly (b) acquired image. A - the mental foramen, B - point of the upper boundary, C - point of the lower boundary, $|DE|$ is the cortical width.

2.2 Gradient Computation and Classification

The main idea behind the gradient computation is to emphasize the brightness differences associated with the region of the lower boundary. The gradient image is computed using multi-scale derivatives of Gaussian kernels to adapt to the spatial scale of edges in an image [7]. The multi-scale gradient value for a point in the region of interest is associated with the greatest gradient value at this point and is computed according to the below equation [8]:

$$G'(m, n) = \max_{\sigma}(G_{\sigma}(m, n)) = \max_{\sigma}(\sqrt{(G_{x,\sigma}(m, n))^2 + (G_{y,\sigma}(m, n))^2}) \quad (1)$$

where $G_{x,\sigma}(m, n)$ and $G_{y,\sigma}(m, n)$ are values of the directional derivative (horizontal and vertical direction respectively) at point (m,n) for scale σ . These components are given by:

$$G_{x,\sigma}(m, n) = I(m, n) * \frac{\partial N_{\sigma}(m, n)}{\partial n} \quad (2)$$

$$G_{y,\sigma}(m, n) = I(m, n) * \frac{\partial N_{\sigma}(m, n)}{\partial m} \quad (3)$$

where I is the original image and $N_{\sigma}(m, n)$ is the two-dimensional zero-mean Gaussian function with scale parameter σ . The distribution of the gradient angles at lower boundary shows that a value of y component of gradient is greater than zero and a value of x component of gradient is greater or less than zero. Therefore, the gradient image which is used in the further processing is computed according to the equations (4) and (5):

$$G_R(m, n) = \begin{cases} G'(m, n), & G_{y,\sigma_{max}}(m, n) \geq 0 \text{ and } G_{x,\sigma_{max}}(m, n) < 0; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

$$G_L(m, n) = \begin{cases} G'(m, n), & G_{y,\sigma_{max}}(m, n) \geq 0 \text{ and } G_{x,\sigma_{max}}(m, n) > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where σ_{max} corresponds to a scale for which a value of $G'(m, n)$ is the greatest, $G_L(m, n)$ and $G_R(m, n)$ denote the gradient value for the left and right side of an image respectively. In order to decrease the processing time, the range of σ is restricted to the set $\{0.5, 1.0, 1.5, 2.0\}$.

There are three main different structures in the region of interest: air, the trabecular bone and the mandibular bone. If an image was acquired correctly, the air area is homogenous. That is why, the gradient values are low. The trabecular bone exhibits low contrast and is characterized by porous texture. Therefore, the gradient values are moderate. The lower part of the mandibular bone lies in the region of high contrast and is described by the high gradient values. The upper part is associated mainly with low values of gradient. Therefore, three classes of pixels based on the gradient value can be distinguished. The simple k-means algorithm is used to classify pixels to one of the three classes. The first

class comprises points with low level gradient, the second class contains points whose gradient value is moderate, and the last one is associated with points of high gradient values.

2.3 Initial Boundary Extraction and Approximation

It can be observed that the lower boundary lies between the lower part of the mandibular bone and the air. The lower part is associated with points of the high gradient values (class 3) and points of the air area are associated with the class of low gradient values (class 1). Therefore, points whose bottom neighbor belongs to class 1 and upper neighbor belongs to class 3 are selected. Together they form the initial boundary. It does not correspond with the lower boundary, because the k-means algorithm does not assure that the boundaries between classes correspond with edges. However, it can be used as a reference curve that will be moved towards the lower boundary (section 2.5). A description showing step-by-step how the initial boundary y_{IB} is extracted is presented below:

1. Let $k = 1$,
2. Find in the k -th column pixels whose upper and lower neighbor belong to class 3 and 2 respectively,
3. If no pixel was not found, find pixels whose upper and lower neighbor belong to class 2 and 1 respectively,
4. Check whether at least one pixel was found. If yes, go to the next step, otherwise $k = k+1$ and go to 2,
5. From the set of pixels found, select the one which is associated with the greatest region. Store the location of the chosen pixel,
6. Increase k by 1 if and only if the last column was not reached, otherwise stop algorithm.

A set of connected pixels creates a group which is called here "an object". Pixels p_1 and p_2 are connected if and only if p_2 is a vertical or horizontal neighbor of p_1 . The term "greatest region" refers to an object created by the greatest number of points. The initial boundary consists mainly of points of the third class (high gradient value). The contrast along the lower boundary varies and depends on noise. Shadows can appear in a specific place in the region of interest if an image is acquired wrongly (the head position is not set appropriately). Then, the points of the third class will not be found in that region and point of the second class are used (gradient value is moderate).

The curve generated in the way described above is not a curve according to mathematical definition. The method does not ensure that the curve is continuous. In general, it contains many small edges. That is why the curve needs to be approximated in the next step. The second order polynomial is used to handle the shape of the lower boundary. Instead of the least square method, the iterative weighted least square technique is used. The latter is more resistant to outliers than the former. To simplify further consideration, let the created curve be called y_{A1} . The next step consists of checking whether the approximation was done correctly or not.

2.4 Curve Correction

If an image is highly noised, y_{A1} curve is usually corrected. The correction process is iterative and consists of two steps: verification of function monotonicity and the improvement of the location of initial boundary points.

Verification of Function Monotonicity. The shape of the lower boundary of the mandibular bone in the region of interest, can be described by the non-ascending (the right image side) or non-descending function (the left image side). Points that do not meet this condition are treated as noised. Therefore, they should be removed and replaced by a segment of the neighbor points of y_{IB} . In this way, points of y_{IB} are modified.

Improvement of the Location of Initial Boundary Points. Next steps of the correction process depend on the improvement of the location of y_{IB} points. The idea behind that is that the points of y_{IB} should lie closely to corresponding points of y_{A1} . Therefore, two types of points are searched for a column of an image: points whose upper and lower neighbor belongs to class 3 and class 1 respectively and points whose upper and bottom neighbor belongs to class 2 and class 1 respectively. For each point found, the distance to a point of y_{A1} in a column is computed. The pixels that belong to class 3 are more important than points of class 2. Therefore, the computed distance for points of class 3 is multiplied by $\frac{2}{3}$. The point that is associated with the smallest distance is selected as the new point of y_{IB} in the considered column. The process is repeated for every column.

Once, the new points of y_{IB} were selected, the robust approximation with third order polynomial is carried out (y_{A2}). The approximation correction module is repeated until the absolute error between y_{A1} and y_{A2} is different than zero or maximum number of iteration was not reached.

2.5 Edge Pixels Detection

The points of the obtained curve (y_{A1}) do not correspond to edge points. Therefore, the curve points have to be moved towards the point of the lower boundary. An edge point is associated with maximum or minimum of the first derivative of brightness function. That is why the first extreme on the gradient image is selected for a point of the curve. It is searched in the normal direction to the curve at this point. The process is repeated for each point of the curve.

The found points (Fig. 2a) should create the lower boundary, unfortunately they are not connected to each other. To solve the problem, the robust approximation with the third order polynomial is used (Fig. 2b).

3 Experiment and Results

To evaluate the performance of the described technique, two radiologists drew the lower boundary of the mandibular bone on 34 dental panoramic radiographs.

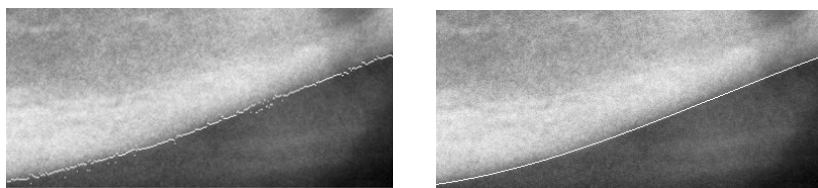


Fig. 2. A - edge points overlapped on region of interest, B - final boundary (approximation with third order polynomial)

The images were provided by the Section of Dental Radiology, Poznan University of Medical Sciences. They were taken using CRANEX TOME dental panoramic unit and a DIGORA PCT PSP DIGITAL SCANNER. The presented technique was implemented using MATLAB. The time of processing of a single image equaled about 30 seconds.

At the beginning, a set of images was divided by a radiologist into the noise-free images (17) and the cluttered images (17). An image was classified as a cluttered one if undesired structures (e.g. shadows) overlapped the region of interest, otherwise the image was classified as noise-free. Next, two radiologists drew the lower boundaries which are treated here as the ground-truth curves. The performance of the described technique is based on the comparative analysis that was carried out in comparison with the above hand-drawn boundaries. The measurements were based on the computation how many boundary pixels generated by the program are placed in a specific region around the ground-truth boundary (the output values are expressed in percentages). The used term "region" refers to area related with parameter $d \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. It is limited by two curves that are the translation of the ground-truth curve by d pixels downwards and upwards. To make the considerations more clear, there is an example in Fig. 3 which demonstrates the area for $d=3$. The measurements were carried out for each image, afterwards the obtained results were averaged separately both for the set of noise-free images and for the set of cluttered images. The outcomes of comparative analysis both between radiologists and between radiologists and program were presented in Table 1 and in Fig. 4. The results

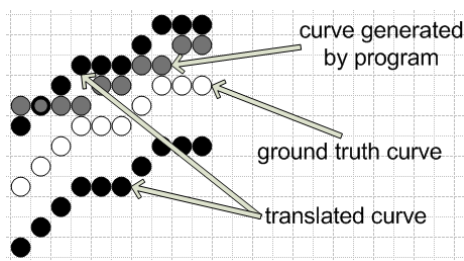


Fig. 3. The illustration demonstrates how the comparative analysis was performed. In the above situation there are $\frac{9}{10}$ (90%) of points that are placed in the region for $d=3$.

obtained are equivalent with the differences between boundaries hand drawn by the radiologists.

Table 1 contains information about mean point to point differences between manual annotations done by radiologists and the boundary model generated by program. The table also presents the accuracy of the ASM approach reported by [3]. The outcomes suggest that the proposed algorithm performs better. However, they cannot be compared directly. Allen et al. [3] used greater set of images whose resolution were different (7.69 pixels/mm) than in the present study (11.76 pixels/mm). Furthermore, the worse results of ASM technique are not surprising, because they refer to both the upper and lower boundary. The extraction of the upper margin is more difficult, because it exists in a region of lower contrast.

Table 1. Point to point differences (mm) for the whole set of images

Comparison	mean(std)
Manual (radiologist 1 and 2)	0.05 (0.04)
ASM approach [3]	0.59 (0.54)
Proposed method	0.14 (0.21)

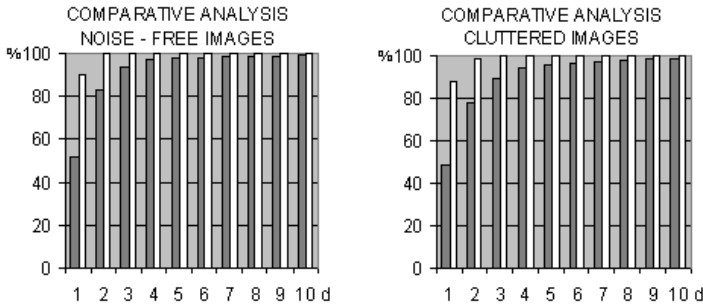


Fig. 4. The comparative analysis between radiologists (white bars) and between program and radiologists (gray bars). The obtained results were presented both for noise-free images (a) and cluttered images (b).

3.1 Final Comments

The outcomes demonstrate that the differences between boundaries drawn by radiologist 1 and radiologist 2 are comparable with the differences between boundaries generated by program and drawn by radiologist 1 and 2. The outcomes are comparable both for the noise-free images and the cluttered images. Therefore, it is possible to extract accurately the lower mandibular boundary based on the described algorithm. The presented technique can be considered as the method of the extraction of the lower mandibular bone boundary alternative to the hand-drawn extraction done by a specialist. It is important to highlight that the described algorithm is robust against the noise that could have been noticed in

the gathered set of images. The described technique can be helpful in the osteoporosis investigation, because it has high accuracy and can be exploited as a first step in the cortical width measurement and examination of the morphology of the mandibular bone.

Acknowledgments

The author would like to thank Marta Dyszkiewicz from Poznan University of Medical Sciences who drew the lower boundary of the mandibular bone. I would also like thank Professor Elżbieta Kaczmarek from Poznan University of Medical Sciences and Professor Marek Domański from Poznan University of Technology for helpful advices.

References

1. Horner, K., Devlin, H., Harvey, L.: Detecting patients with low skeletal bone mass. *Journal of Dentistry* 30, 171–175 (2002)
2. Karayianni, K., Horner, K., Mitsea, A., Berkas, L., Mastoris, M., Jacobs, R., Lindh, C., van der Stelt, P.F., Harrison, E., Adams, J.E., Pavitt, S., Devlin, H.: Accuracy in osteoporosis diagnosis of a combination of mandibular cortical width measurement on dental panoramic radiographs and a clinical risk index (OSIRIS): The OSTEO-DENT project. *Bone* 40, 223–229 (2007)
3. Allen, P.D., Graham, J., Farnell, D.J.J., Harrison, E.J., Jacobs, R., Nicopolou-Karayianni, K., Lindh, C., van der Stelt, P.F., Horner, K., Devlin, H.: Detecting Reduced Bone Mineral Density From Dental Radiographs Using Statistical Shape Models. *Transaction on Information Technology in Biomedicine* 6, 601–610 (2007)
4. Arifin, A.Z., Asano, A., Taguchi, A., Nakamoto, T., Ohtsuka, M., Tsuda, M., Kudo, Y., Tanimoto, K.: Computer-aided system for measuring the mandibular cortical width on dental panoramic radiographs in identifying postmenopausal women with low bone mineral density. *Osteoporosis International* 17, 753–759 (2006)
5. Canny, J.: Computational approach to edge detection. *Transactions on Pattern Analysis and Machine Intelligence* 6, 679–698 (1986)
6. Marr, D., Hildreth, E.: Theory of edge detection. *Proceedings of the Royal Society of London - Biological Sciences* 207, 187–217 (1980)
7. Mortensen, E.N., Barrett, W.A.: Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing* 60, 349–384 (1998)
8. Mortensen, E.N., Barrett, W.A.: Toboggan-Based Intelligent Scissors with a Four-Parameter Edge Model. *Computer Vision and Pattern Recognition* (2), 452–458 (1999)

Football Player Detection in Video Broadcast

Sławomir Maćkowiak, Jacek Konieczny, Maciej Kurc,
and Przemysław Maćkowiak

Chair of Multimedia Telecommunication and Microelectronics
Poznań University of Technology, Polanka St. 3, 60-965 Poznań, Poland
{smack,jkonieczny,mkurc}@multimedia.edu.pl,
przemyslaw.mackowiak@gmail.com
<http://www.multimedia.edu.pl>

Abstract. The paper describes a novel segmentation system based on the combination of Histogram of Oriented Gradients (HOG) descriptors and linear Support Vector Machine (SVM) classification for football video. Recently, HOG methods were widely used for pedestrian detection. However, presented experimental results show that combination of HOG and SVM is very promising for locating and segmenting players. In proposed system a dominant color based segmentation for football playfield detection and a 3D playfield modeling based on Hough transform is introduced. Experimental evaluation of the system is done for SD (720×576) and HD (1280×720) test sequences. Additionally, we test proposed system performance for different lighting conditions (non-uniform pith lightning, multiple player shadows) as well as for various positions of the cameras used for acquisition.

1 Introduction

Many approaches to football video segmentation and interpretation have been proposed in recent years. One of the most interesting techniques are shape analysis-based approaches used to identify players and ball in the roughly extracted foreground [1,2] or techniques based on different classification schemes to segment football players and identify their teams [3,4,5]. However, most of existing approaches assume specific conditions such as fixed or multiple cameras, single moving object, and relatively static background. In football video broadcast, those strict conditions are not met. First, cameras used to capture sport events are not fixed and always move in order to follow the players. Secondly, the broadcasted video is a set of dynamically changed shots selected from multiple cameras according to broadcast director's instructions. Third, there are numerous players moving in various directions in the broadcasted video. Moreover, the background in sports video changes rapidly. These conditions make detection and tracking of players in broadcasted football video difficult. Therefore, future approaches should aggregate different techniques to detect and track the objects in football video.

In this paper a novel approach to football broadcast video segmentation is proposed. We develop a hybrid segmentation system which uses a dominant

color based segmentation for football playfield detection, line detection algorithm based on the Hough transform to model the playfield and a combination of Histogram of Oriented Gradients (HOG) descriptors [15] with Support Vector Machine (SVM) classification [16] to detect players. The system is designed to detect location and orientation of the playfield as well as detect and track players on the playfield. Nevertheless the system is still under development and not all of the features are implemented.

Original contribution of this paper is to apply low complexity techniques with significant potential, until now, used mostly for pedestrian detection. Therefore, the aim of this thesis is to explore the aptitude of the above methods and verify if proposed approach is sufficient for the purpose of segmentation of football video broadcast.

2 Previous Work

In order to create a complex football video segmentation system several types of techniques need to be incorporated. Concerning the system presented in this thesis the following techniques were selected as the most important ones. One of techniques used for dominant color detection is MPEG-7 dominant color descriptor (DCD), however, it operates on three dimensional color representation and its results are not illumination independent [6]. Approach [7] is based on Euclidean distance to trained dominant color in IHS color space. Ren et al. [8] presented an image block classification method based on color hue variance followed by hue value classification by trained Gaussian mixture model. Most of line detection algorithms are based on Hough transform of binary line image [9] which can detect presence of a straight line structure and estimate its orientation and position. Some other approaches use modified Hough transforms like probabilistic Hough transform [10] or Block Hough transform [11] for computation speed improvements. Thuy et al. [12] proposed Hough transform modification which allows line segment detection instead of straight line presence. On the other hand, random searching methods might be also used. Such methods [13] incorporate a random searching algorithm which selects two points and checks whether there is a line between them. Another issue is line image generation. Here, edge detection approaches and other gradient based techniques perform best [14].

Object detection is always based on extraction of some characteristic object features. Dalal et al. [15] introduced a HOG descriptor for the purpose of pedestrian detection and achieved good results.

Another important issue in object detection is object classification which separates objects belonging to different classes to distinguish requested objects from the others. One of the most commonly used object classifiers is SVM classifier which has been successfully applied to a wide range of pattern recognition and classification problems [16,17]. The advantages of SVM compared to other methods are: 1) better prediction on unseen test data, 2) a unique optimal solution for training problem, and 3) fewer parameters.

3 System Overview

To handle specific conditions in segmentation of the football video broadcast a dedicated system for football player detection was proposed (Fig. 1). The main operations in the system are: playfield detection, 3D playfield model fitting, object region recognition and object tracking module.

3.1 Playfield Detection

Information about playfield area position is crucial for further line detection, player detection and tracking. First, we assume that playfield is a homogenous region with relatively uniform color hue. Therefore the first step of playfield detection is vector quantization of color chrominance values. Quantized colors are then classified based on a-priori green color definition and their joint coverage area is considered as initial playfield mask. The playfield area is supposed to be the largest green area in the whole image.

3.2 3D Playfield Model Fitting

Playfield lines are detected using modified Hough transform [11] applied to binary line image. First step is a playfield lines image generation according to modified algorithm from [14] followed by thresholding and morphological thinning. Next, generated image is divided into blocks. For each block line parameters are estimated for voting procedure in modified Hough transform algorithm. Initial line candidates are obtained by searching Hough transform parameter space for local maxima. Then, candidates are refined using linear regression. Refinement stage is crucial for accuracy and false detection rejection. Candidates with too large regression error are rejected. Finally, candidates are aggregated based on their orientation and position. To achieve temporal consistency line candidates are estimated for each frame independently and aggregated with lines from previous frames. We use predefined goal area model with appropriate dimension

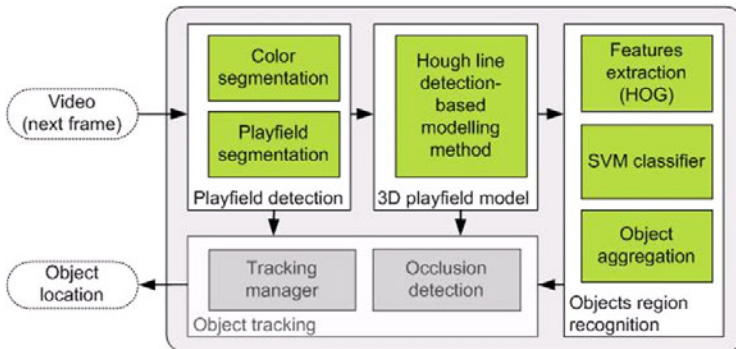


Fig. 1. Detection system overview

proportions as a template. For each frame every possible template position is considered. Each template line is verified against playfield lines on input frame after transformation to playfield space and a fitting error is computed. Finally, playfield template position with smallest fitting error is selected.

3.3 Object Region Recognition

Our player detection module is based on HOG descriptor [15]. In order to represent player shape we use window size of 16×32 pixels. HOG Descriptor block size is 8 pixels, cell size equals 4 and block stride is also 4. Number of angle bins is set to 9. We use unsigned gradient and L2Hys [15] block normalization scheme. Detection is performed using multiple input image scales. Multiple scale approach ensures proper detection of players regardless of their distance to the camera. We assume that smallest player size must be at least 16×32 pixels and the largest can be at most half of the image height.

HOG descriptors are classified by linear SVM algorithm trained on player template database. Our player template database contains over 600 vertical frontal and vertical profile poses as positive examples and over 3000 negative vertical, non-player images. Positive templates were manually generated, negative examples were obtained manually and by bootstrapping procedure. We trained three SVM classifiers working in parallel in order to detect different poses of players: first one was trained on images with vertical frontal poses, second on vertical profile poses and the last on joint set of all vertical poses. All SVM classifiers were using the same negative sample set. If the SVM detector classifies analyzed image area as belonging to a player class, its location is marked using a rectangular area, called a bounding box.

Eventually, the considered object detection system uses a post-processing stage which aggregates resultant player bounding boxes from all SVM classifiers in order to increase the number of detected objects, decrease false detections rate and improve segmentation precision. As we observed, a single player can cause multiple detections at a time and, hence, the resultant bounding boxes from single player detector overlap in many cases. In order to overcome this problem an additional merging operation is proposed, which consists of the following stages: detection box filtering (boxes of not appropriate size or containing too many playfield pixels are rejected), overlap test for each pair of detected boxes (includes finding the biggest coherent area of non-playfield points in each box and testing if these areas overlap) and box aggregation (two overlapping boxes are merged into one resultant box).

3.4 Object Tracking

Object tracking improves the system robustness in case of player occlusion and focus changes produced by a rapid camera movement or zoom. However, the tracking module is currently under intensive development and therefore is not used in presented revision of our system.

4 Evaluation of HOG+SVM Player Detection Method

In this section a combination of HOG and SVM techniques is evaluated as a player detection method for our system. As the analyzed player detection algorithm should be independent from the input video resolution, performance of the system was evaluated on both SD (720×576) and HD (1280×720) test sequence resolution. Additionally, the created test set takes into consideration different lighting conditions (non-uniform playfield lighting, multiple player shadows etc.) as well as the different position of cameras used for acquisition. We selected 9 test sequences with football events and length of 25 to 50 frames for the evaluation. To perform the evaluation step the ground truth regions need to be selected manually for each frame in the test set.

The considered system is evaluated using the precision and recall performance metrics, defined as follows:

$$precision = TP / (TP + FP) \quad (1)$$

$$recall = TP / (TP + FN) \quad (2)$$

where: TP is the set of true positives (correct detections), FP - the set of false positives (false detections) and FN - the set of false negatives (missed objects) defined as:

$$TP = \{r | r \in \mathbf{D}: \exists g \in \mathbf{G}: s_0(r, g) \geq T\}, \quad (3)$$

$$FP = \{r | r \in \mathbf{D}: \forall g \in \mathbf{G}: s_0(r, g) < T\}, \quad (4)$$

$$FN = \{r | r \in \mathbf{G}: \forall g \in \mathbf{D}: s_0(r, g) < T\}. \quad (5)$$

$s_0(a, b)$ is called a degree of overlap between two regions a and b (i.e. bounding boxes of detected objects):

$$s_0(a, b) = (a \cap b) / (a \cup b). \quad (6)$$

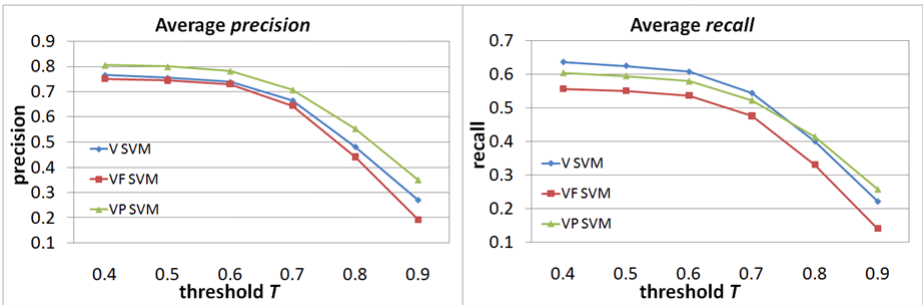
T is a threshold defining the degree of overlap required to determine two regions as overlapping. The set of ground truth regions G and detected regions D for a given frame are defined as: $G = \{g_1, \dots, g_n\}$ and $D = \{d_1, \dots, d_m\}$, with n - the number of ground truth regions and m - the number of detected regions in analyzed frame.

For the purpose of this thesis the analyzed system is evaluated with threshold values T equal 0.4 to 0.9 using three different Support Vector Machines (SVM), namely: vertical (V SVM), vertical frontal (VF SVM) and vertical profile (VP SVM) (see Section 3.3). Average results for threshold $T = 0.4$ to 0.9 are illustrated in Fig. 2. Table 1 presents detailed evaluation results for threshold $T = 0.6$.

Analysis of Table 1 show that selected test sequences provided diverse difficulty level for analyzed player detectors as the *precision* and *recall* metric values differ remarkably among the test set. The average *precision* values (Fig. 2) reached by the examined SVMs are between 0.19 to 0.80 depending on the threshold T parameter used for evaluation and the average recall metric values do not exceed 0.64 with a noticeable decrease for larger threshold T values. The impact of the threshold T on the evaluation results will be discussed in detail later in this section.

Table 1. Detection evaluation results for threshold $T=0.6$ (*P-precision*, *R-recall*)

Test sequence	VF SVM		VP SVM		V SVM	
	P	R	P	R	P	R
1. fast camera pan, uniform lighting	0.92	0.60	0.92	0.66	0.92	0.67
2. non-uniform lighting, shadows, occlusion	0.48	0.40	0.65	0.63	0.57	0.60
3. occlusion, various players poses	0.56	0.46	0.54	0.44	0.49	0.44
4. non-uniform lighting, occlusion	0.68	0.41	0.73	0.51	0.73	0.50
5. non-uniform lighting, occlusion	0.68	0.53	0.82	0.62	0.75	0.61
6. interlaced, uniform lighting	0.97	0.99	0.93	0.99	0.87	0.98
7. motion blur, small figures, occlusion	0.65	0.39	0.71	0.39	0.72	0.47
8. motion blur, occlusion, uniform lighting	0.75	0.47	0.86	0.37	0.77	0.54
9. green uniforms, occlusion	0.90	0.59	0.86	0.62	0.82	0.65
Average	0.73	0.54	0.78	0.58	0.74	0.61

**Fig. 2.** Average values of *precision* and *recall* metrics for threshold $T=0.4$ to 0.9

Evaluation results presented above show clearly that image database used for training of analyzed SVM detectors need to be extended in order to increase the recall value especially. Despite insufficient size of the database there is a number of further improvements which should also help in reducing the missed objects rate for presented detection system. First, we observe that the playfield detection algorithm often classifies green and white player uniforms as a playfield which is an obvious mistake. Next, size of the resultant bounding boxes from the aggregation algorithm should be determined more accurately, as the playfield model fitting can provide additional information about expected size of a player depending on his or her location in analyzed image. This information, together with application of an additional tracking algorithm should also be useful for improving the recall metric of presented system in case of player occlusion and temporary focus changes produced by a rapid camera movement or zoom.

Another crucial improvement is increasing the number of parallel player detectors used in the system and aggregating their results to produce a single output. This step includes both increasing the number of SVM classifiers to detect e.g.

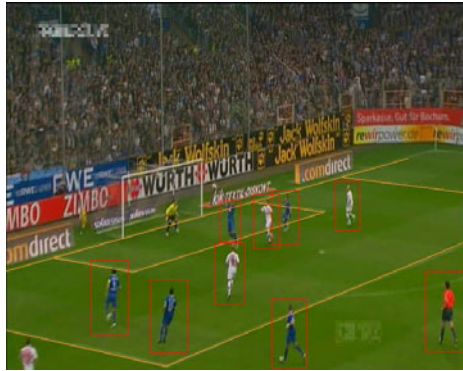


Fig. 3. Player detection and 3D playfield fitting results

horizontal player poses as well as application of other detection methods based on object color and shape. Above step is planned in future release of the system.

We now turn to the discussion on the impact of the threshold T parameter on the object detection system evaluation. Taking into consideration the evaluation result presented in Table 1, we believe that threshold values T larger than 0.6 are too restrictive for analyzed system. Our research show that applied evaluation metrics turn out to be very sensitive for bounding boxes inaccuracies, especially for small objects detection. In case the object size is 16×32 points, decreasing the size of the bounding box only by 1 pixel in each dimension, which seems to be rather slight inaccuracy, decreases the $s_0(a,b)$ (eq. 6) by about 0.1. This issue influences the results even more if we consider the fact that the ground truth bounding boxes usually do not retain a fixed aspect ratio and the detection system produces bounding boxes with one constant aspect ratio, which is our case. Large bounding box inaccuracies may result only by the above fact and not the detector mistakes. On the above analysis it can be concluded that the most reasonable approach would be the threshold T parameter adaptation depending on the size of detected object, e.g. in range of 0.4-0.8. Moreover, in case of presented evaluation results, we can observe a clear saturation of both precision and recall metric values for $T=0.6$ (Fig. 2), which turns to be the most reliable result confirmed also by the subjective detection results evaluation (Fig. 3).

5 Conclusion

In the paper, a novel approach to football broadcast video segmentation is proposed based on a combination of several techniques used to detect players and playfield: dominant color based segmentation, 3D playfield modeling based on Hough transform, HOG descriptor detection and SVM classification which show potential robustness in case of great inconstancy of weather, lighting and quality of video sequences. A dedicated test set and ground-truth database were created to perform objective evaluation of the player detection algorithm applied in our

segmentation system and to explore the aptitude of selected methods. Results show that there are some works deserving further research in proposed approach. Image database used for training of analyzed SVM detectors need to be extended as well as the number of parallel detectors used in the system. Our future work will also focus on dealing with occlusion issue for player detection and tracking.

References

1. Haiping, S., Lim, J.h., Tian, Q., Kankanhalli, M.S.: Semantic labeling of soccer video. In: Proc. of IEEE Pacific-Rim Conf. on Mult. ICICS-PCM, pp. 1787–1791 (2003)
2. Huang, Y., Llach, J., Bhagavathy, S.: Players and Ball Detection in Soccer Videos Based on Color Segmentation and Shape Analysis. In: Sebe, N., Liu, Y., Zhuang, Y.-t., Huang, T.S. (eds.) MCAM 2007. LNCS, vol. 4577, pp. 416–425. Springer, Heidelberg (2007)
3. Nuñez, J.R., Facon, J., Brito Junior, A.d.S.: Soccer Video Segmentation: referee and player detection. In: 15th Int. Conference on Systems, Signals and Image Processing, IWSSIP 2008, pp. 279–282 (2008)
4. Vandenbroucke, N., Ludovic, M., Postaire, J.-G.: Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. *Computer Vision and Image Understanding* 90, 190–216 (2003)
5. Guangyu, Z., Changsheng, X., Qingming, H., Wen, G.: Automatic multi-player detection and tracking in broadcast sports video using support vector machine and particle filter. In: Int. Conf. Multimedia & Expo., pp. 1629–1632 (2006)
6. Hong, S., Yueshu, W., Wencheng, C., Jinxia, Z.: Image Retrieval Based on MPEG-7 Dominant Color Descriptor. In: ICYCS, pp. 753–757 (2008)
7. Ying, L., Guizhong, L., Xueming, Q.: Ball and Field Line Detection for Placed Kick Refinement. In: GCIS, vol. 4, pp. 404–407 (2009)
8. Ren, R., Jose, J.M.: Football Video Segmentation Based on Video Production Strategy. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 433–446. Springer, Heidelberg (2005)
9. Candamo, J., Kasturi, R., Goldgof, D., Sarkar, S.: Detection of Thin Lines using Low-Quality Video from Low-Altitude Aircraft in Urban Settings. *IEEE Trans. on Aerospace and Electronic Systems* 45(3), 937–949 (2009)
10. Guo, S.Y., Kong, Y.G., Tang, Q., Zhang, F.: Hough transform for line detection utilizing surround suppression. In: Int. Conf. on Machine Learning and Cybernetics (2008)
11. Yu, X., Lai, H.C., Liu, S.X.F., Leong, H.W.: A gridding Hough transform for detecting the straight lines in sports video. In: ICME (2005)
12. Thuy, T.N., Xuan, D.P., Jae, W.J.: An improvement of the Standard Hough Transform to detect line segments. In: ICIT (2008)
13. Jiang, G., Ke, X., Du, S., Chen, J.: A straight line detection based on randomized method. In: ICSP (2008)
14. Li, Q., Zhang, L., You, J., Zhang, D., Bhattacharya, P.: Dark line detection with line width extraction. In: ICIP (2008)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition* 1, 886–893 (2005)
16. Yu-Ting, C., Chu-Song, C.: Fast Human Detection Using a Novel Boosted Cascading Structure With Meta Stages. *IEEE Trans. on Img. Proc.* 17, 1452–1464 (2008)
17. Paisitkriangkrai, S., Shen, C., Zhang, J.: Performance evaluation of local features in human classification and detection. *IET Computer Vision* 2, 236–246 (2008)

Spectrum Evaluation on Multispectral Images by Machine Learning Techniques

Marcin Michalak^{1,3} and Adam Świtoński^{2,3}

¹ Central Mining Institute, Plac Gwarków 1, 40-166 Katowice, Poland

Marcin.Michalak@gig.eu

² Polish-Japanese Institute of Information Technology

Aleja Legionów 2, 41-902 Bytom, Poland

³ Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

{Marcin.Michalak,Adam.Switonski}@polsl.pl

Abstract. Multispectral pictures of skin are considered as the way of detection of regions with tumor. This article raises the problem of post-processing of the color spectrum for the improvement of the tumor region detection accuracy. As the reference point spectra of 24 model colors were aquired and then compared with their original spectra. Difference between the original and aquired spectra motivated the authors to use data mining nonparametrical techniques to find the measured spectra postprocessing technique. Two different approaches are described: classificational and regressional.

1 Introduction

Typical representation of color brings to several values, known also as the color model (e. g. RGB, HSV, etc.). Multispectral analysis of images should provide more information about particular pixels. This article raises the problem of post-processing of aquired color spectrum.

It is a common problem, that measurements of spectra of the same color points may give different result because of lightening intensity difference, the distance from camera or the surface quality. In the case, when the real spectra for some colors are given, it should be possible to describe the dependence between the real point (color) spectrum and the observed one. This task is the main target of the article: to find the methodology of obtained color spectrum postprocessing.

Two different approaches were examined: classificational and regressional. In the classificational one there exists kind of spectra dictionary. For each specimen colour, now considered as the specific class, there exists the specimen spectrum. The measured spectrum is classified to the one of classes (specimen colors) and the classification result points the output spectrum in the dictionary.

In the experiments the GretagMacbethTM ColorChecker was used as the specimen of 24 colors (Fig. [1](#)). For every color its 36 channel spectrum was given: from 380nm to 730nm, with the 10nm step. The set of these spectra were called the spectra dictionary.

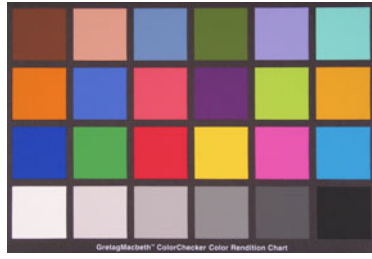


Fig. 1. Colors from the ColorChecker

Color spectra were acquired with the acquiring device containing the endoscope, liquid crystal tunable filter VariSpec and AndorLuca monochrome camera. For every color 21 channel picture was taken as the set of 21 pictures from monochrome camera. Finally, for every color from 1142 to 2130 21 channel spectra were obtained (from 400nm to 720nm, with the 16nm step).

First part of this article describes theoretical backgrounds of used data mining tools. Then the object of the interest is described: classificational and regressional approach to the spectrum postprocessing. The next part shows the results of experiments and is followed by the final conclusions.

2 Methods of Classification and Nonparametrical Regression

Kernel estimators are ones of the simplest and probably the clearest examples of nonparametric estimators. The Nadaraya–Watson estimator is defined in the following way:

$$\tilde{f}(x) = \frac{\sum_{i=1}^n y_i K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)} \quad (1)$$

where $\tilde{f}(x)$ means the estimator of the $f(x)$ value, n is a number of train pairs (x, y) , K is a kernel function and h the smoothing parameter.

One of the most popular is the Epanechnikov kernel [4] (Table 1), where $I(A)$ means the indicator of the set A . Other popular one and multidimensional kernel function are presented in the Table 1. V_q denotes the volume of the q -dimensional unit sphere in the \mathbb{R}^q space:

$$V_q = \begin{cases} \frac{(2\pi)^{q/2}}{2 \cdot 4 \cdot \dots \cdot q} & \text{for } q \text{ even} \\ \frac{2(2\pi)^{(q-1)/2}}{1 \cdot 3 \cdot \dots \cdot q} & \text{for } q \text{ odd} \end{cases}$$

The second step of the kernel estimator creation is the selection of the smoothing parameter h . As it is described in [7] the selection of h is more important than the selection of the kernel function. Small values of the h cause that the estimator fits data too much. Big values of this parameter h lead to the estimator that oversmooths dependencies in the analysed set.

The most popular method of evaluation the h parametr is the analysis of the approximation of the Mean Integrated Square Root Error ($MISE$), after some generalization leads to the following value of h : $h_0 = 1,06 \min(\tilde{\sigma}, \tilde{R}/1,34)n^{-\frac{1}{5}}$. Details of derivations can be found in [7].

Table 1. Popular kernel functions

kernel type	onedimensional	multidimensional
Epanechnikov	$K(x) = \frac{3}{4}(1-x^2)I(-1 < x < 1)$	$K(x) = \frac{q+2}{2V_q}(1- x)$
Uniform	$K(x) = \frac{1}{2}I(-1 < x < 1)$	$K(x) = \frac{1}{V_q}$
Triangular	$K(x) = (1- x)I(-1 < x < 1)$	$K(x) = \frac{q+1}{V_q}(1-\sqrt{ x })$
Biweight	$K(x) = \frac{15}{16}(1-u^2)I(-1 < x < 1)$	$K(x) = \frac{(q+2)(q+4)}{V_q}(1- x)^2$
Gaussian	$K(x) = \frac{1}{\sqrt{2\pi}} \exp -u^2/2$	$K(x) = (2\pi)^{-q/2} \exp(- x /2)$

Support Vector Machines (SVM) were defined in [1] as a tool for the classification and later became also the regression tool [3].

In general case of SVM the classification function has the form: $f(x) = \text{sgn}(\sum_{i=1}^l y_i \alpha_i K(x, x_i) + b)$ where α are Lagrange multipliers derived as the solution of minimization problem $\langle w, w \rangle/2 + C \sum_{i=1}^l \xi_i$ with some constraints: $y_i(\langle w, K(x_i) \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0$.

In general case of SVM regression, the regression function has the form: $f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i)K(x, x_i) + b$ where α, α^* are Lagrange multipliers from the dual optimization problem of minimizing $\langle w, w \rangle/2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$ with constraints $y_i - \langle w, K(x_i) \rangle - b \leq \epsilon + \xi_i, -y_i + \langle w, K(x_i) \rangle + b \leq \epsilon + \xi_i^*, \xi_i, \xi_i^* \geq 0$. The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated. Typical kernel functions used in multivariate SVM are shown in the Table 2.

Table 2. Popular SVM kernel functions

Linear	$K(u, v) = u'v$
Polynomial	$K(u, v) = (\gamma u'v + c)^d$
RBF	$K(u, v) = \exp(-\gamma u - v ^2)$
Sigmoidal	$K(u, v) = \tanh(\gamma u'v + c)$

3 Object of Interest

The goal of the work, presented in this article, is to find the mapping between the observed (measured) color spectrum and its real one. This mapping may be performed as the classification or the regression task. Let us define some notions, that will be helpful in the further part of the article: **specimen** - the real spectrum of the colour from the GretagMacbeth ColorChecker; **sample** - the measured spectrum from the ColorChecker; (**color**) **profile** - the aggregated

information about all samples from the same colour of the ColorChecker; (**color**) **mapping vector** - the vector of multipliers for a single color.

The authors present two different approaches to the observed spectrum mapping and preprocessing. The first approach is based on the classification methods and consists of assigning sample to the one of known colors, which mapping vectors are given. This approach was performed with the usage of the *SVM* and *kNN* classifiers. The second approach is based on the regression methods. For every component of the spectrum the nonparametrical estimator is created, that points the estimation of the real value of the spectrum component. This approach was performed with the usage of kernel estimators and *SVR*.

3.1 Color Profile and Mapping Vector Definitions

When spectrums of all samples from the same color are shown on the same figure (Fig. 3.1) it occurs that this color profile should present the aggregated information about all samples. The profile of the color is the vector of average amplitudes for every spectrum component. If the number of samples (x) from the c -th color is cn , then the profile π_c is calculated as: $\pi_c = \frac{1}{cn} \sum_{i=1}^{cn} x_i$. Additionally, also the integral of the profile (I_π), considered as the sum of components averages, can be considered as the part of the color profile (kind of its reference point): $I_\pi = \sum_{i=1}^C \pi^{(i)}$. C means the number of components and $a^{(i)}$ is the i -th component of the vector. The color mapping vector μ_c (Fig. 2) for the c -th color is defined as the vector of quotients of color profile π_c and its specimen s_c from the spectra dictionary: $\mu_c = \pi_c ./ s_c$. The operator $./$ is analogical to the Matlab operator and means the array right division (component by component quotient of vectors).

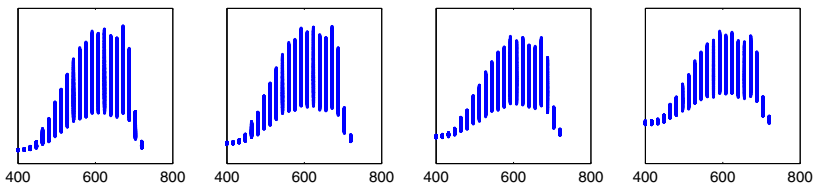


Fig. 2. Scatters of spectrums for four colors

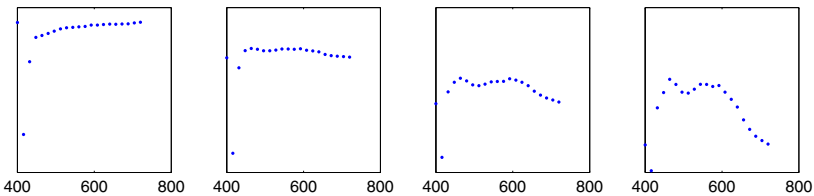


Fig. 3. Color mapping vectors for four colors

3.2 Classificational Approach

Classification approach is the simplest model of spectrum mapping and can be evaluated in two similar ways. The common step of the both of version consists of building the classifier on the basis of the train set. Then, for the certain test spectrum classification is performed and the result class label points the proper spectrum from the spectra dictionary as the basis spectrum. For the sample spectrum x , $c(x)$ is the color that is the result of classifying the x and is the result of the first step.

The second step may be performed in two ways. In the first variant the original spectrum estimator $Sp(x)$ depends only on the classification result and may be expressed as $Sp(x) = x / \mu_{c(x)} = x \cdot s_{c(x)} / \pi_{c(x)}$. The operator \cdot means array multiplication. The second variant of the second step is connected with the integral of the color profile and integral of the sample. The result of the classification points the spectrum from the spectra dictionary. Then it is multiplied by the fraction of the sample integral and the color profile integral: $Sp(x) = I_x \cdot s_{c(x)} / I_{\pi_{c(x)}}$.

The quality of the classificational postprocessing is measured as the classification quality and also as the difference between the specimen spectrum and the postprocessed measured spectrum. Accuracy, defined as the fraction of the number of correctly classified test objects and the number of test objects, is the measure of classification quality. The quality of the postprocessed spectrum is based on the difference between the specimen and observed spectrum. For every specimen channel the squared absolute error is calculated. Then the square root of the average channel error describes the total regression error. This leads to the definition of the root of the mean value of the squared absolute error: $RMSAE = [n^{-1} \sum_{i=1}^C ((\tilde{c}_i - c_i) / (c_i))^2]^{1/2}$.

3.3 Regressional Approach

The second approach is based on the assumption that there exists some unknown regression function, that can map the observed spectrum to the real one (close to the specimen). The regression function may describe how the c -th color of the specimen depends on the c -th color of original spectrum. This model is called "1 vs 1". On the other hand, also the dependence between the c -th color of the specimen from all measured colors (the whole spectrum) may be described by the regression function. We call this model "1 vs all". This model may be also extended, including the integral of the profile or the integral of the color respectively. The extended model is called "1 vs all (ext.)".

Fig. 3.3 shows the dependence between the input and output value of channel amplitude for some colors. It can be seen that also the linear and nonlinear mappings between channels occurs in the data. In the regressional approach, on the basis of the train set a nonparametrical regression function estimator is built. For every channel of the spectrum the separate regressor is created, which returns the estimate of the output spectrum channel. The quality of the regressional approach is also defined with $RMSAE$.

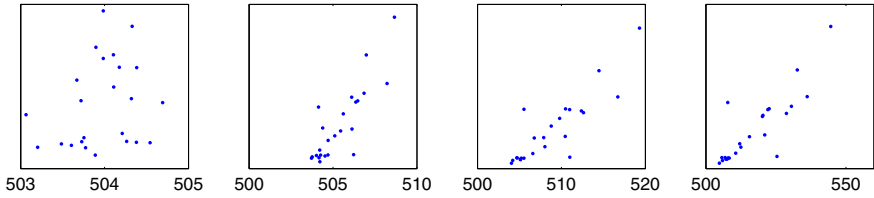


Fig. 4. Dependence between the input and output value of amplitude in spectrum for four colors

4 Experiments and Results

The data set contained 31 456 observed spectra from 24 colors. Four of them (1, 2, 3, 4) were represented by 2130 instances, one (5) by 1238 instances and the rest of them by 1142. Because the domain of specimen spectra (36 channels) was different from the domain of sample spectra (21 channels), specimen spectra were mapped into the 21 dimensional space with the usage of linear interpolation.

All experiments were performed with 10 cross validation scheme and all tables show averaged results: the accuracy and *RMSAE* for the classificational approach and only the *RMSAE* for the regressional approach. The cross validation division was the same in each experiment, so it is ensured that subexperiments (each of 10 iterations) were performed on the same data sets. *kNN* experiments were performed with WEKA [6]. *SVM* regression experiments were performed with SVM and Kernel Methods Matlab Toolbox [2] and classification experiments with LIBSVM [5].

Table 3 shows the accuracy of classifying spectrums to colors with the usage of *kNN* algorithm. Experiments were performed with several values of *k* parameter. Table 4 shows the accuracy of predicting the color of the spectrum using Support Vector Machine and the *RMSAE* error in correctly classified spectrums after the postprocessing. The *kNN* algorithm is not included in the comparison because it occurred to be a little bit worse than *SVM*.

Table 3. Classification results for kNN classifier

k	acc	k	acc	k	acc	k	acc	k	acc
1	91,14%	4	91,49%	7	92,41%	10	92,41%	100	91,79%
2	90,17%	5	92,20%	8	92,41%	20	92,61%	200	91,22%
3	91,59%	6	92,10%	9	92,42%	50	92,29%	500	90,07%

Last table (Table 5) shows the *RMSAE* of the kernel estimation of spectra. Three algorithms were used: Nadaraya–Watson estimator (NW) and Support Vector Machine (*SVR*) with two types of kernel (gaussian and polynomial). Three models of regressional approach were considered: "1 vs 1", "1 vs all" and "1 vs all (ext.)".

Table 4. Results for *SVM* classification approach: *acc* – classification accuracy, *RMSAE* of estimated spectrum without (a) and with (b) the usage of the spectrum integral as a additional variable

kernel type	<i>acc</i>	(a)	(b)
linear	95,28 %	3,34%	3,01%
polynomial	95,26 %	3,34%	3,01 %
RBF	95,09 %	3,28%	2,95 %

Table 5. Results for regressors

model	regressor		
	NW	SVR	
		gaussian	polynomial
1 vs 1	80,38%	114,23%	118,04%
1 vs all	100,00%	130,62%	915,42%
1 vs all (ext.)	100,00%	147,05%	898,93%

5 Conclusions and Further Works

The main target of performed experiments was to improve spectrums that came from multispectral camera, make them closer to real values and remove the error included during the aquisition step. There were two different approaches: classificational and regressional.

First of them gave relatively good result: over 90% accuracy classification for *kNN* algorithm and over 95% for Support Vector Machine. Considering the better of these two algorithms, the error of spectrum estimation was very small and took the value around 3%. The usage of the spectrum integral as the additional spectrum variable decreased the spectrum estimation error insignificantly. Although these results may be considered as very satisfactory it must be emphasized, that this method is applicable only when new colors, capured by the mutispectral camera, are from the same subset of colors that were used as the train set. In other words, the method presented has a very poor ability of scaling for new or similar colors.

The second approach – regression methods – solves the problem of classification approach nonscalability. For every component of the spectrum a nonparameter regression function is created, that explains the difference between the original color spectrum component and measured by the camera. This approach is more scalable than the classificational one because it makes it possible to use the model for colors that were not used in the train step. Although the model satisfies the criterion of scalability, it is not satisfactory as far as the results are concerned. For the simplest kernel estimator (Nadaraya–Watson) the absolute regression error level is very high and takest the value of 80% for „1 vs 1” model and 100% for two other models.

The nature of that error comes from the problem of empty neighborhood for test objects. Let us define the object *u* neighborhood as the set of objects *v* that

satisfy condition $K(\langle u, v \rangle) > 0$. For the values of smoothing parameter h that are too low it is possible that all test objects will have empty neighborhoods. The nature of the Support Vector Machine regression error is hard to explain at this moment. The error measured in hundreds of percent proves, that the regressor overestimates real spectrum values. As *SVR* has data dependent parameters (like smoothing parameter) it may be possible that for the specific kind of data, like color spectra, a modified algorithm of this parameter evaluation should be developed.

The conclusions above suggest to continue the research on the development of regression approach of spectral postprocessing. The analysis of the regression approach error should give the answer, why that error is so big and how to improve regressor parameters evaluation. Afterwards, the influence of the spectrum postprocessing on the tumor region detection accuracy will be examined.

Acknowledgments

This work was financed from the Polish Ministry of Science and Higher Education resources in 2009-2012 years as a research project.

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proc. of the 5th Annu. Workshop on Comput. Learn. Theory, pp. 144–152 (1992)
2. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and Kernel Methods Matlab Toolbox, Perception Systemes et Information, INSA de Rouen, Rouen, France (2005), <http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html>
3. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.N.: Support vector regression machines. In: Adv. in Neural Inf. Process. Syst., vol. IX, pp. 155–161 (1997)
4. Epanechnikov, V.A.: Nonparametric Estimation of a Multivariate Probability Density. Theory of Probab. and its Appl. 14, 153–158 (1969)
5. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training SVM. J. of Mach. Learn. Res. 6, 1889–1918 (2005)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
7. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall, Boca Raton (1986)

Fatigue Detector Using Eyelid Blinking and Mouth Yawning

Helmi Adly Mohd Noor and Rosziati Ibrahim

Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
adly@uthm.edu.my, rosziati@uthm.edu.my

Abstract. Matlab is a well-known tool for processing images. It provides engineers, scientists, and researchers with an intuitive, flexible environment for solving complex imaging problems. For measurement of human's fatigue level, the images for the eyelid blinking or mouth yawning can be processed using Matlab in order to exhibit that the human is fatigue or not. This paper discusses the algorithm that combines 2 factors (eyelid blinking and mouth yawning) for measurement of human's fatigue level. The purpose of combining these 2 factors is to get a better measurement of the level of human fatigue due to drowsiness. The process and activities in this algorithm are elaborated in details as a guide and reference to build a prototype using Matlab programming language software. Insight acquired through this study is expected to be useful for the development of simulation. This research and invention will be a technological solution to address some human problems such as accident prevention and safety for transportation used and also for educational area.

Keywords: Image processing, Fatigue detection, Eyelid blinking, Yawning theory.

1 Introduction

Providing a systematic algorithm to measure human's fatigue level using eyelid blinking and mouth yawning is the main purpose of this study. This study is expected to be useful for the development of simulation for measurement of human's fatigue level using these two factors. From this research, we are going to carry out Malaysia agenda on research and development activities in bioinformatics technology.

Research and development (R&D) in bioinformatics technology is currently an agenda of Malaysian government. This research combines image processing and artificial intelligence techniques. Such findings will be utilized to develop innovative applications that can benefit Malaysians. Therefore, this research discusses the algorithm that combines 2 factors (eyelid blinking and mouth yawning) for measurement of human's fatigue level.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 proposes the framework for measuring human's fatigue level using eyelid blinking and mouth yawning. Section 4 describes experimental result. Section 5 concludes the paper and suggests future work for the research.

2 Literature Review

The word fatigue is used in everyday living to describe a range of afflictions, varying from a general state of lethargy to a specific work-induced burning sensation within one's muscles. Physiologically, "fatigue" describes the inability to continue functioning at the level of one's normal abilities due to an increased perception of effort [1,2,3].

Fatigue is ubiquitous in everyday life. Fatigue has two known forms; one manifests as a local, muscle-specific incapacity to do work, and the other manifests as an overall, bodily or systemic, sense of energy deprivation. Due to these two divergent facets of fatigue symptoms, it has been proposed to look at the causes of fatigue from "central" and "peripheral" perspectives. Fatigue can be dangerous when performing tasks that require constant concentration, such as driving a vehicle. When a person is sufficiently fatigued, he or she may experience micro sleeps (loss of concentration) [4]. Fatigue also includes mental fatigue, not necessarily including any muscle fatigue. Such a mental fatigue, in turn, can manifest itself either as somnolence (decreased wakefulness) or just as a general decrease of attention, not necessarily including sleepiness. In any case, this can be dangerous when performing tasks that require constant concentration, such as driving a vehicle. For instance, a person who is sufficiently somnolent may experience microsleeps.

We found that many researchers on fatigue detection is based only on one factor: eyelid blinking. For example Eriksson *et al.* [5] develop a system that tracks the eyes of a driver. The system is using a small camera that used to monitor the face of the driver and her/his eye movement which indicate that the driver is no longer in condition to drive. The system can also determine if the eyes are open or closed.

Generally, eyes can be detected using two steps: locating the face and then eyes detection. Wenhui *et al.* [6] suggest a vision-based real-time driver fatigue detection method. It uses the characteristics of skin colours. Then, after the eyes were detected, it detects the distance change of eyelid that indicates whether the driver is fatigue or not. The team from Carnegie Mellon Research Institute [7] has considered two drowsiness detection methods. The first is a video-based system that measures the percentage of time a driver's eyes are closed (PERCLOSE). The second detection method is using a non-parametric (neural network) model to estimate PERCLOSE using measures associated with lane keeping, wheel movements and lateral acceleration of the vehicle.

Bretzner *et al.* [8] proposed a Smart Eye AntiSleep system that measures 3D head position, head orientation, gaze direction and eyelid closure. It is a compact one-camera system specially designed for automotive incabin application. Besides eye movement, mouth movement can also can be used as driver's fatigue indication. Wang *et al.* [9] proposed to locate and track the driver's mouth movement to detect his/her level of fatigue or inattention with one vehicled-mounted camera, and offer the driver with a safe assistant.

Mandalapu *et al.* [10] proposed a method to locate and track a driver's mouth using cascade of classifiers training and mouth detection. This method also

trained the mouth and yawning images using SVM. Finally, SVM is used to classify the mouth regions to detect yawning then alert fatigue.

Recently, there are many previous research discussed related topic about image processing to detect human’s fatigue. Currently, a systematic algorithm has been developed for measuring human’s fatigue level using one factor only such as face or head movements, yawning and eyelid blinking. However, the response time is still low due to a single factor used compared to merging of two factors. This paper proposes an algorithm by using two factors in one fatigue level and we found the response time is faster for measurement of human’s fatigue level rather than using one single factor.

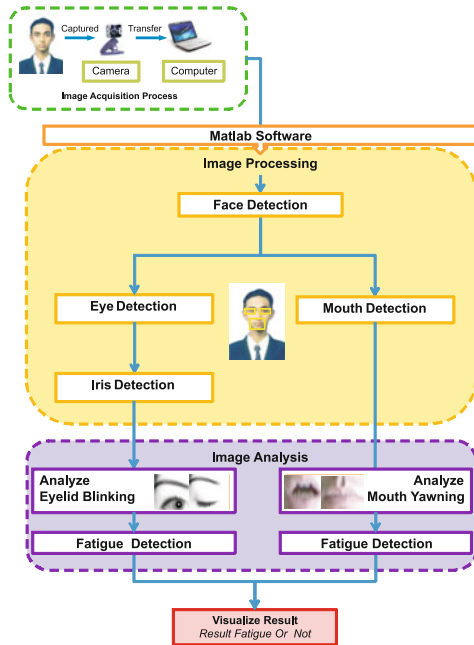


Fig. 1. Image Processing Using Eyelid Blinking and Mouth Yawning to Measure Human’s Fatigue Level Framework

3 Proposed Framework

In this study, we proposed our own framework to detect the human’s fatigue level. From the framework in Figure 1, these 2 levels (eyelid blinking and mouth yawning) of human’s fatigue are combined into one framework to measure the human’s fatigue level. This systematic framework provides better understanding of developing an algorithm of the prototype. The end result combines the eyelid blinking and mouth yawning to draw the conclusion whether the human exhibit a fatigue level or not.

We present an algorithm of detecting driver fatigue by analyzing the rate of eyelid blinking and mouth yawning by calculated the area opening. Basically, the approaches contain four main phases; Image Acquisition Process, Image Processing, Image Analysis and Visualize Result.

The process begins with image acquisition process. In this process, face image captured by using camera, and then it will transfer to the computer. Image acquisition process is running in this phase in Matlab software. In the Matlab software, the process begins with Image processing phase. The third phase is image analysis. In this phase, the program will analyze the rate of eyelid blinking and mouth yawning by calculated the area opening. The end result will visualize result whether the human is fatigue or not. Figure 2 shows the flowchart of the system.

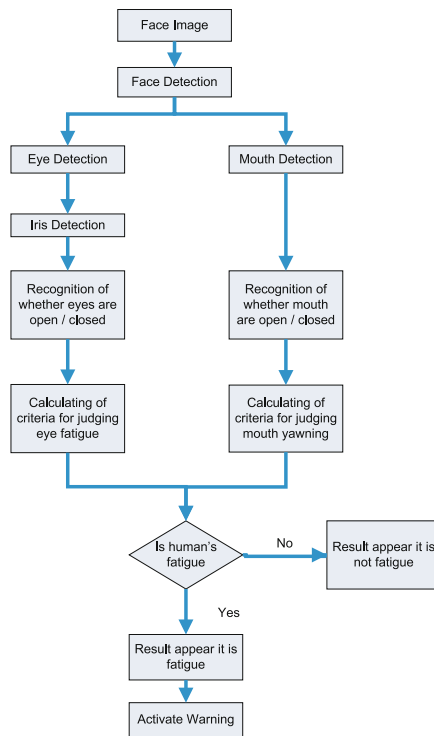


Fig. 2. Flow chart of the system



Fig. 3. Example result of the eye blink and mouth yawning detection

The last phase is visualizing the result. In this phase, the end result will visualize whether the human is fatigue or not. Video files which have been processed will be produced as shown in Figure 3 and three graphs will be shown after the program finishes processing the videos.

Figure 4 shows eye blink frequency level graph. This graph shows the eye blink within 30 seconds. This graph is a binary graph, which the value of x axis is a frame of videos within 30 seconds, while y axis is an eye blinking movements which the value is 1 if both eyes are closed and 0 if eyes are opened.

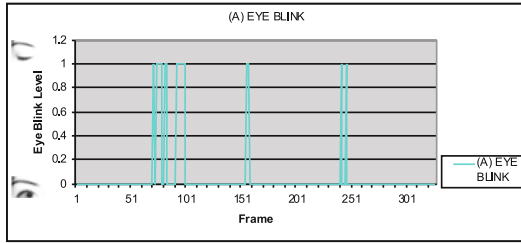


Fig. 4. Eye blink frequency level graph

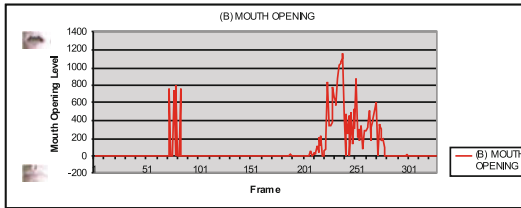


Fig. 5. Mouth opening frequency level graph

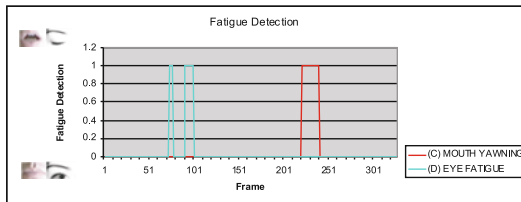


Fig. 6. Eye blink & mouth yawning fatigue frequency level graph

Second graph is mouth opening frequency level graph as shown in Figure 5. This graph shows the level of mouth opening. The value of this graph in x axis is a frame of videos within 30 seconds, while y axis is calculated the pixel of mouth opening.

The last graph is a combination of eye blink and mouth yawning fatigue detection graph as shown in Figure 6. This graph is a binary graph, which the value of x axis is a frame of videos within 30 seconds. While y axis is eye fatigue and yawning detected. If the value is 1, eye fatigue and yawning is detected. From here, total of both fatigue detection is visualize in a single graph.

```

***** RESULT *****
Eye blinks : 8 times
Drowsy eye blinks : 1 times
Microsleep detected: 1 times
Sleep detected: 0 times

Eye fatigue detected: 2 times
Mouth Yawning detected: 1 times
Time first fatigue is detected: second 5 (frame 125)
Total fatigue detected: 3 times

Processing time: 6.81 minutes
Results: FATIGUE is detected.
    
```

Fig. 7. Human's is fatigue or not result

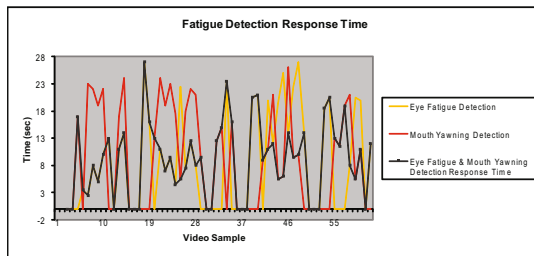


Fig. 8. Fatigue detection response time results

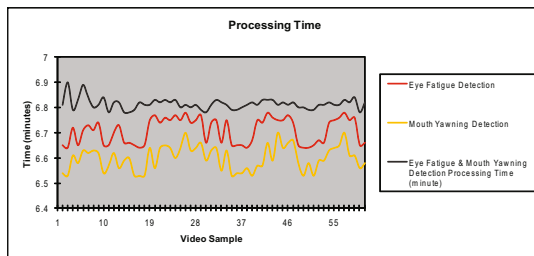


Fig. 9. Time processing results

The end result appears after the program is finished processing the video. Figure 7 shows that the overall results from the experiment conducted. The result shows the total number of detected eye blink, drowsy eye blink, microsleep,

sleep and mouth yawning. Fatigue level is detected due to eye fatigue and mouth yawning detection found.

4 Experimental Results

In this section, the performance comparison between measure human's fatigue level using single factor (eyelid blinking or mouth yawning) versus two factor (combination of eyelid blinking and mouth yawning) is evaluated. Single factor and two factors response time of the fatigue detected is compared. Response time is determined to check how long it takes at the earliest time to detect human's fatigue. Time processing is also compared with single factor and two factors to determine the duration of each algorithm to complete the processing.

4.1 Fatigue Detection Response Time

Response time is determined to check how long it takes to detect human's fatigue as early as possible. From this experiment, the results show that by using two factors, the earliest fatigue detection can be found for these two factors which come first. Rather than using single factor, it is limited to detect the earliest one factor such as eye fatigue or mouth yawning. Figure 8 shows the fatigue detection response time results.

4.2 Single and Two Factor Time Processing

Time processing is check how on long it takes to process from the first frame to the end. Time processing also compared with single factor and two factors to determined duration of each algorithm to complete the processing. From this experiment, the results show that by using two factors, the processing time is increased. Although the processing time is increased a little bit, but the response time is much better in order to detect human's fatigue as early as possible. Figure 9 shows the time processing results.

5 Conclusion

In this paper, the main contribution of this research is introducing a new algorithm in measuring human's fatigue level by using 2 factors (eyelid blinking and mouth yawning). The experimental results show that the proposed algorithm is an efficient approach in terms of response time accuracy as compared with a single factor level only. With these new algorithm approaches, the value of measurement of human's fatigue level by using 2 factors has better performance in terms of response time accuracy. Hence, further development and experiments in the future will be focusing on 3 factors level or 4 factor level combining eyelid blinking, mouth yawning, head motion and red eye effect. This project can be used for accident prevention and safety for transportation and also for educational area.

Acknowledgement

This work was supported by the FRGS under the Grant No. Vote 0271, Ministry of Higher Education, Malaysia.

References

1. Gandevia, S.C.: Some central and peripheral factors affecting human motoneuronal output in neuromuscular fatigue. *Sports medicine (Auckland, N.Z.)* 13(2), 93–98 (1992); PMID 1561512
2. Hagberg, M.: Muscular endurance and surface electromyogram in isometric and dynamic exercise. *Journal of applied physiology: respiratory, environmental and exercise physiology* 51(1), 1–7 (1981); PMID 7263402
3. Hawley, J.A., Reilly, T.: Fatigue revisited. *Journal of sports sciences* 15(3), 245–246 (1997); PMID 9232549
4. Enoka, R.M., Stuart, D.G.: Neurobiology of muscle fatigue. *Journal of Applied Physiology* 72, 1631–1648 (1992)
5. Eriksson, M., Papanikolopoulos, N.P.: Eyetracking for detection of driver fatigue. In: *IEEE International Conference on Intelligent Transportation Systems, ITSC 1997, IEEE Conference on 9-12*, pp. 314–319 (1997)
6. Dong, W., Wu, X.: Fatigue detection based on the distance of eyelid. In: *IEEE International Workshop on VLSI Design and Video Technology*, pp. 365–368 (2005)
7. Grace, R., Byrne, V.E., Bierman, D.M., Legrand, J.M., Gricourt, D., Davis, B.K., Staszewski, J.J., Carnahan, B.: A drowsy driver detection system for heavy vehicles. In: *Proceedings of the 17th DASC, Digital Avionics Systems Conference*, vol. 2, pp. 136/1–136/8. AIAA/IEEE/SAE (1998)
8. Bretzner, L., Krantz, M.: Towards low-cost systems for measuring visual cues of driver fatigue and inattention in automotive applications. In: *IEEE International Conference on Vehicular Electronics and Safety*, pp. 161–164 (2005)
9. Rongben, W., Lie, G., Bingliang, T., Lisheng, J.: Monitoring mouth movement for driver fatigue or distraction with one camera. In: *The 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 314–319 (2004)
10. Devi, M.S., Bajaj, P.R.: Driver Fatigue Detection Using Mouth and yawning Analysis. *IJCSNS International Journal of Computer Science and Network Security* 8(6), 183–188 (2008)

Inferior Maxillary Bone Tissue Classification in 3D CT Images

Silvia Moreno¹, Sandra L. Caicedo¹, Tonny Strulovic³, Juan C. Briceño²,
Fernando Briceño³, Soledad Gómez², and Marcela Hernández Hoyos^{1,2}

¹ IMAGINE: Computación Visual, Universidad de Los Andes, Bogotá, Colombia
sc.moreno43@uniandes.edu.co
<http://imagine.uniandes.edu.co/>

² Grupo de Ingeniería Biomédica, Universidad de Los Andes, Bogotá, Colombia

³ Centro de Investigaciones Odontológicas, Pontificia Universidad Javeriana,
Bogotá, Colombia

Abstract. This paper presents a method for segmenting the inferior maxillary bone in CT images and a technique to automatically classify bone tissue without requiring a training stage. These methods are used to measure the mean density of seven main anatomical zones of the mandible, making the difference between cortical and cancellous bone. The results lead to determine the normal density values in each region of the inferior maxillary bone and help to evaluate the success of the bone regeneration process. The proposed method was validated on ten axial slices from different zones of a patient mandible, by comparing automatic classification results with those obtained by expert manual classification. A 4% mean difference was found between percentages of bone tissue types, and the mean difference between mean density values was of 88 HU. Once the method was validated, it was applied to measure density in the seven anatomical zones of the inferior maxillary bone.

1 Introduction

Dental implants are currently one of the best options to replace missing teeth. However, when no appropriate measures are taken after the teeth loss, bone reabsorption may occur. This would cause an atrophy of the jaw that would difficult the placement of dental implants. In order to reconstruct the jaw bone structure, bone grafts have been used. The Small Intestinal Submucosa (SIS) is a material derived from the porcine small intestine. When SIS is employed as a surgical implant, it induces the patient to regenerate its own damaged tissue without being rejected by the host immune system [1]. The results of recent research in this area have found that the implantation of SIS scaffolds with bone cells in the oral cavity does induce bone regeneration. This advance can lead to better treatment solutions when there is atrophy of the jaw bone. In order to increase the knowledge of bone regeneration effectiveness and progress, it is interesting to study this process through Computed Tomography (CT) images. One objective of this study would be to analyze what type of bone tissue can be

found in the original injury area and in what proportion. These percentages and density values would be compared to normal values, so this information would provide an indicator of the success of the procedure. There are two types of bone tissue: cortical bone and cancellous bone. The compact or cortical bone is found forming the outer shell of bone, whereas cancellous or spongy bone usually occupies the internal region. Cortical bone has greater hardness and stiffness, which makes this tissue denser than cancellous bone.

Previous work has been proposed to determine the density values in regions of the maxillary bone. However, the studied regions are quite extensive, and little is known about the total 3D density of the anatomical zones of the inferior maxillary bone, or the bone tissues proportion in each one of them. In order to support the study of bone regeneration, we want to measure the normal density of the mandible in seven main anatomical zones: condyle, neck, coronoid process, ascending ramus, angle, basal region of the body, and basal mental region. On each zone the tissue is identified as cortical or cancellous bone, determining the percentage of each type and their mean density in Hounsfield Units (HU). In this paper we propose a semi-automatic method to segment the inferior maxillary bone in CT images and automatically classify bone tissue in the mandibular anatomical zones. The remainder of this document is organized as follows: Section 2 of this document presents a summary of the previous work in image processing related to mandible segmentation and tissue differentiation. Section 3 describes the details of the developed method. Section 4 shows the validation process and the results obtained by applying the method to CT images. Finally, section 5 presents the conclusions and suggestions for future work.

2 Previous Work

Previously there have been works that have proposed methods for segmenting the mandible in order to obtain 3D models. These models can be used in simulation or surgical planning. In several of these algorithms, the selection of a threshold is included as part of the segmentation to separate the inferior maxillary bone from other tissues. Different techniques have been applied to determine this threshold. In the method proposed by Tognola et al [2], an histogram equalization is performed and the maximum histogram point is selected as the threshold. Barandiaran et al. [3] apply the Otsu method for multiple thresholds with three classes to obtain the threshold that separates the lower jaw from the rest of the image. Then, a region growing technique is used to obtain the connected component formed by the brightest spots. The purpose of these two methods is to obtain the denser outer shell of the bone. This outer shell is useful to generate the 3D model of the mandible, and frequently these methods do not segment the internal region. In the process suggested by Krsek et al. [4] a fixed density range of 1200 to 4000 HU is utilized for segmenting the outer shell of the inferior maxillary bone. Then, an adapted 3D line seedfill algorithm is applied to obtain the cancellous bone. With this algorithm a better segmentation of the internal tissue of the bone is achieved, but the range of density values used may exclude cortical bone areas which are less dense. None of the previous work emphasizes on the automatic differentiation of bone tissue.

Regarding the classification of bone tissue, Futterling et al. [5] developed a model of the human mandible for finite element analysis, in which seven different materials are assigned to bone tissue depending on the gray levels presented in the CT image. In this work, the ranges of value are uniform throughout the lower jaw, and location is not taken into account to classify bone tissue. In the work of Rueda et al. [6] automatic segmentation of cortical and cancellous bone is achieved by applying active appearance models. The model requires a training stage, in which landmarks have to be manually placed to identify the contours of the regions corresponding to cortical bone, cancellous bone and the dental nerve in multiple images.

The method proposed in this work uses a fixed density range to segment the mandible in order to reduce the computational cost of this stage of the process. This is followed by an additional processing to ensure that both outer and internal regions of the bone are obtained. Finally, an automatic classification of bone tissue is performed, that does not require a training stage, and that takes into account both intensity and location to determine the tissue type.

3 Method

The developed method for the automatic differentiation of bone tissue in the mandible consists of two main steps: inferior maxillary bone segmentation and bone tissue classification.

3.1 Inferior Maxillary Bone Segmentation

The first stage of the process is aimed to extract from the image the volume that corresponds to the mandible (Figure 1). The process begins with the bone outer shell segmentation, which eliminates other tissues that can be found in CT images. To achieve this goal, a seed point located in the cortex of the mandible is manually selected and a 3D region growing technique is applied [7], using as criterion the point gray level. In order to estimate the range of values that allows to segment the outer shell of the jaw, a study on density values of 45 profile lines was performed, and the threshold of 850 HU was selected. The next step in the segmentation process is to extract the internal tissue of the bone. For this

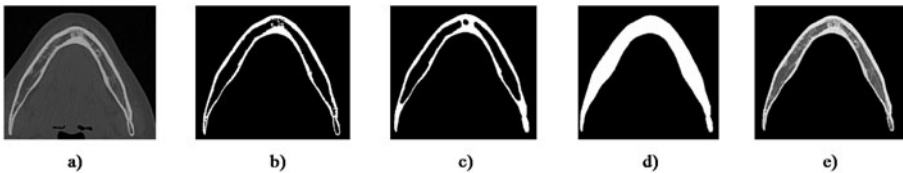


Fig. 1. Example of segmentation in an axial view, a) original image, b) image after outer shell segmentation, c) image after closing operation d) image after 3D ray casting, e) segmented image with original gray values

purpose, a morphological operation of closing is applied. As a result, holes in the previously segmented surface are filled.

Next, in order to segment the internal region of the bone, an algorithm of 3D ray casting is applied to determine which points lie inside of the outer shell that has already been extracted. The ray casting scheme used by the algorithm is based on the model proposed by Lorenz and von Berg [8]. According to this model, a ray casting scheme has 4 elements. The ray generator determines the pattern followed by the rays (circular, spherical, parallel etc.). The ray sampler, that captures the intensity values and calculates the directional derivative along the ray. The ray analyzer examines the sampled values and evaluates a certain criterion. Finally, the ray result is obtained when the analyzer criterion is met. Generally, the ray result of one ray is not considered as a definitive result, but as a candidate. This candidate can be evaluated later with the results of the other rays, to determine the final result. In the ray casting scheme of the algorithm, six rays are casted from every voxel in the main directions of the x, y and z axes. The ray analyzer of the algorithm seeks to determine if the ray intercepts the surface that was previously segmented. The analyzer checks the points along the ray, looking for a point from the external shell of the bone (white voxels). When it is found, the ray returns a positive result. Finally, the results of all the rays are analyzed, and if all of them returned a positive value, the point is included in the region. As the final part of the segmentation process, the logical operation AND is applied between the extracted region and the original image to obtain the inferior maxillary bone in original gray levels.

3.2 Bone Tissue Classification

The second stage of the proposed method seeks to determine the type of bone tissue, classifying voxels as cortical or cancellous bone. A Fuzzy c-means algorithm [9] is applied on a manually selected volume of interest, to separate the image into three classes. This cluster analysis technique divides the image into classes or clusters, assigning to each point of the image a membership grade to a cluster. The membership grade is inversely related to the distance between the intensity value of the point and the gray level of the centroid of the class. The centroid is defined as:

$$centroid_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m} \tag{1}$$

where u_k is the membership grade of x to class k , m is the fuzziness parameter, which in this case a value of 2 was assigned. The degree of membership is calculated as:

$$u_k(x) = \frac{1}{\sum_j \left(\frac{dist(centroid_k, x)}{dist(centroid_j, x)} \right)^{\frac{2}{m-1}}} \tag{2}$$

The first class is formed by the points of higher intensity which correspond to cortical bone. The second class contains lower intensity points which belong to cancellous bone. Finally, the third class is formed by points with intermediate gray values which are considered undetermined tissue (Fig. 2). The reason for

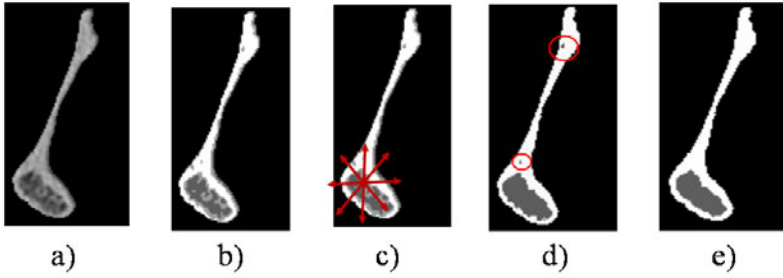


Fig. 2. Classification in an axial view a) original image, b) classification into cortical bone (white), cancellous bone (dark gray) and undetermined tissue (light gray), c) pattern followed by 2D ray casting scheme, d) classification into cortical and cancellous bone using 2D ray casting, e) result of the adapted median filter

segmenting the image into three classes and not two is that there are structures with considerably high intensity, such as trabeculae, that may be mistakenly classified as cortical bone. According to their physical location, these structures should be part of cancellous bone. To identify the areas of undetermined tissue another algorithm is then applied. This algorithm is based in a ray casting scheme (see section 3.1), that analyzes the location of the voxels classified as undetermined to determine the type of bone tissue they belong to.

From every point of intermediate intensity level, eight rays are casted on the axial slices in a 2D radial pattern. The ray analyzer of the scheme examines the points along the rays, searching for the first pixel with a non-intermediate intensity level. When the ray analyzer finds a point that meets this criterion, the intensity level of the pixel is returned as the result. Finally, the results of all the rays are examined, and if a ray intercepted the background (black pixel), the point is considered to be located in the external shell of the bone. In this case the point is classified as cortical bone. Otherwise, it is classified as cancellous bone. An adapted median filter is then applied to remove isolated pixels. This

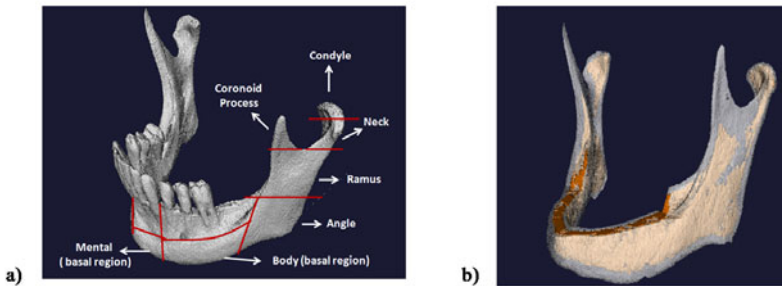


Fig. 3. a) Anatomical zones of the mandible. The alveolar zone was not considered in this study, b) 3D representation of the mandible classification.

filter evaluates only the points that are initially classified as undetermined, and ignores the pixels of the image background in order to preserve the edges. Finally, the inferior maxilla is classified into cortical and cancellous bone (Fig. 3).

4 Experiments and Results

4.1 Method Validation

For the validation of the proposed method, the percentage of bone tissue type and the mean density value were calculated by automatic classification and compared to the values obtained from manual expert classification (Fig. 4). An expert was asked to draw contours delimiting the regions corresponding to cortical and cancellous bone on ten axial images of different anatomical zones of the mandible of one patient (Fig. 5). The tests were made on regions corresponding to the ascending ramus, the condyle, the neck, the coronoid process and the angle. For every image, ImageJ [10] was used to calculate the mean density and the percentage of every bone tissue. To evaluate the automatic classification, an implementation of the proposed method was developed on C++ with VTK [11] libraries for the processing of the 3D images. BBTk [12] was employed to generate the interaction interfaces of the program. This tool was used to automatically classify the bone tissues by anatomical zone, and the measurements of interest were made on the selected axial images.

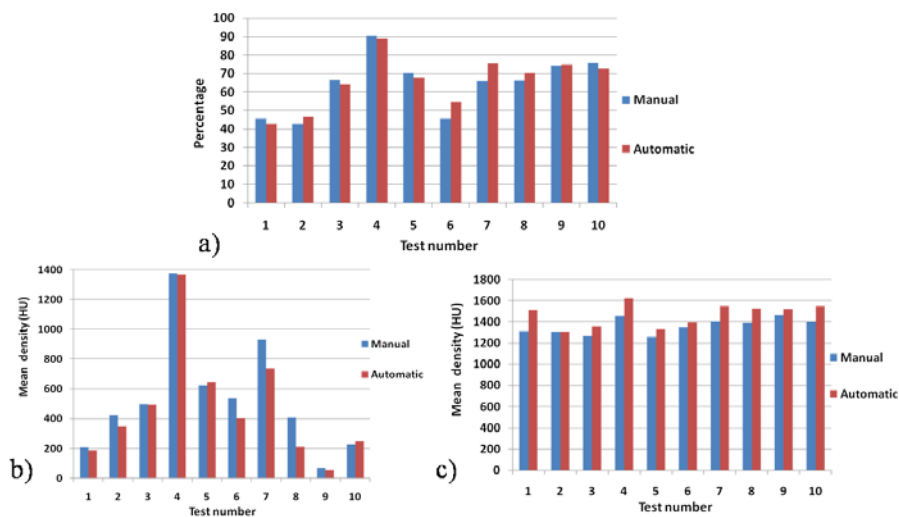


Fig. 4. Validation results, a) percentage of cortical bone measured by manual and automatic classification, b) mean density in HU of cancellous bone measured by manual and automatic classification, c) mean density in HU of cortical bone measured by manual and automatic classification

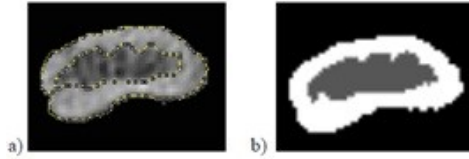


Fig. 5. Validation example, a) manual classification, b) automatic classification

4.2 Measurement of Density Values

Once the method was validated, it was applied on CT images of four hemimandibles (two patients), with the purpose of analyzing the density in the anatomical zones of the mandible (Fig. 3). The percentage of each bone tissue, the mean density value and the standard deviation were measured (Table 1). The volumes of interest for each region were manually selected by an expert.

Table 1. Measurements in the mandibular anatomical zones

Zone	Mean Percentage(%)		Mean Density(HU)		Standard Deviation(HU)	
	Cortical	Cancellous	Cortical	Cancellous	Cortical	Cancellous
Condyle	59	41	1488	420	228	342
Neck	58	42	1552	408	211	348
Coronoid Process	99	1	1781	531	194	248
Ramus	78	22	1616	327	184	471
Angle	55	45	1566	193	205	382
Body	56	44	1560	307	190	380
Mental	71	29	1543	379	188	403

5 Conclusions

A semi-automatic method to segment the jaw bone from CT images and automatically classify bone tissues that facilitates the measurement of the normal density values for cortical and cancellous bone in comparison with the manual process has been presented. In the validation tests the results calculated through manual and automatic classification were compared. A 4% mean difference between the percentages of each bone tissue was obtained, and a 88 HU mean difference was found among the mean density values. It has been observed that with the automatic classification, narrow areas of dense cancellous bone may be mistakenly classified as cortical bone. The estimated density values of the anatomical zones contribute to the analysis of bone density in the inferior maxillary bone. In order to really establish the normal ranges, a greater number of images must be examined, and variables like age and sex must be considered. Future work will attempt to eliminate teeth from the image, so that measurements of the alveolar zone can be taken without wrong values caused by dental structures. Also, a more exhaustive validation of the method will be performed.

References

1. Rickey, F.A., Elmore, D., Hillegonds, D., Badylak, S., Record, R., Simmons-Byrd, A.: Re-generation of tissue about an animal-based scaffold: AMS studies of the fate of the scaffold. *Nucl. Instrum Meth. B* 172(1-4), 904–909 (2000)
2. Tognola, G., Parazzini, M., Pedretti, G., Ravazzani, P., Svelto, C., Norgia, M., Grandori, F.: Three- Dimensional Reconstruction and Image Processing in Mandibular Distraction Planning. *IEEE T. Instrum. Meas.* 55(6), 1959–1964 (2006)
3. Barandiaran, I., Macía, I., Berckmann, E., Wald, D., Dupillier, M.P., Paloc, C., Graña, M.: An Automatic Segmentation and Reconstruction of Mandibular Structures from CT-Data. In: Corchado, E., Yin, H. (eds.) *IDEAL 2009*. LNCS, vol. 5788, pp. 649–655. Springer, Heidelberg (2009)
4. Krsek, P., Spanel, M., Krupa, P., Marek, I., Cernocho, P.: Teeth and Jaw 3D Reconstruction in Stomatology. In: *Proceedings of the International Conference on Medical information Visualisation - Biomedical Visualisation*, Zurich, Switzerland, pp. 23–28. IEEE Computer Society, Los Alamitos (2007)
5. Futterling, F., Klein, R., Straber, W., Weber, H.: Automated Finite Element Modeling of a Human Mandible with Dental Implants. In: *6th International Conference in Central Europe on Computer Graphics and Visualization* (1998)
6. Rueda, S., Gil, J.A., Pichery, R., Alcañiz, M.: Automatic Segmentation of Jaw Tissues in CT Using Active Appearance Models and Semi-automatic Landmarking. In: Larsen, R., Nielsen, M., Sporning, J. (eds.) *MICCAI 2006*. LNCS, vol. 4190, pp. 167–174. Springer, Heidelberg (2006)
7. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Prentice Hall, Englewood Cliffs (2008)
8. Lorenz, C., von Berg, J.: Fast automated object detection by recursive casting of search rays. In: *CARS 2005: Computer Assisted Radiology and Surgery*, vol. 1281, pp. 230–235 (2005)
9. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Cybernetics and Systems* 3(3), 32–57 (1976)
10. ImageJ (2009), <http://rsbweb.nih.gov/ij> (Cited November 30)
11. Kitware, Inc.: The Visualization Toolkit (2009), <http://www.vtk.org> (Cited July 30)
12. CreaTools LRMN (2009) CreaTools Available from, <http://www.creatis.insa-lyon.fr/creatools> (Cited July 30)

Vector Median Splatting for Image Based Rendering

Krzysztof Okarma, Aleksandra Miętus, and Mateusz Teclaw

West Pomeranian University of Technology, Szczecin
Faculty of Electrical Engineering
Department of Signal Processing and Multimedia Engineering
26. Kwietnia 10, 71-126 Szczecin, Poland
okarma@zut.edu.pl

Abstract. In the paper a method of missing data completion based on the vector median filtering is presented, which may be useful as an adaptive splatting algorithm for Image Based Rendering. Presented approach has been verified using some scanned face models rendered using the IBR method with the use of standard splatting and proposed method. Obtained results are promising and may be a starting point for further research related to the adaptive change of the splat weighting function, especially as a support for the real-time face recognition applications.

Keywords: image based rendering, splatting, vector median filter, image warping.

1 Introduction

Image Based Rendering is one of the most useful image synthesis methods, mainly due to its ability of fast synthesis of good quality images, what is an important advantage in many real-time video applications such as e.g. free view-point television. Nevertheless, one of the most relevant limitations of such technique is the necessary knowledge of depth maps acquired for the reference images. It was proposed by Leonard McMillan [4] and is based on the warping equation, which is used for mapping the reference image into the destination plane. A few input parameters comprise the 3D warping method by McMillan: locations and parameters of both (reference and destination) cameras and the depth map acquired for the reference image. Nevertheless, the resolution of the destination image is limited by the amount of useful data present in the reference image (or images). In all the cases when the number of available reference pixels is insufficient, an additional splatting algorithm should be used in order to avoid the presence of holes in the resulting image. The basic version of conventional splatting [10], originating from the volume rendering, is the simple copying of the warped pixel into the neighbouring ones (with additional checking of the depth buffer if multiple reference cameras are used). Taking into account the influence of such simplified splatting on the quality of the resulting image and the the fact

that some other methods are usually developed with a view to the volume rendering applications, the vector median approach, similar to the nonlinear colour image filtering methods, is proposed.

Proposed method may be useful also for some other techniques used for the visualisation of volume data, such as volume rendering or surface rendering [2]. However, they both are more computationally demanding in comparison to IBR and they are not the main topic of this paper. The main difference between volume and surface rendering is that in the first approach no geometry is created, similarly as in IBR. The final image is directly rendered from volumetric data. In surface rendering, on the other hand, first the volumetric data is mapped to an initial geometry (e.g. isosurfaces). Rendering of the image is processed in further steps with standard rendering techniques [5], so the whole procedure may be time consuming.

Recently, a lot of work has been done to create images that may represent translucent objects with important structural details [7]. Volume rendering is widely used to generate medical images of inner organs such as liver or to help assess the level of vein degradation. Thanks to many optimisation techniques a variety of different visualisation results may be achieved.

2 Image Based Rendering

2.1 Overview of the Method

Assuming the pinhole camera model used during the computations, there is no need to take into consideration all the possible aberrations (barrel, chromatic) and all the distortions of camera's sensor or film. All the rays focus in the centre of projection and the linear mapping function may be used, described as [4]:

$$\vec{d} = \begin{bmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a_k & b_k & c_k \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (1)$$

where: \vec{d} represents a ray from the centre of projection to the image plane, $\vec{a} = [a_i \ a_j \ a_k]$ and $\vec{b} = [b_i \ b_j \ b_k]$ are the transposed horizontal and vertical unit vectors, u and v are the point's coordinates in image space, and $\vec{c} = [c_i \ c_j \ c_k]$ is the transposed vector directed to the centre of projection from the image corner (for the minimum values of u and v).

To properly project all the pixels from reference image, additional information such as depth is necessary for each pixel. It can be obtained as a depth map being in fact a greyscale image of the same size as the reference one, seen from the very same view, but carrying different information. It is generated in such a way that the pixels far from the viewer (in infinity) have the highest value. Because the depth map is 8 bit greyscale image, the infinity is seen as pure white, which stands for 255. The darker the pixel, the closer it is to the observer. Thanks to the depth maps, the complete information about object's geometry and size is available.

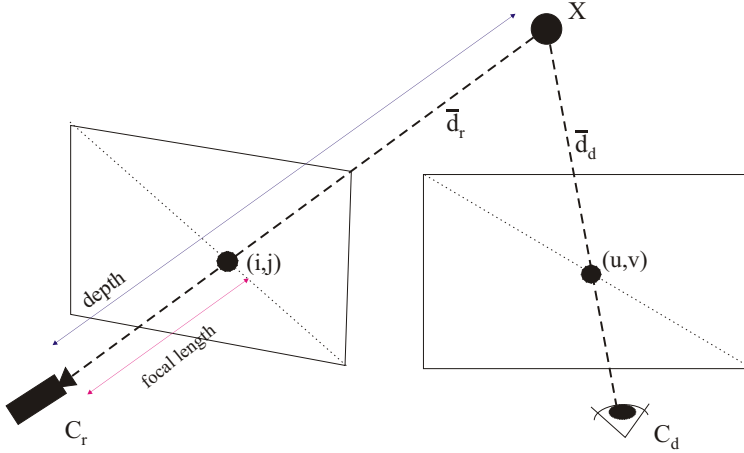


Fig. 1. Illustration of the image planes and centers of projection

Analysing the Fig. 1 it can be noticed that the image coordinates in picture to the left side, i and j , determine the ray passing from a photographed object X , through an image plane, to the reference camera. In real cameras image plane is generated on camera’s sensor or film, where the focal length is defined as a distance between the sensor on which the image is generated and the focus, where all the rays converge. The image plane and focal length are illustrated outside the camera to simplify the visualisation. According to [4] the rays of initial point in X to the reference camera C_r and the destination camera C_d , let’s consider them as \vec{d}_r and \vec{d}_d , are described as:

$$\vec{d}_r = P_r \cdot \vec{x}_r, \quad \vec{d}_d = P_d \cdot \vec{x}_d \tag{2}$$

where \vec{x}_d denotes the point (u, v) where the ray crosses the image plane.

Assuming that the rays having the same direction are equivalent regardless of the length of the three-space vector specifying this direction, using the projective geometry, the simplified planar ray-to-ray mapping equation is received as:

$$P_d \cdot \vec{x}_d = P_r \cdot \vec{x}_r + \delta(\vec{x}_r) \cdot (C_r - C_d) \tag{3}$$

where $\delta(\vec{x}_r)$ is a generalised disparity (projective depth) computed from the ray lengths. The units of disparity depend on the distance between centre of projection and the viewing plane. From practical point of view, the generalized disparity is related to depth by a scaling factor (range).

Finally, after some transformations the planar warping equation can be expressed in the matrix form as:

$$\alpha \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [\bar{a}_d \ \bar{b}_d \ \bar{c}_d]^{-1} \cdot [\bar{a}_r \ \bar{b}_r \ \bar{c}_r (C_r - C_d)] \cdot \begin{bmatrix} i \\ j \\ 1 \\ \delta(i, j) \end{bmatrix} \tag{4}$$

Finally, the rational expressions for warping equation can be obtained, so the destination coordinates for corresponding input pixels can be computed as:

$$u = \frac{W_{1,1} \cdot i + W_{1,2} \cdot j + W_{1,3} + W_{1,4} \cdot \delta(i, j)}{W_{3,1} \cdot i + W_{3,2} \cdot j + W_{3,3} + W_{3,4} \cdot \delta(i, j)} \quad (5)$$

and

$$v = \frac{W_{2,1} \cdot i + W_{2,2} \cdot j + W_{2,3} + W_{2,4} \cdot \delta(i, j)}{W_{3,1} \cdot i + W_{3,2} \cdot j + W_{3,3} + W_{3,4} \cdot \delta(i, j)} \quad (6)$$

where the 4×3 warping matrix W is defined as:

$$W = \begin{bmatrix} \bar{a}_r \cdot (\bar{b}_d \times \bar{c}_d) & \bar{b}_r \cdot (\bar{b}_d \times \bar{c}_d) & \bar{c}_r \cdot (\bar{b}_d \times \bar{c}_d) & (C_r - C_d) \cdot (\bar{b}_d \times \bar{c}_d) \\ \bar{a}_r \cdot (\bar{c}_d \times \bar{a}_d) & \bar{b}_r \cdot (\bar{c}_d \times \bar{a}_d) & \bar{c}_r \cdot (\bar{c}_d \times \bar{a}_d) & (C_r - C_d) \cdot (\bar{c}_d \times \bar{a}_d) \\ \bar{a}_r \cdot (\bar{a}_d \times \bar{b}_d) & \bar{b}_r \cdot (\bar{a}_d \times \bar{b}_d) & \bar{c}_r \cdot (\bar{a}_d \times \bar{b}_d) & (C_r - C_d) \cdot (\bar{a}_d \times \bar{b}_d) \end{bmatrix} \quad (7)$$

2.2 Application of Splatting for Image Based Rendering

Solving the warping equation presented and computing the desired coordinates for a novel view, does not produce satisfying results because of possible errors caused by occlusions and missing data. Another problem with warping equation is rounding of the numbers representing the coordinates, because the input and destination images are not continuous. In the consequence, the empty spaces in the desired image may occur, which can be filled using the splatting or meshing algorithm. Meshing approach treats an image as a mesh of pixels but performs badly at object boundaries and is relatively slow.

Many splatting ideas were presented in various publications, however two main ways of computing the lost pixels may be distinguished. The first one is with the use of depth information, so it is possible when the geometry of the whole scene is known. The second one is seen as postprocessing of desired image, where the holes from rounding should be treated similarly as a noise.

The easiest method in the first attempt is known as basic splatting with the constant splat size. In that method the splat with e.g. 3×3 pixels is drawn instead of a single pixel and the final result is dependent of the drawing order. Further increase of the splat size covers larger empty spaces but the loss of important details may be significant. In order to prevent this the idea of adaptive splatting has been proposed, where the splat size is computed depending on the angle of rotation for the desired image e.g. according to the following equation [6]:

$$size \approx (z_d)^2 \cdot \frac{(d_r)^2 \cdot \cos(\phi_d) \cdot w_d \cdot h_d \cdot \tan(FOV_r/2)}{\cos(\phi_r) \cdot w_r \cdot h_r \cdot \tan(FOV_d/2)} \quad (8)$$

where: z_d is the distance from camera to the look point, d_r is the depth of reference image (in given point), ϕ_r and ϕ_d are the angles between the central ray from the corresponding cameras and the normal to surface being projected, $w \cdot h$ denotes the images' resolutions and FOV is a field of view for both cameras.

Considering the postprocessing approach, various sophisticated techniques can be applied, especially for the volume rendering algorithms, but their applicability

for the real-time IBR is limited. In order to fill the missing pixels with the colour present in the nearest neighbourhood, the adaptive vector median splatting is proposed, which is based on the algorithms used for the multichannel image filtering.

3 Proposed Approach

3.1 Vector Median Filtering Techniques

Multichannel median filtering algorithms are not the straightforward extensions of the greyscale median filter. Standard definition of the median value should not be used independently for each channel, because some new colours could be introduced into the image. Such filtration requires the definition of a new quantity, used for sorting the vectors representing the colours of each pixel within the filter mask in order to find the most typical one. One of the simplest methods is the Vector Median Filter (VMF), proposed by Astola [1], based on the minimum aggregated distance (Euclidean or city-block) between each pixel and the remaining ones within the current sliding window (mask), which can be interpreted as a measure of similarity. Many modifications of such approach have been proposed e.g. combination with the Arithmetic Mean Filter (AMF) known as Extended Vector Median Filter or α -trimmed VMF (α pixels with the lowest aggregated distance are averaged by the AMF).

Another method known as Basic Vector Directional Filter (BVDF) is based on the directional approach using the angular distance defined as:

$$A_i = \sum_{j=1}^N \operatorname{acos} \left(\frac{X_i \cdot X_j}{|X_i| \cdot |X_j|} \right) \quad (9)$$

within the N -element mask where X stands for the three-components vectors in the chosen colour space.

An interesting combination of both approaches is the Distance Directional Filter (DDF) [3] with the following weighted ordering criterion

$$\Omega_i = \left[\sum_{j=1}^N \operatorname{acos} \left(\frac{X_i \cdot X_j}{|X_i| \cdot |X_j|} \right) \right]^{1-k} \cdot \left[\sum_{j=1}^N d(X_i, X_j) \right]^k \quad (10)$$

where k is parameter used for smooth changing in the filter type from VMF ($k = 1$) to BVDF ($k = 0$) and d denotes the Euclidean distance.

3.2 Application of Vector Median Filtering for Splatting

Considering the multichannel filtering techniques as useful for the IBR post-processing, the application of the weighted DDF filter with adaptive change of the mask (splat) size is proposed. The splat size is chosen as a single pixel (no filtering), 3×3 , 5×5 or 7×7 pixels depending on the number of missing pixels

within the mask. The algorithm works only for the pixels representing the holes in order to avoid the loss of useful information.

The adaptive choice of the splat size is based on the calculation of the minimum, maximum and median values within the mask of 3×3 pixels for each of the RGB channels. If at least one of the three differences between the median and minimum values is non-zero and the same condition is fulfilled for at least one of the three differences between the maximum and median values, the splat size is considered as large enough. Otherwise both conditions are checked for the larger splat. The non-background pixels within the chosen splat are treated as the input for the generalized DDF filter and the result (the most typical pixel) is placed as the splat's central one in the resulting image.

4 Experiments and Results

The experiments have been performed using the the pictures of 3D models of heads and faces, acquired by 3D scanning by Cyberware, as input images. The models are full, 360 degrees scans of human (both male and female) heads in PLY format (known as Polygonal File Format or Stanford Triangle Format) consisting of wireframe information about a scan, polygonal model and finally, a high resolution, colourful texture. To generate the input image, 3D Studio Max 2008 has been used along with gw:Ply Importer plugin, created by GuruWare. In order to verify the splatting algorithms, some virtual scenes have been created. A single destination image was being considered for each scene, projected from single reference camera, just to simplify the computations.

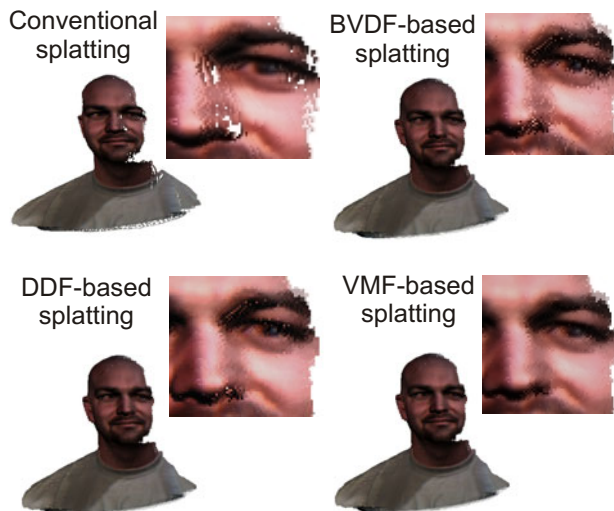


Fig. 2. Comparison of the results obtained using the standard splatting and proposed approach using the three nonlinear filters with unweighted mask (model 1)

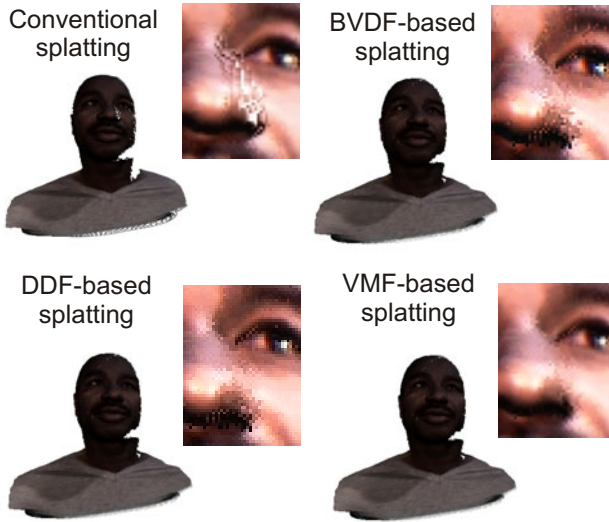


Fig. 3. Comparison of the results obtained using the standard splatting and proposed approach using the three nonlinear filters with unweighted mask (model 2)

The results presented in the paper have been obtained for the three filters: VMF, BVDF and DDF with $k = 0.5$ and compared to the applications of standard splatting. The comparison of obtained results is presented in Figs. 2 and 3 containing zoom of the critical fragments with additionally enhanced contrast to improve the visibility. Analysing presented images the advantages of the VMF-based approach can be easily noticed, especially for zoomed areas with enhanced contrast. Application of the directional approach produces many artefacts caused by improper choice of the most typical pixel so the VMF-based approach is more suitable for splatting purposes. The combination of both methods (DDF) does not lead to better results than the VMF because of the strong influence of the directional factor (higher values of the parameter k should be used to reduce it). Further increase of the image quality can be obtained using more sophisticated filters as a base for the adaptive splatting e.g. with weighted masks. Such approach can be investigated in our future research.

5 Summary

The main advantages of Image Based Rendering are modest rendering cost and its independence on the scene complexity but its main weak side is the need of depth information. Nevertheless, taking into account the computational power, widely available even for the home use, the application of some additional algorithms improving the image quality seems to be an important direction of research. Such improvements may be obtained e.g. by the use of the image fusion methods, super-resolution based techniques, sophisticated interpolation

algorithms or better splatting methods. Application of the Vector Median Filter as an extension for the fast splatting, presented in the paper, seems to be a promising direction of further research in this area.

Therefore, our future work should be related to the application of some modern image quality metrics, such as e.g. Structural Similarity [8] and its modifications [9] for the automatic quality assessment of the results of splatting in order to develop the weighted version of the VMF-based splatting algorithm. Nevertheless, it would require an additional precise image matching because of the sensitivity of many full-reference image quality assessment method on the shifts and rotations in relation to the rendered reference image.

References

1. Astola, J., Haavisto, P., Neuvo, Y.: Vector Median Filters. *Proceedings of IEEE* 78(4), 678–689 (1990)
2. Guan, X., Mueller, K.: Point-based Surface Rendering with Motion Blur. In: *Eurographics Symposium on Point-Based Graphics* (2004)
3. Karakos, D.G., Trahanias, P.E.: Generalized Multichannel Image-Filtering Structures. *IEEE Trans. Image Processing* 6(7), 1038–1045 (1997)
4. McMillan, L.: An Image-Based Approach to Three-Dimensional Computer Graphics. Ph.D. Dissertation, University of North Carolina, Chapel Hill (1997)
5. Rheingans, P.: Expressive Volume Rendering. *Journal of WSCG* 12(1-3), 7–10 (2004)
6. Shade, J., Gortler, S., He, L., Szeliski, R.: Layered Depth Images. In: *Proc. Int. Conf. SIGGRAPH 1998*, pp. 231–242 (1998)
7. Viola, I., Kanitsar, A., Gröller, M.A.: Importance-Driven Volume Rendering. In: *Proc. IEEE Visualization 2004*, pp. 139–145 (2004)
8. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image Quality Assessment: From Error Measurement to Structural Similarity. *IEEE Trans. Image Processing* 13(4), 600–612 (2004)
9. Wang, Z., Simoncelli, E., Bovik, A.: Multi-Scale Structural Similarity for Image Quality Assessment. In: *Proc. 37th IEEE Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA (2003)
10. Westover, L.: Footprint Evaluation for Volume Rendering. *Computer Graphics* 24(4), 367–376 (1990)

Feature Extraction Using Reconfigurable Hardware

Wiesław Pamula

Silesian University of Technology, Krasińskiego 8, 40-019 Katowice, Poland
wieslaw.pamula@polsl.pl

Abstract. Feature extraction is an important stage in image processing for object classification, tracking or identification. Real time processing adds stringent constraints on the efficiency of this task. The paper presents a discussion of a reconfigurable hardware processing architecture, based on components, for performing feature calculations using convolutions, morphology operators and local statistics. Special attention is directed to pipelining calculations, fast determination of minimum, median and maximum of values. The architecture is optimised for video streams, which provide the image contents using horizontal scanning. An implementation using a low cost FPGA is presented proving the feasibility of this approach.

1 Introduction

Reconfigurable hardware has been successfully used for low level image processing algorithms. Advances in integration technology bring the complexity of devices to the level of millions of logic gates. Xilinx's Virtex and Spartan families, Altera's Stratix, Arria and Cyclone families are examples of FPGA devices commonly used for hardware processing [1], [2].

Early hardware solutions used a number of devices configured into processing pipelines controlled by processors, which also organised data transfers from auxiliary memories [3]. The hardware components, due to limited logic resources, were dynamically reconfigured using elaborate schemes for tracking algorithm flow [5], [6].

The organisation of these solutions may be described as hybrid, that is algorithm implementation was decomposed into hardware tasks, run in FPGA devices, and into software based control and data transfer tasks. Efficient use of FPGA resources was accomplished using HDL tools. Efforts were made in using higher level abstraction tools such as Handel-C or SystemC [14], [15].

Current FPGA devices have ample resources to implement complete processing solutions inside single components. The specifics of internal logic organisation prohibit however a simple conversion from memory oriented processing to hardware logic. FPGA devices are divided into CLBs configurable logic blocks consisting of a number of LUTs and flip-flops with programmable inter connections. The LUTs are used for general-purpose combinatorial and sequential logic support. Devices have relatively little RAM at most enough to buffer only a

single image frame. Sparse memory is organized into small blocks usually dual ported. Blocks are a few kB in size. SRAM based devices provide additional memory resources by converting LUTs into distributed RAM. Distributed RAM may be useful for high-performance applications that require FIFOs or small register files.

Aiming at efficient utilisation of FPGA resources a systematic approach has to be devised to implement processing algorithms optimally exploiting the internal organisation of these devices. Attaining this goal requires a close reference to HDL constructions rather than to C language based tools. Incorporated in mature design suits, provided by FPGA manufacturers, VHDL or Verilog compilers support the design process with vast component and IP libraries tailored to specifics of their devices.

Processing image contents involves multiple operations determining local characteristics of pixel patches for object extraction, constructing scene descriptions and further on for elaborating interactions between marked elements. Locality means in practice a vicinity of a radius of few pixels.

One can distinguish operations based on convolutions, morphology operations and statistics. All have a common demand that is parallel access to local pixels. This demand stays in contrary to real time operation of hardware that receives a serial stream of horizontally scanned image pixels.

Considering these constraints a systematic approach for designing a FPGA based feature extraction solution is presented. The design process is based on parameterised components which can be utilised in integrated design environments without the introduction of a high level abstraction layer. A discussion of required FPGA resources and of implementation hints elaborated in prototyping a vehicle detector are presented.

The paper is organized as follows. Section 1 introduces the framework for processing video streams, next the organisation of processing in components for convolution, morphology and interest point operations are discussed. Section 3 summarizes FPGA resources for implementing the components making reference do the development of a vehicle detector. Concluding section ends with the outline of further enhancement of the presented design approach.

2 Designing Components for Feature Extraction

Real time video stream provided by cameras using horizontal scanning is a one dimensional representation of the image content S_v . An image I with m rows and n columns:

$$I(x, y) = \{x, y : x = 1, \dots, n, y = 1, \dots, m\} \quad (1)$$

is converted into video stream:

$$S_v(i) = I((i \bmod n), (i \div n)) \quad i = 1, \dots, mn \quad (2)$$

In order to acquire parallel access to pixels of a rectangular image patch P ,

$$\begin{aligned} P(s, t) &= \{s, t : s = 1, \dots, w, t = 1, \dots, h\} \\ P_v(j) &= P((j \bmod w), (j \div w)) \quad j = 1, \dots, hw \end{aligned} \quad (3)$$

pixels have to be appropriately delayed:

$$P(s, t)_{x_0, y_0} = S_v((y_0 n + x_0) + (tn + s)) \tag{4}$$

Maximum delay is thus

$$D_{max} = (hn + w)/f \tag{5}$$

where f is the pixel clocking frequency.

This shows that crucial is the height of the image patch, h values may be in the range 3 to 7 which amounts for delays of about 0,2 to 0,5ms in the case of feeding a video stream from CCTV cameras.

One of the possible organisations of delaying pixels is shown in figure 1. Patch pixels are stored in registers at the beginning of delay lines which are one image row long. VHDL description of this construction contains an expression equivalent to a shift register. This shifting array is the basic hardware processing

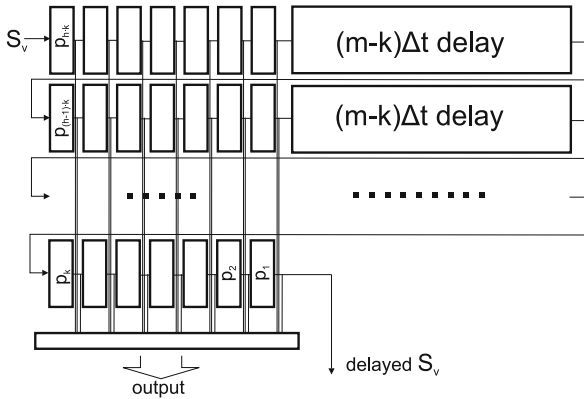


Fig. 1. Basic processing component - patch

component, which supplemented with calculation assemblies will constitute the core of processing of the device. The main parameters of this component are width and height of the patch and image row length.

2.1 Convolutions

Neighbourhood operations, filtering are carried out using convolution masks [6]. Resulting value is the weighted sum of pixel values covered by the convolution mask:

$$S_v(i) = \sum_{j=1}^{hw} w(j)P(j) \quad i = 1, \dots, nm \tag{6}$$

Two important problems arise in executing calculations: exceeding the range of values and dealing with fractional weights. It is solved by extending the size of value codes and normalising the results after calculations.

Figure 2. presents the organisation of the calculation assembly. The summing is done with registered adders in stages to avoid large propagation delays and enable pipelining. The number of stages depends on the \log_2 of the size of the patch, as summing is done in pairs. To calculate a set of convolutions on the same

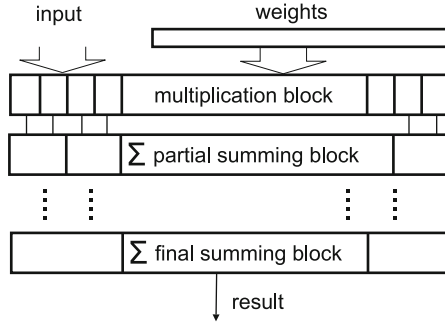


Fig. 2. Convolution calculation assembly - conv

image patch convolving assemblies may work in parallel. The total delay of the component calculating a convolution is the sum of delays of the basic component and of the convolution assembly, which is the sum of delays of the processing stages. The delay of the assembly is negligible in all as there are usually 3 to 7 stages of processing.

A useful example is the calculation of a Gaussian filter. Using a 5x5 pixel patch the patch component consists of a 4 row delay, provides a 25 pixel register result, which is multiplied by weights stored in the convolution mask register, next products are summed up in a 5 stage pipeline.

This assembly is characterised by the set of weights and their ranges which determine the required coding parameters and normalisation parameters.

2.2 Morphology Operations

Morphology operations are extensively used in processing image data, because of their high effectiveness [10]. Basic operations such as erosions and dilations condition image objects enhancing visibility and segmentation tasks. Compositions of the operations and especially HMT allow for extraction of features of image objects. Morphology based operations for gray scale pixels require the determination of extrema values in the vicinity defined by the structuring element SE or patch as defined earlier.

$$S_v(i) \ominus SE = \min_{j \in SE} (S_v(i) + SE(j)) \quad i = 1, \dots, nm \quad (7)$$

$$S_v(i) \oplus SE = \max_{j \in SE} (S_v(i) - SE(j)) \quad i = 1, \dots, nm \quad (8)$$

Determination of extrema is performed using a sorting network [12]. Odd-even merge sorting network was chosen as it requires the least number of comparison elements [13]. The network is modified to enable pipeline processing. Each comparison element is registered and data which does not enter the comparison chain is pushed through delay registers. The sorting network consists of $p(p+1)/2$ stages and $(p^2 - p + 4)2^{p-2} - 1$ comparison elements, where $p \geq \log_2(wh)$.

When a large structuring element is utilised as is the case in template matching the SE may be decomposed into smaller elements [12].

$$SE = SE1 \oplus SE2 \oplus \dots \oplus SEN \tag{9}$$

$$I \oplus SE = (((I \oplus SE1) \oplus SE2) \oplus \dots) \oplus SEN \tag{10}$$

The operation is performed by serially linking components. This results in summing the component delays which gives a total delay predominantly determined by the height of the SE.

The centre value in the output register is the median value of the patch, it can be regarded as the output of a median filter, useful for cleaning noise in images. Calculating morphological gradient is done in parallel as the sorting

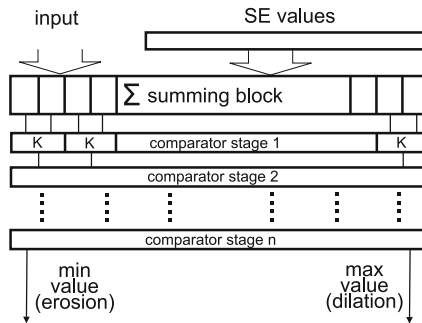


Fig. 3. Morphology operations assembly - morph

network provides both min and max at the same time. This may be regarded as representative of using morphology operations. Utilising a flat circular SE with 32 pixels the assembly consists of a summing stage and a 15 stage sorting network. The gradient is calculated with an additional subtraction stage, which subtracts the minimum from the maximum of the patch values.

2.3 Interest Points Operations

Detection of interest points is based on analysing pixel value changes or pixel value statistics inside a patch, which is usually a circular mask centred on the potential interest point. Most significant for tracking applications and object matching are corner detectors for instance SUSAN, Harris, SIFT [8], [9].

SUSAN algorithm consists of counting pixels which differ from the centre point by a defined threshold and finding their proportion to the total number in the vicinity defined by the mask. The resulting value is compared to a detection level which is set to discriminate corners, edges or other salient points of objects. Fig.4. illustrates the processing organisation. In the case of the Harris detector

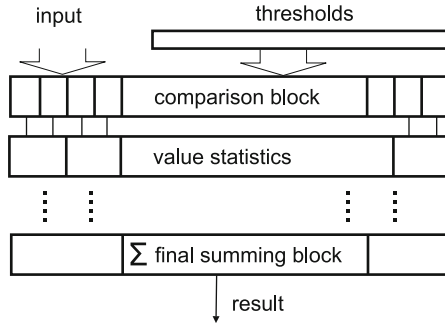


Fig. 4. Interest points calculation assembly - intp

vertical, horizontal and diagonal pixel value changes, in the patch, are calculated and a cornerness function is evaluated. The calculating assembly is of similar complexity to SUSAN.

3 FPGA Resources

Evaluation of the proposed design approach was done by examining resource utilisation and performance of the components. Examples of proposed components were implemented using VHDL based integrated design tools from leading FPGA vendors Xilinx and Altera. Solutions using high performance and low cost devices were analysed. Table 1. presents the results.

The patch component consisted of 6 delay lines and provided up to 49 patch pixels for parallel processing. This implementation shows a significant difference between the FPGAs. Altera devices implement shift registers using block memory, whereas Xilinx uses distributed RAM in LUTs. All other implementations conv: a gaussian 5x5 filter, morph: a 5x5 median, intp: SUSAN operators, excluding Virtex6, are similar in resource utilisation. Virtex6 architecture is based on 6 input LUTs which enable a much more efficient implementation of the components functionality.

Speed performance of all components, due to optimisation for pipeline processing, depends only on silicon technology of the devices and is in the range 220 to 480 MHz for the fastest units. This means that processing a single CCTV frame may be accomplished in less than one millisecond using a number of parallel or serially linked components. The processing results however are delayed

Table 1. FPGA resources utilised by components

component	FPGA family			
	high performance		low cost	
	Virtex6	Stratix III	Spartan 3E	Cyclone II
patch	379 slices 1216 LUTs	515 LE 161 ALUTs 34kb bl. mem.	1274 slices 2168 LUTs	517 LE 61 LUTs. 34kb bl. mem.
morph	1748 slices 5495 LUTs	4108 LE 2925 ALUTs	3017 slices 3855 LUTs	2493 LE 2221 LUTs
conv	53 slices 195 LUTs	608 LE 435 ALUTs	308 slices 514 LUTs	682 LE 435 LUTs
intp	231 slices 421 LUTs	581 LE 473 ALUTs	390 slices 717 LUTs	702 LE 696 LUTs

proportionally to the number of patch components used as equ. 5 shows. A prototype of a vehicle detector uses these components and successfully detects vehicles in detection fields [16]. It is based on a low cost Spartan 3E FPGA. Fig. 5 shows the main processing diagram and exemplary results of implementing a HMT based feature detector for highlighting vehicles in images. The detection result is delayed by 12 image lines which is acceptable for using this solution in a traffic control application.

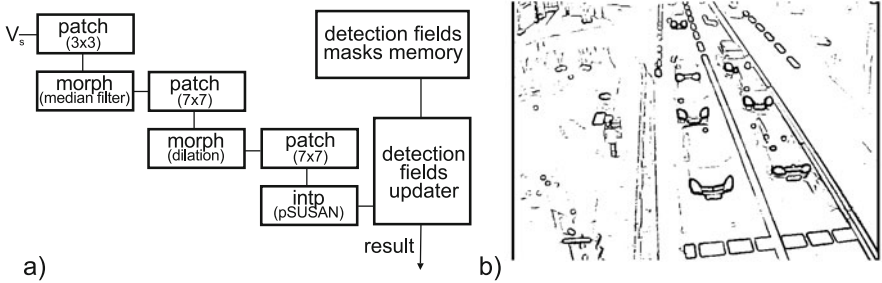


Fig. 5. a) Block diagram of a vehicle detector, b) HMT calculation result - binarized and inverted

4 Conclusions

The proposed reconfigurable hardware processing architecture based on components adapts well to available VHDL development environments. Elaborated components function as library entities enabling efficient design of processing algorithms and flexible tuning of their properties. Integration with new Mathworks Simulink tools for developing HDL code may provide efficient paths for verifying the functioning of the devised components and enhancing their properties.

References

1. Virtex-6, Spartan-6 Family Overview. Xilinx Inc. San Jose, CA USA (2010)
2. Altera Product Catalog. Altera Co. San Jose, CA, USA (2009)
3. Benitez, D.: Performance of Reconfigurable Architectures for Image-Processing Applications. *Journal of Systems Architecture* 49, 193–210 (2003)
4. Porter, R., Frigo, J., Conti, A., Harvey, N., Kenyon, G., Gokhale, M.: A Reconfigurable Computing Framework for Multiscale Cellular Image Processing. *Microprocessors and Microsystems* 31, 546–563 (2007)
5. Kessal, L., Abel, N., Karabernou, S.M., Demigny, D.: Reconfigurable Computing: Design Methodology and Hardware Tasks Scheduling for Real-Time Image Processing. *J. Real-Time Image Proc.* 3, 131–147 (2008)
6. Porter, R., Frigo, J., Gokhale, M., Wolinski, C., Charot, F., Wagner, C.: A Run-Time Reconfigurable Parametric Architecture for Local Neighbourhood Image Processing. In: *Proceedings of the 9th Euromicro Conference on Digital Systems Design*, pp. 362–370. IEEE Computer Society, Los Alamitos (2006)
7. Torres-Huitzil, C., Arias-Estrada, M.: An FPGA Architecture for High Speed Edge and Corner Detection. In: *Proceedings of the 5th IEEE International Workshop on Computer Architectures for Machine Perception*, pp. 112–116. IEEE Computer Society, Los Alamitos (2000)
8. Claus, C., Huitl, R., Rausch, J., Stechele, W.: Optimizing The SUSAN Corner Detection Algorithm for a High Speed FPGA Implementation. In: *International Conference on Field Programmable Logic and Applications, FPL 2009*, pp. 138–145. IEEE Computer Society, Los Alamitos (2009)
9. Bonato, V., Marques, E., Constantinides, G.A.: A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 1703–1712 (2008)
10. Baumann, D., Tinembart, J.: Designing Mathematical Morphology Algorithms on FPGA: An Application to Image Processing. In: *Gagalowicz, A., Philips, W. (eds.) CAIP 2005. LNCS, vol. 3691*, pp. 562–569. Springer, Heidelberg (2005)
11. Yamaoka, K., Morimoto, T., Adachi, H., Koide, T., Mattausch, H.J.: Image Segmentation and Pattern Matching Based FPGA/ASIC Implementation Architecture of Real-Time Object Tracking. In: *Asia and South Pacific Conference on Design Automation*, pp. 176–181. IEEE Computer Society, Los Alamitos (2006)
12. Batcher, K.E.: Sorting Networks and Their Applications. In: *Spring Joint Computer Conf., AFIPS Proc.*, vol. 32, pp. 307–314 (1968)
13. Liszka, K.J., Batcher, K.E.: A Generalized Bitonic Sorting Network. In: *Proceedings of the International Conference on Parallel Processing*, pp. 105–108 (1993)
14. Muthukumar, V., Rao, D.V.: Image Processing Algorithms on Reconfigurable Architecture Using HandelC. In: *Proceedings of the 7th Euromicro Conference on Digital Systems Design*, pp. 362–370. IEEE Computer Society, Los Alamitos (2004)
15. Beun, R., Karkowski, I., Ditzel, M.: C++ Based Design Flow for Reconfigurable Image Processing Systems. In: *International Conference on Field Programmable Logic and Applications*, pp. 571–575. IEEE Computer Society, Los Alamitos (2007)
16. Project Report: Modules of Video Traffic Incidents Detectors ZIR-WD for Road Traffic Control and Surveillance. WKP-1/1.4.1/1/2005/14/14/231/2005, vol.1-6, Katowice, Poland (2007)

Automated Counting and Characterization of Dirt Particles in Pulp

Maryam Panjeh Fouladgaran¹, Aki Mankki², Lasse Lensu¹, Jari Käyhkö²,
and Heikki Kälviäinen¹

¹ Machine Vision and Pattern Recognition Laboratory
Department of Information Technology
Lappeenranta University of Technology, Finland
53850 Lappeenranta, Finland
`lasse.lensu@lut.fi`

² FiberLaboratory, Department of Chemical Technology
Lappeenranta University of Technology, Finland
53850 Lappeenranta, Finland

Abstract. Dirt count and dirt particle characterization have an important role in the quality control of the pulp and paper production. The precision of the existing image analysis systems is mostly limited by methods for only extracting the dirt particles from the images of pulp samples with non-uniform backgrounds. The goal of this study was to develop a more advanced automated method for the dirt counting and dirt particle classification. For the segmentation of dirt particles, the use of the developed Niblack thresholding method and the Kittler thresholding method was proposed. The methods and different image features for classification were experimentally studied by using a set of pulp sheets. Expert ground truth concerning the dirt count and dirt particle classes was collected to evaluate the performance of the methods. The evaluation results showed the potential of the selected methods for the purpose.

1 Introduction

It is important to evaluate the quality of pulp and paper to control the production and to give information about a product for marketing. It should be produced wanted quality, not too high or low quality, in order to minimize the use of raw material, energy costs, etc., and to maximize profits. Depending on the end use of the product, there exists different quality grades of pulp and paper. Common to all grades, the product must contain properties according to its specifications. Therefore, the testing of pulp quality is an important method to characterize the properties of a paper product, and it is especially effective in case of paper grades with no or low amount of coating. However, there is no general definition of the pulp quality, and the concept of the quality always depends on the end purpose of the product.

Modern imaging equipment with advanced image processing and analysis methods enables many quality control applications. Properly designed machine vision systems can be used to characterize also pulp and paper, and such systems can handle the different requirements for the end products.

One of the most important quality factors in pulp production is the dirt count. Dirt counting describes the proportion of any foreign material in the pulp. Depending on the raw material and production process, the final product contains different levels of dirt count that highly affects the quality of the product and its possible use in an application [1]. Most of the current image analysis systems for automated dirt counting use only a single threshold value for extracting dirt particles from digital images. This is one of the most important drawbacks of these systems. Since the background color and texture of the sample sheets from different stages of manufacturing process is not uniform, the precision of the systems becomes limited. Therefore, the goal of this study was to define an automatic thresholding method to extract the dirt particles of different kind of sample sheets which is as invariant as possible to the varying background.

The dirt count is not the only quality factor of interest. Dirt particles can be classified into different categories such as bark, shives, knots, sand, etc. Since the process control would gain the most from information concerning the source of a quality problem, extracting relevant features of dirt particles and classifying the particles into the before-mentioned classes would enable fast recovery from certain issues. Most of the current image analysis systems have focused on the dirt count to give information about the quality of the pulp and paper. The dirt characterization with automatic segmentation approach which does not suffer from the variation of the background is proposed in this study, using the developed Niblack method [5]. Based on the observations, the performance of the proposed automatic thresholding method is compared to the generally used Kittler thresholding [4], and also to visual inspection which is considered as an accurate method for dirt counting. Therefore, the aim is to determine the features, which could separate different type of dirt particles properly and classify them.

2 Materials and Methods

2.1 Pulp Sheets

The pulp samples used in this study were provided by a company producing equipment for the pulp industry. The samples were laboratory sheets of chemical pulp from different mills around the world. The various pulp sheets, which include different kinds of dirt particles such as bark, shives, knots, and sand, give an overview of the varying pulp used in producing paper and board. Front-illuminated reflection images of the samples were taken by using a high-quality scanner with all automatic settings disabled. The images were captured with the spatial resolutions of 1250 and 2500 dpi where a separate ICC profile was

used for the both modes, and each pixel was represented by 48-bit RGB color information.

2.2 Segmentation

Segmentation of regions of interest in an image is one of the most important methods in image analysis. Despite the considerable research effort, there exists no universal method for the general-purpose segmentation. One of the most applied category of these methods relies on global or local thresholding. Several methods exist, and some those methods suit well for segmenting small low-contrast defects in varying backgrounds [7].

Developed Niblack Thresholding. The method by Niblack [5] (Local Niblack) adapts the threshold according to the local mean $m(i, j)$ and standard deviation $\sigma(i, j)$ as follows:

$$T(i, j) = m(i, j) + k \cdot \sigma(i, j) \quad (1)$$

The Niblack thresholding is a local thresholding method. In this study, the defined equation is utilized to globally determine the threshold level. Based on the above equation, k is increased step by step until the number of segmented parts becomes constant which means that the background is segmented. Thus, in this paper the method is called the developed Niblack method.

Kittler Thresholding. The method was selected based on an earlier study focusing on the detection of small details on paper surface [2]. In the method, the gray-level images are segmented into background and foreground, or alternately are modeled as a mixture of two Gaussians [4]. The method assumes that the image can be characterized by a mixture distribution of the foreground and background pixels, that is,

$$p(g) = P(T) \cdot p_f(g) + [1 - P(T)] \cdot p_b(g) \quad (2)$$

where $p_f(g)$ and $p_b(g)$ are the probability mass functions (PMF) of foreground and background and $P(T) = \sum_{i=0}^T p(i)$ is a cumulative probability function. The method avoids the assumption concerning equal variances and, in essence, addresses the minimum error Gaussian density-fitting problem.

2.3 Feature Extraction

Geometric Features. After the segmentation, geometric features are computed based on the boundary shape of segments (see, for example, [6]). Next these features are introduced.

$$FormFactor = \frac{4\pi \cdot Area}{Perimeter^2} \quad (3)$$

where *Area* is the number of pixels inside the boundary and *Perimeter* is the path surrounding an area. In an image processing application, it can be estimated as the length of the boundary around the object by counting the pixels that touch the background.

$$Elongation = \frac{FiberLength}{FiberWidth} \quad (4)$$

where the *FiberLength*, *FiberWidth* are defined as

$$FiberLength = \frac{Perimeter - \sqrt{Perimeter^2 - 16 \cdot Area}}{4} \quad (5)$$

$$FiberWidth = \frac{Area}{FiberLength} \quad (6)$$

The other geometric features used to characterize the dirt particle boundary are as follows:

$$Curl = \frac{MajorAxisLength}{FiberLength} \quad (7)$$

where *MajorAxisLength* is the length of major axis of the ellipse with the same normalized second central moments as the segmented region.

$$Extent = \frac{Area}{BoundaryBoxArea} \quad (8)$$

where *BoundaryBox* is the smallest rectangle containing the region.

$$Roundness = \frac{4 \cdot Area}{\pi \cdot MajorAxis^2} \quad (9)$$

$$AspectRatio = \frac{MajorAxisLength}{MinorAxisLength} \quad (10)$$

where *MinorAxis* is the minor axis of the ellipse with the same normalized second central moments as the segmented region.

$$Solidity = \frac{Area}{ConvexHullArea} \quad (11)$$

where *ConvexHull* is the smallest convex polygon that can contain the region. In addition, the fractal dimension of the boundary is calculated as follows:

$$D = \frac{\log(N(l))}{\log(l)} \quad (12)$$

where $N = l^D$ is the number of self-similar objects to cover the original object. In this study, the fractal dimension of boundary of the dirt particles is defined by the number of pixels on the boundary N and its length l .

Texture Features. Texture characterizes an image region as the variation in brightness [3]. To consider the applicability of texture as a feature, the histograms of each segmented part can be utilized. Next, the histogram differences between the segmented parts belonging to the same class of a dirt particle are computed, and the mean values of intra-class differences are considered as texture. The histograms are defined in 16 intervals (from the range of 8 bits) and they are normalized to the identical mean.

Statistical Texture Features. Another approach to describe texture is to use statistical moments of intensity histogram of the region [3]. The n th moment of z is

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (13)$$

where z is a random variable representing the intensities $p(z_i)$, $i = 0, \dots, L - 1$ is the corresponding histogram, and m is the mean value of z , i.e., $m = \sum_{i=0}^{L-1} z_i p(z_i)$. R as a descriptor of the relative *smoothness* is defined as

$$R = 1 - \frac{1}{1 + \mu_2^2(z)} \quad (14)$$

where $\mu_2(z)$ is the second moment which is a measure of the intensity contrast. In addition, U as a measure of *uniformity* based on the intensity histogram of the region is defined as

$$U = \sum_{i=0}^{L-1} p^2(z) \quad (15)$$

In this study, *smoothness* and *uniformity* are used to represent the texture of the dirt particles.

3 Experiments

3.1 Evaluating Segmentation

To evaluate the accuracy of the developed Niblack method, the dirt count was compared to the Kittler thresholding method. The dirt count gives the number of dirt particles in four different size groups according to the TAPPI standard. Fig. 1 shows the mean square error (MSE) of the methods as compared to a commercial system called the Digital Optical Measurement and Analysis System (DOMAS) and the visual inspection result, whereas Table 1 presents the exact counts for the eight samples from the different stages of the pulping process. It should be noted that the relatively low number of samples and dirt particles in specific samples did not allow for a full statistical evaluation of the methods. Based on the result, however, the Kittler thresholding method can be considered to perform better than the developed Niblack thresholding method.

For the first sample with uniform background, the both thresholding methods produced practically identical results. The second sample with non-uniform background shows small differences because of the dark particles. However, the last example illustrates that the developed Niblack thresholding method produced

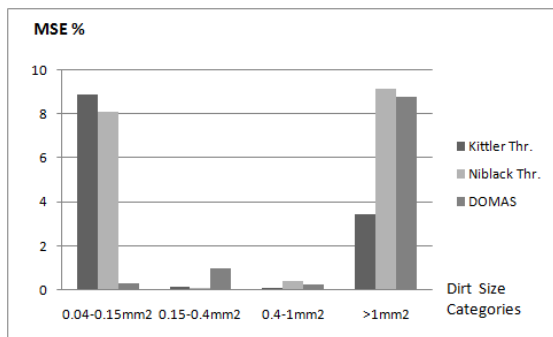


Fig. 1. MSE of dirt count in four dirt particle size categories by the Kittler thresholding method, the developed Niblack thresholding method and DOMAS as compared to visual inspection

Table 1. The number of categorized dirt particles based on their size for the eight pulp samples. The dirt particles are segmented by the developed Niblack thresholding method (DN), the Kittler thresholding method (KT) and DOMAS (DO) as compared to visual inspection (VS).

#	Number of Dirt Particles/Size Class															
	0.04mm ² - 0.15mm ²				0.15mm ² - 0.4mm ²				0.4mm ² - 1mm ²				> 1mm ²			
	VS	DO	DN	KT	VS	DO	DN	KT	VS	DO	DN	KT	VS	DO	DN	KT
1.	> 200	213	184	145	33	49	54	41	10	19	9	9	6	6	7	6
2.	> 200	309	207	211	48	81	82	76	23	29	18	14	0	22	3	3
3.	> 200	65	218	192	48	16	67	57	12	9	13	11	2	3	3	2
4.	≈ 200	232	237	138	46	56	99	41	29	27	48	23	5	26	33	18
5.	> 200	111	100	33	10	18	13	5	0	1	3	0	0	0	0	0
6.	10	4	10	1	0	1	2	2	0	0	0	0	0	0	0	0
7.	5	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
8.	3	2	16	3	0	0	1	0	0	0	1	0	0	0	0	0

more segmented pixels. In this sample, the dirt particle color is close to the background which caused the Kittler thresholding to segment it as background. Therefore, the Kittler thresholding method produced better segmentation results in this case.

3.2 Feature Evaluation

The proposed geometric and texture features were considered for the classification of the dirt particles in the sample sheets.

Fig. 2 shows the comparison of the used features for the dirt particle classes barks and fiber bundles of the five pulp sample sheets from different stages of pulp production with varying background. The challenge of this case is realistic for the

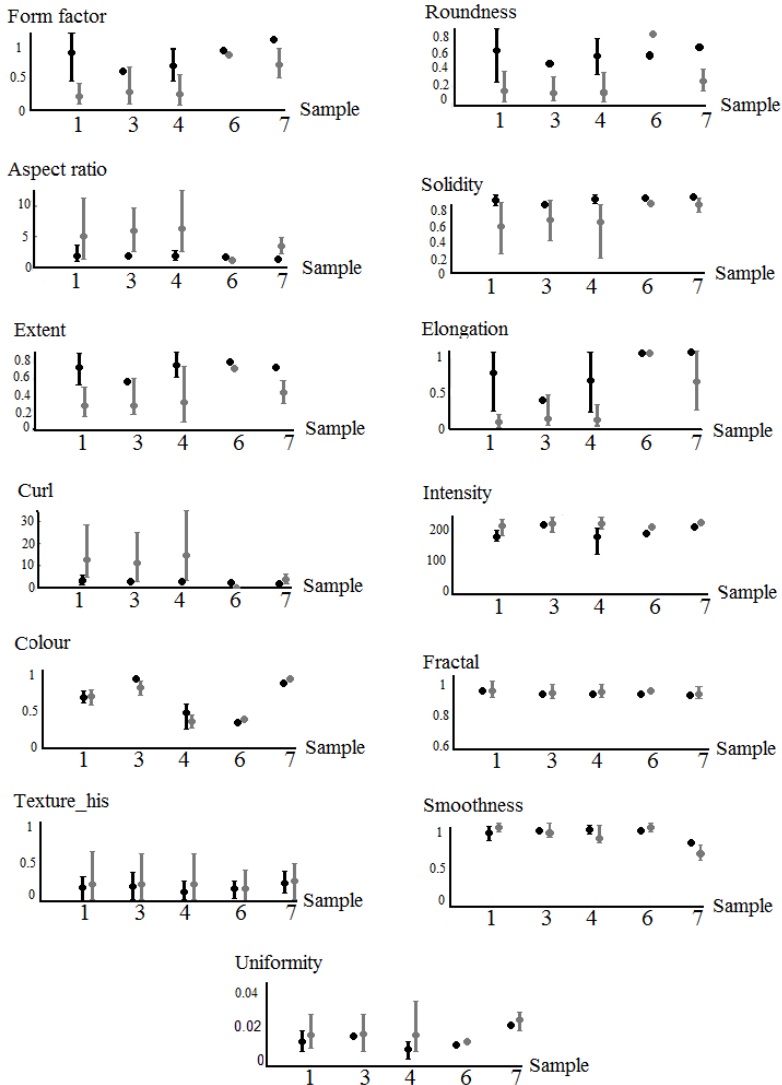


Fig. 2. Variation of the value of all the features for fiber bundles (gray) and bark (black) in different pulp samples. The circle points show the mean value and the limits are the minimum and maximum; segmentation was performed by the Kittler thresholding method.

application. Most intra-class feature values do not keep constant in different pulp process stages, but some features, such as curl, can separate the classes well.

3.3 Dirt Particle Classification

As the final step, the features were utilized to classify the dirt particles. Table 2 presents the accuracy of the classification with a multilayer perceptron (MLP) neural network which consisted of three neural layers and non-linear response functions. In the tests, MLP showed the best performance when compared to k-nearest neighbor (k-NN) and linear discriminant analysis (LDA). Two sets of features were used in the classification of dirt particles in samples with uniform and non-uniform background color. Table 2 shows the classification accuracy of two sets of pulp sample sheets with a uniform background (49 plastic and 25 fiber bundle dirt particles), and a non-uniform background (56 fiber bundle and 11 bark dirt particles).

Table 2. The classification accuracy of two sets of pulp sample sheets with uniform and non-uniform background

Samples	Uniform Background		Non-uniform Background	
	Plastics	Fiber bundles	Bark	Fiber bundles
Accuracy	100	66.67	85.71	95.83

4 Discussion

The goal of this study was to develop an automated method for dirt counting and particle classification. The motivation for this development arose from the problems of the current systems relying on a single threshold level or even manual selection of the threshold used in segmentation.

The two thresholding methods, Niblack and Kittler, were studied to segment the dirt particles from the image background. Despite the small differences, the Kittler thresholding method was found to produce smaller error when compared to visual inspection. In addition, Kittler thresholding produced better results especially in the case where the difference between the background and foreground is recognizable. However, the Niblack thresholding method can segment more particles from the samples collected from the last stages of the pulping process, which is a clear benefit for application development.

The selected geometric and texture features were compared to each other with samples from the different stages of the the pulp process. Most of the features contained a different inter-class value implying that they can be used to distinguish two classes of dirt. Curl was the best feature to separate bark and fiber bundles, which can be difficult to differentiate. As the last step, the features were utilized to classify the dirt particles. It was shown that all the plastic particles are correctly classified.

5 Conclusion

In this study, the dirt counting and the characterization of dirt particles in pulp sheets was considered. Two segmentation methods, which can adapt to different background color and uniformity, were proposed to use to overcome the problem of current systems for dirt counting.

The future work includes the evaluation of the methods with a fully representative set of pulp samples, and the use of back-illuminated images for the feature extraction.

Acknowledgements. The work was carried out in the QVision consortium funded by Forestcluster, Ltd., Finland.

References

1. Campoy, P., Canaval, J., Pena, D.: An on-line visual inspection system for the pulp industry. *Computer in Industry* 56, 935–942 (2005)
2. Drobchenko, A., Vartiainen, J., Kämäräinen, J.K., Lensu, L., Kälviäinen, H.: Thresholding based detection of fine and sparse details. In: *Proceedings of the Ninth IAPR Conference on Machine Vision Applications (MVA 2005)*, Tsukuba Science City, Japan, May 16–18, pp. 257–260 (2005)
3. Gonzales, R.C., Woods, R.E.: *Digital Image Processing*, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)
4. Kittler, J., Illingworth, J.: On threshold selection using clustering criteria. *IEEE Trans. Syst. Man Cybern.* 15, 652–655 (1985)
5. Niblack, W.: *An Introduction to Image processing*. Prentice Hall, Englewood Cliffs (1986)
6. Russ, J.C.: *The Image Processing Handbook*, 4th edn. CRC Press, Boca Raton (2002)
7. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* 13(1), 146–165 (2004)

Keypoint-Based Detection of Near-Duplicate Image Fragments Using Image Geometry and Topology

Mariusz Paradowski^{1,3} and Andrzej Śluzek^{2,3}

¹ Institute of Informatics, Wrocław University of Technology, Poland
mariusz.paradowski@pwr.wroc.pl

² Faculty of Physics, Astronomy and Informatics
Nicolaus Copernicus University, Toruń, Poland
assluzek@fizyka.umk.pl

³ School of Computer Engineering, Nanyang Technological University, Singapore
assluzek@ntu.edu.sg

Abstract. One of the advanced techniques in visual information retrieval is detection of *near-duplicate fragments*, where the objective is to identify images containing almost exact copies of unspecified fragments of a query image. Such near-duplicates would typically indicate the presence of the same object in images. Thus, the assumed differences between near-duplicate fragments should result either from image-capturing settings (illumination, viewpoint, camera parameters) or from the object's deformation (e.g. location changes, elasticity of the object, etc.). The proposed method of near-duplicate fragment detection exploits statistical properties of keypoint similarities between compared images. Two cases are discussed. First, we assume that near-duplicates are (approximately) related by affine transformations, i.e. the underlying objects are locally planar. Secondly, we allow more random distortions so that a wider range of objects (including deformable ones) can be considered. Thus, we exploit either the *image geometry* or *image topology*. Performances of both approaches are presented and compared.

1 Introduction

Content-based visual information retrieval (CBVIR) is often based on the notion of *near-duplicate images* which indicate images of basically the same scene captured under varying settings (e.g. illumination, camera position, minor variations in the scene contents, etc.). Although this is obviously not the most universal model of CBVIR, it is gaining popularity (e.g. [1]) in several important applications, including analysis of video-clips (e.g. [2]).

In sub-image retrieval, we focus on a more general task of identifying images which contain *near-duplicate fragments*. However, the most general formulation of sub-image retrieval, i.e. detection of unspecified similar fragments in random images of unpredictable contents, is conceptually very complex and until recently

very few reports (e.g. [3], [4]) addressed this issue. Our paper presents methods which can be considered a continuation and improvement of these attempts. Nevertheless, to the best of our knowledge, no reported technique provides functionalities available in our algorithms.

2 Background

The expected functionality of the presented method is illustrated in Fig. 1. Given two images of unknown and unpredictable contents, near-duplicate fragments should be detected and localized.



Fig. 1. Examples of manually extracted near-duplicate regions for planar objects (a) and for objects with irregular/deformable surfaces (b)

Two scenarios are considered. First, we assume the objects of interest are (approximately) piecewise planar solids. Then, appearances of the same face of a solid in different images are related by affine transformations which sufficiently accurately approximate perspective projections in a wide range of viewing angles and focal lengths. The objective is, therefore, to identify in matched images affine-related visually similar regions which presumably represent the same (subject to partial occlusions) planar objects present in both images (see Fig. 1a).

In the second case, we consider objects of any shapes (including objects with deformable surfaces). Differences between visual appearances of such objects cannot be modeled by any regular mapping. Instead, we attempt to identify in matched images visually similar regions that may have arbitrary shapes (see Fig. 1b) but their topologies are consistent.

2.1 Keypoint Detectors and Descriptors

Both approaches are based on the same low-level mechanism, i.e. detection and description of *keypoints* (a survey of the most popular detectors is provided in [5]). Keypoints are features representing location with prominent characteristics of image intensities/colours. Even though keypoints do not reflect global image semantics or contents, and their performances are not always perfect (in terms of repeatability under viewpoint and illumination changes) they provide large numbers of fairly stable local data that can be subsequently used to detect geometrically or topologically similar image fragments.

Keypoints are characterized by various descriptors (see [5]) so that local image similarities can be estimated by determining distances between descriptors of the matched keypoints. The proposed algorithm can work practically with any fairly reliable keypoint detector, but (for the reasons presented in Section 3) we prefer to use Harris-Affine or MSER detectors [5] combined with either SIFT [6] or SURF [7] descriptors.

Keypoints from the matched images can be compared using various schemes (see [2]) including *many-to-many* (M2M), *one-to-many* (O2M), and *one-to-one* (O2O). O2O assumes that two keypoints are matched if their descriptors are the mutual nearest neighbors. Usually, O2O matching provides the highest precision (i.e. the highest ratio of correct matches over the total number of matches) and, therefore, the O2O scheme is applied in our algorithms. However (subject to a higher computational complexity) other matching schemes can be used as well. In particular, M2M is recommended when multiple copies of the same objects are expected in the processed images.

Large numbers of keypoints (usually over 1000 in typical images) produce correspondingly large sets of keypoint pairs matched between two images. Even if those matches correctly identify *local* visual similarities, most of them are incorrect in the context of the whole image. Therefore, the search for near-duplicate image fragments should be based on statistical properties of keypoint similarities. However, the RANSAC-based methods are not applicable because they can remove outliers only if a significant percentage of all matches are inliers. In matching random images, the population of correctly matched keypoints can be statistically negligible (or even non-existing if images do not contain near-duplicate fragments) compared to all matches.

Therefore, in the geometric approach, we propose to use histograms of affine transformation parameters, while the topological method is based on the local distributions of oriented vectors.

3 Affine-Related Near-Duplicate Regions

An affine transformation is represented by six parameters, e.g.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \text{where } \mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

i.e. $\mathbf{K} = \langle a, b, d, e \rangle$ and $\langle c, f \rangle$ are the linear and the translation parts of the transformation, respectively.

Keypoints of planar near-duplicates are mapped by (approximately) the same affine transformations. Therefore, we identify groups of keypoints related by affine transformations (i.e. the alleged near-duplicate regions) by detecting local maxima (spikes) in the distribution of all affine transformations that can be built between sets of matched keypoints from two images. Three factors, however, should be considered to make this approach practical:

- Affine transformations have six parameters and $6D$ histograms cannot be easily built and processed.
- The algebraic form of affine transformations is sensitive to parameter errors, in particular in modeling deformations/motions of the underlying objects.
- Affine mappings are defined by the correspondence of three pairs of points. If n pairs of keypoints matched between two images exist, the total number of transformations can reach n^3 , i.e. the computational complexity is $O(n^3)$.

The first problem is handled by hashing. Instead of using $6D$ arrays of affine parameters, histograms are built using hash-tables. Each histogram bin is represented as a single entry in the hash-table containing the affine parameters and the coordinates (in both images) of keypoint pairs contributing to the transformation. Then, hundreds of thousands of affine transformations can be effectively handled by a PC-run program.

3.1 Decomposition of Affine Transformations

In order to provide a more meaningful description of affine transformations, we decompose them using two different schemes. The first one is *SVD-based decomposition* (see [9]) where the linear part of affine transformations is decomposed into two planar rotations and two scaling coefficients (the translational part is unchanged), i.e. $\mathbf{K} = \mathbf{Rot}(\gamma) \cdot \mathbf{Rot}(-\theta) \cdot \mathbf{N} \cdot \mathbf{S} \cdot \mathbf{Rot}(\theta)$, where γ and θ are $2D$ rotation angles, \mathbf{S} is a diagonal scaling matrix, and \mathbf{N} is either an identity matrix or a mirror reflection matrix.

Alternatively, we decompose the *whole* affine transformation into three rotations in a $3D$ space, followed by a $3D$ translation, and a $2D$ orthogonal projection, i.e. $\mathbf{A} = \mathbf{Rot}(\phi_Z) \cdot \mathbf{Rot}(\phi_X) \cdot \mathbf{Rot}(\phi_F) \cdot \mathbf{Trans}(P) \cdot \mathbf{Proj}(Z)$, where ϕ_Z and ϕ_F are rotations about Z -axis, ϕ_X is a rotation about X -axis, P is the vector of a $3D$ translation (that, in effect, combines a planar translation and scaling) and $\mathbf{Proj}(Z)$ is a projection along Z -axis.

Both decomposition schemes have certain advantages and they can be interchangeably used to build histograms of affine transformations.

3.2 Histogram Building

Originally, histograms of affine transformations are built using triangles of matching keypoints (i.e. the complexity is $O(n^3)$). To reduce the complexity, we restrict the number of processed triangles by taking only m neighbors of a given keypoints to form triangles. Additionally, triangles that are too small, too large or too "flat" are also rejected. In this way, a uniform distribution of triangles over the whole image is maintained.

The obtained histograms usually contain highly distinctive peaks for images containing affine-related regions (if enough keypoints can be detected within the regions). Fig. 2 shows an exemplary pair of images in which two nearly-planar similar fragments exist. The $6D$ histogram of affine transformations built from matching keypoints has two prominent peaks corresponding to both fragments.

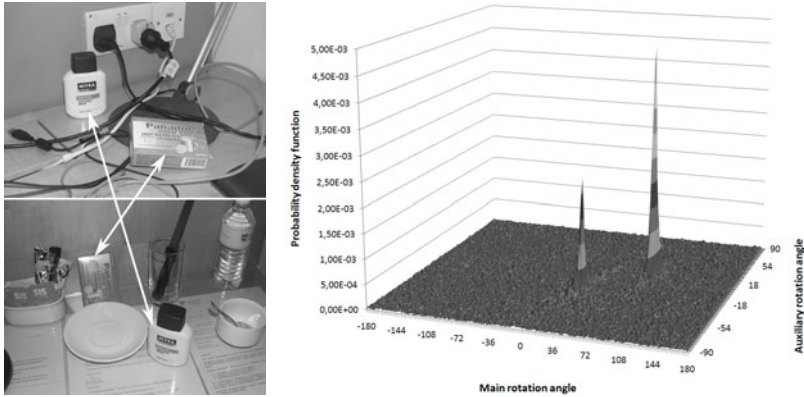


Fig. 2. A pair of images with two near-duplicate planar regions. Histogram of SVD-decomposed affine transformations (rotation-angles subspace only).

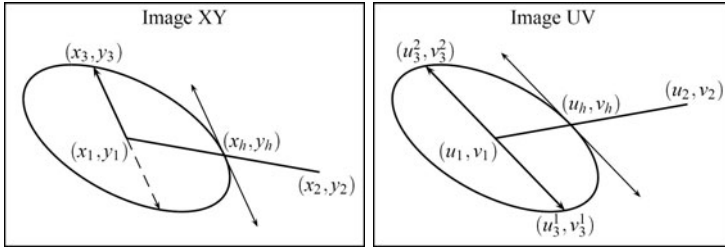


Fig. 3. The third pair of points to build an affine transformation is found from intersections of ellipses and shifted tangent lines

The histogram projection onto a $2D$ subspace (two rotation angles) is also shown in the figure to illustrate the prominence of peaks.

The alternative method to build histograms has $O(n^2)$ complexity. Although the correspondences between three points are still needed, we use only *two* matching keypoints. If the keypoints are actually *key regions* of elliptical shapes (e.g. provided by Harris-Affine, MSER or Fast-MSER [10] keypoint detectors) the third correspondence can be derived from a pair of ellipses (key regions) as explained in Fig. 3. More details on building affine transformation histograms, and on the identification of affine-related near-duplicates from local maxima of the histograms, are provided in [8] and [10].

4 Topological Near-Duplicate Regions

Affine relations cannot be used to detect near-duplicates of objects with non-planar surfaces (although, as shown in Fig. 5a, a certain amount of non-linearity can be tolerated) or objects which are naturally deformable (e.g. Fig. 1b). For the

retrieval of such near-duplicate regions we propose an algorithm based on point-based topological *pseudo-invariants*. They are referred to as "pseudo" because they are not invariant to all mappings between topologically equivalent shapes; instead they are invariant only under typical distortions of objects encountered in machine vision. Currently, we use only one such invariant, namely the order of rotations in a set of vectors sharing the same origin.

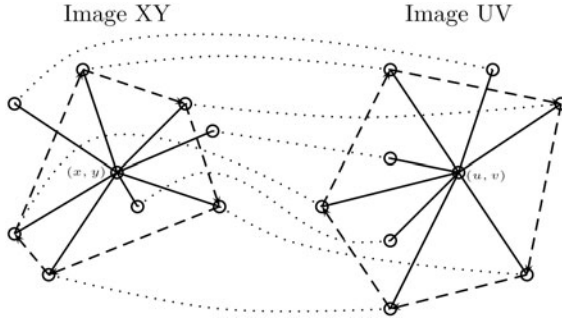


Fig. 4. The longest sequence (5 out of 8) of preserved rotations for vectors originating from a pair of matched keypoints (x, y) and (u, v) . The matches between the heads of vectors are shown as well.

If the order of rotations in both images is preserved for the majority of vectors (see the example in Fig. 4) in two sets originating from a pair of matched keypoints (x, y) and (u, v) and terminating in other matched keypoints, (x, y) and (u, v) are identified as keypoints belonging to the near-duplicate regions. Grouping keypoints into objects is performed according to the topology graph adjacency matrix. If any two vertexes are connected by a directed edge, they form a single object. The theoretical complexity of this algorithm is $O(n^4)$ but we have constrained the number of vectors similarly to the triangle method.

5 Performance Analysis

Both algorithms have been tested on a dataset of 100 highly diversified indoor and outdoor images (see <http://www.ii.pwr.wroc.pl/~visible>) i.e. 10,000 image-to-image matches can be performed. The ground truth has been established semi-automatically. First, objects shared by at least two images are manually outlined in the whole dataset. Secondly, for a given pair of images (and a selected keypoint detector) the matched keypoints within outlined regions are automatically clustered, and their convex hulls are formed (i.e. the detector's abilities to extract visual data from the images are taken into account). Such convex hulls are considered the ground truth for each image-to-image matching. Since the detected near-duplicates are also represented by convex hulls of keypoint groups, performances can be evaluated numerically using the popular measures of *precision* and *recall*. Two criteria are used, namely the *object-based* estimate

(overlaps between the ground-truth areas and the returned areas) and the *area-based* estimate (the shape conformity between the ground-truth areas and the returned areas).

The affinity-based method is able to extract both planar and nearly planar objects. It is highly resistant to false positives. However, it fails to detect objects in some cases. Exemplary results of image matching for the affine method are given in Fig. 5a. The topological method extracts a higher percentage of objects, but fragments of the surrounding background are often merged with the objects. Thus, the ratio of false positives (in terms of the *area* criterion) increases, but the number of false negatives is lower. Exemplary matching results are given in Fig. 5b. Some highly non-planar near-duplicates have not been correctly identified by the affine-based matching. Details of the evaluation are provided in [8] and [10], but the summarized results are provided in Table 1.



(a) Some non-planar objects are wrongly detected.



(b) Non-planar objects are better captured, but some background area are included.

Fig. 5. Exemplary results of the affinity (a) and topology (b) based near-duplicate detection

Table 1. Average *recall* and *precision* for the test dataset

<i>Method</i>	Affine-Triangle					Affine-Ellipse			Topology
	HarAff	HarAff	SURF	MSER	fMSER	fMSER	MSER	HarAff	HarAff
<i>Descriptor</i>	SIFT	GLOH	SURF	SIFT	SIFT	SIFT	SIFT	SIFT	SIFT
Prec. (area)	0.96	0.96	0.90	0.95	0.96	0.91	0.92	0.71	0.64
Recall (area)	0.64	0.50	0.49	0.53	0.46	0.48	0.50	0.56	0.79
Prec. (obj.)	0.97	0.97	0.98	0.94	0.94	0.93	0.93	0.81	0.98
Recall (obj.)	0.81	0.71	0.61	0.68	0.63	0.65	0.68	0.76	0.92

The longest processing time (i.e. approx. 4sec per image pair) obviously exists for the topological method. The triangle-based approach is slightly faster (approx. 1-2sec) while the ellipse-based algorithm with Fast-MSER detector can almost perform in near real time (100-200msec). A modern PC with Intel Core 2 DUO 2.66 GHz with moderately optimized Java codes have been used.

6 Summary

In the paper we have discussed two approaches to image fragment matching, both being the result of our research. The geometrical approach is mathematically strict and captures the underlying 3D motion of objects and camera. The proposed topological approach (the main novelty of the paper) employs local spatial constraints between keypoints and manages to capture deformed and non-planar objects. Detection mistakes are rare, both methods have very high precision and recall, calculated in terms of detected objects.

Our further research will be focused on application of both presented methods to various problems. One of such interesting applications is automatic formation of visual classes from a completely unknown and unpredictable environment.

Acknowledgments. The research presented in this paper is a part of a joint Poland–Singapore research program. The financial support of A*STAR Science & Engineering Research Council (grant 072 134 0052) and the Ministry of Science and Higher Education Republic of Poland (project 65/N-SINGAPORE/2007/0) is gratefully acknowledged.

References

1. Cheng, X., Hu, Y., Chia, L.-T.: Image near-duplicate retrieval using local dependencies in spatial-scale space. In: Proc. 16th ACM ICM, pp. 627–630 (2008)
2. Zhao, W.-L., Ngo, C.-W., Tan, H.-K., Wu, X.: Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Transactions on Multimedia* 9(5), 1037–1048 (2007)
3. Heritier, M., Foucher, S., Gagnon, L.: Key-places detection and clustering in movies using latent aspects. In: Proc. 14th IEEE Int. Conf. IP, vol. 2, pp. II.225–II.228 (2007)
4. Zhao, W.-L., Ngo, C.-W.: Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. *IEEE Trans. on IP* 18(2), 412–423 (2009)
5. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. *Foundations & Trends in Computer Graphics & Vision* 3(3), 177–280 (2007)
6. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vision* 60(2), 91–110 (2004)
7. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). *Comp. Vision & Image Understanding* 110(3), 346–359 (2008)
8. Paradowski, M., Śluzek, A.: Matching planar fragments of images using histograms of decomposed affine transforms (unpublished 2009)
9. Xiao, J., Shah, M.: Two-frame wide baseline matching. In: Proc. 9th IEEE Int. Conf. on Computer Vision, pp. 603–609 (2003)
10. Paradowski, M., Śluzek, A.: Detection of image fragments related by affine transforms: matching triangles and ellipses. In: Proc. of ICISA 2010, vol. 1, pp. 189–196 (2010)

Three Cameras Method of Light Sources Extraction in Augmented Reality

Przemysław Wiktor Pardel¹ and Konrad Wojciechowski²

¹ Chair of Computer Science, University of Rzeszów, Poland
przemyslaw.pardel@univ.rzeszow.pl
<http://www.augmentedreality.pl/pardel>

² Institute of Information Science, Silesian University of Technology, Poland
konrad.wojciechowski@polsl.pl

Abstract. Transmission information of light sources is very important part of Augmented Reality (AR) system. Most essential aspects to make virtual objects look like real ones is lighting. To calculate adequate shadows it is necessary to know the position and shape of all light sources in the real scene. It is necessary to collect image of real environment for knowledge of geometry and information about position and shape of all light sources. In standard solution of using Image Based Lighting (IBL) in AR all images are captured from single digital camera (low dynamic range - LDR). We present research results of aspect using the High Dynamic Range (HDR) images from two digital cameras and IBL in AR environment to better lighting calculation and extraction of the strongest light sources.

1 Introduction

Augmented Reality (AR) is a field of computer research which deals with the combination of real world and computer generated data. AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world [3].

The beginnings of AR, date back to Ivan Sutherland's work in the 1960s (Harvard University and University Utah). He used a seethrough HMD to present 3D graphics. Milgram et al 1994, introduce the reality-virtuality continuum that defines the term mixed reality and portrays the „link” between the real and the virtual world (Fig. 1).

If the real world is at one of the ends of the continuum and VR (i.e. computer-generated, artificial world) is at the other end, then the AR occupies the space closer to the real world. The closer a system is to the VR end, the more the real elements reduce [13].

AR can potentially apply to all senses, including hearing, touch, and smell. Certain AR applications also require removing real objects from the perceived environment, in addition to adding virtual objects [2]. AR systems have been already used in many applications as military, emergency services (enemy locations, fire cells), prospecting in hydrology, ecology, geology (interactive terrain

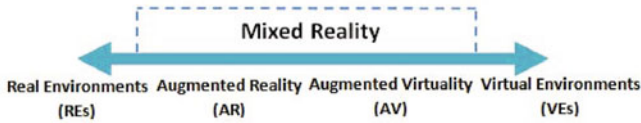


Fig. 1. Reality-Virtuality continuum

analysis), visualization of architecture (resurrect destroyed historic buildings), enhanced sightseeing, simulation (flight and driving simulators), virtual conferences, entertainment and education (game eg. ARQuake).



Fig. 2. Virtual reconstruction of destroyed historic building (Olympia, Greece) (source: official project site ARCHEOGUIDE, <http://archeoguide.intranet.gr>)

Main claims put for AR systems:

- Combines real and virtual objects in a real environment;
- Runs interactively, and in real time;
- Registers (aligns) real and virtual objects with each other;
- Virtual augmentation should be imperceptible to real object;
- Real objects can be augmented by virtual annotations;
- Virtual augmentation can be without limitations viewed and examined;
- User should have capability of simple entrance and omission of system;
- Multiple users can see each other and cooperate in a natural way;
- Each user controls his own independent viewpoint;

All AR systems should to aim to realise this claims. It is important to draw attention to fact that all this claims are very hard to realize or sometimes impossible to realize with using current methods and technology.

2 Difficulties of Transmitting Information of Light Sources from Reality to Augmented Reality

In Virtual Reality (VR) environment information about all light sources (position, shape, brightness, etc.) is defined by environment creator. In AR environment this information are taken directly from real environment. It is not easy task, but transmission information of light sources is very important part of AR

system, because most essential aspects to make virtual objects look like real ones is lighting. To calculate adequate shadows it is necessary to know the position and shape of all light sources in the real scene.

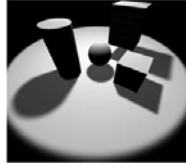


Fig. 3. Soft shadows generated with using "Smoothies" algorithm (source: [4])

Difficulties of transmitting information of light sources:

- The geometry of the real environment is usually not known;
- We do not have information about position and shape of all light sources.

2.1 Capture the Real Environment Image Methods

It is necessary to collect image of real environment for knowledge of geometry and information about position and shape of all light sources. From many used methods most important are:

- Capture image of mirrored sphere (In its reflection nearly the entire environment can be seen);
- Merging multiple images into single image;
- Using fisheye lens (all the ultra-wide angle lenses suffer from some amount of barrel distortion);
- Capture only selected part of environment (e.g. ceiling);
- Using special rotating cameras.

Currently Used Methods (Mirrored Sphere): One Camera Method.

This is very popular method because only one camera is needed to observe environment and capture real environment image (Figure 5). Camera captures a video of the scene which afterwards is overlaid with virtual objects. In this method a mirrored sphere is used which reflects the environment. The sphere

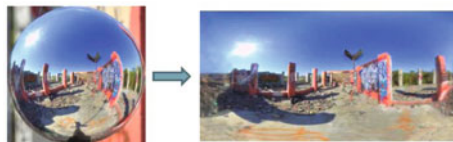


Fig. 4. Mirrored sphere image and its environment image (source: [12])

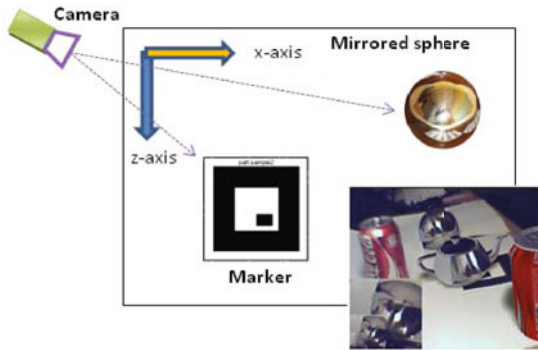


Fig. 5. One camera method

is in a fixed position relative to a marker of a visual tracking system. Marker is a characteristic tag, placed in visible place of scene, used in AR systems to compute virtual object positions.

This method have lot of disadvantages like:

- complicated extraction of mirror sphere from scene image
- bad resolution of extracted mirrored sphere
- mirrored sphere must be placed within observed scene

Two Cameras Method. This method is free from most of disadvantages One Camera Method. Reduction of this disadvantages was obtained by using two cameras (two devices: first to capture real environment image and second to observe environment).

In standard setup (Figure 6) first camera capture real environment image (mirrored sphere or fiseye lens (Figure 7)).

Second camera captures a video of the scene which afterwards is overlaid with virtual objects.

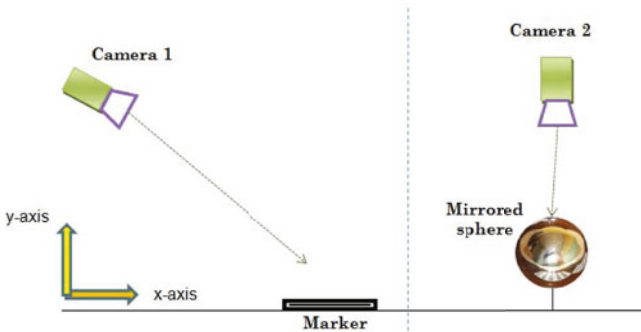


Fig. 6. Two cameras method



Fig. 7. Fisheye lens

2.2 High Dynamic Range Images (HDR)

Real-world scenes usually exhibit a range of brightness levels greatly exceeding what can be accurately represented with 8-bit per-channel images [5].

High dynamic range imaging (HDRI or just HDR) is a set of techniques that allow a greater dynamic range of luminances between the lightest and darkest areas of an image than standard digital imaging techniques or photographic methods merging of multiple photographs, which in turn are known as low dynamic range (LDR) [5] or standard dynamic range (SDR) [11] images.

High-dynamic-range photographs are generally achieved by capturing multiple standard photographs, often using exposure bracketing, and then merging them into an HDR image

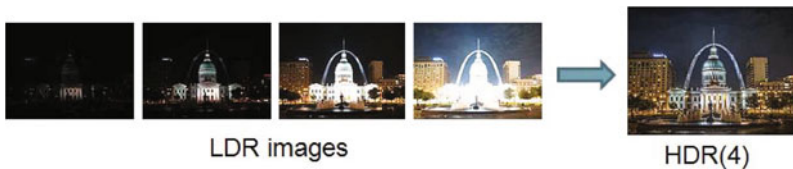


Fig. 8. HDR creation

3 Proposal of a New Method: Three Cameras Method

In this method two cameras capture real environment image (mirrored sphere or fisheye lens). It belongs to modify view parameters (light filters, exposure times, bracketing), in order to collect series of image with different exposure. Images from two cameras was taken advantages to create HDR image which is used as a environment image.

Third camera captures a video of the scene which afterwards is overlaid with virtual objects.

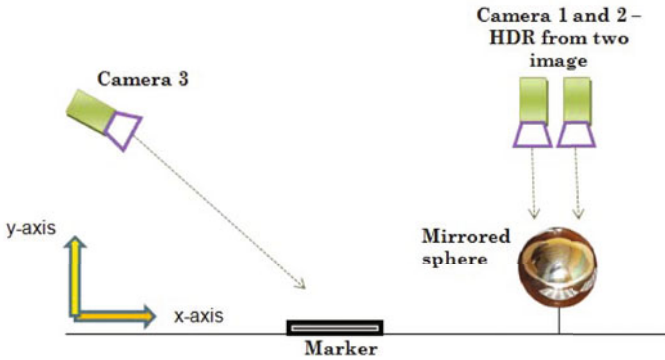


Fig. 9. Three cameras method

3.1 Hypothesis

Three cameras method give better light sources prediction.

Inspiration to frame a hypothesis: Due to the limited dynamic range of digital cameras it is very likely that a lot of pixels have the same value (pure white), although the real light emission of the objects differs greatly:

- Bulb placed on window in sunny day,
- Two sources with considerable difference between brightness neighbourhood placed (Figure 10).

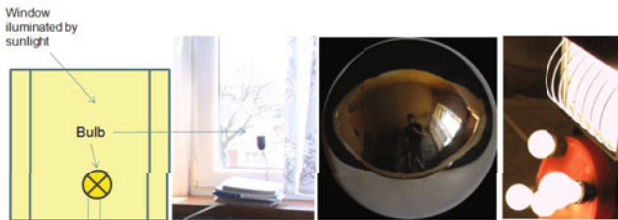


Fig. 10. Bulb placed on window in sunny day and five light sources neighbourhood placed

3.2 Experiment and Experimental Proof of Investigative Hypothesis in AR Research

Experiments goals is to test earlier framed hypothesis, which has been confirmed in result of experiment.

The Results of this experiments was HDR images (Figure 12), which advanced analysis with using Light Mask [1] technique allow test the hypothesis.

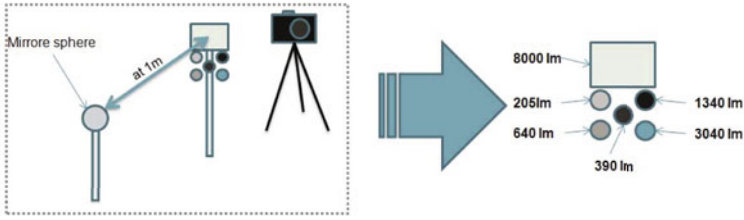


Fig. 11. Scheme of experiment

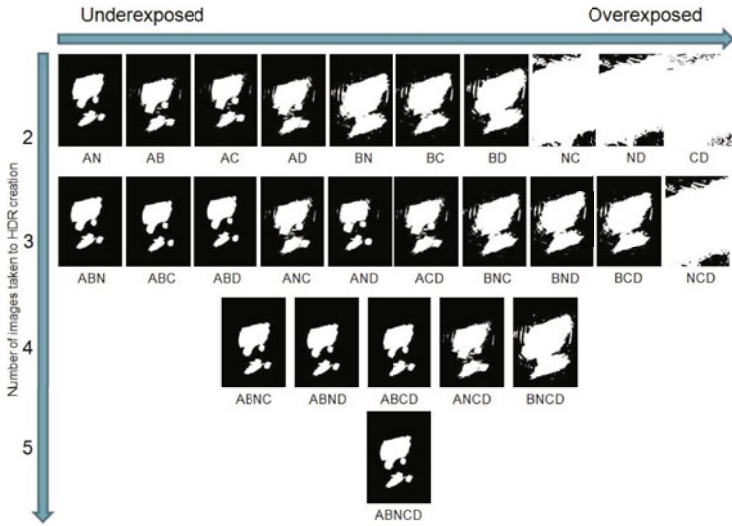


Fig. 12. Results of experiment

Experimental proof applies in natural sciences, in AR was use to verification of hypothesis as using Image Based Lighting (IBL)[\[9\]](#) and image based shadows in real time AR environment[\[6\]](#).

3.3 Experiment Conclusion

Incrementation of amount of LDR image taken to create HDR is not tantamount to growth the knowledge about position and shape of all light sources in the real scene. Most important factor is exposures of images taken to create HDR.

- Number of image taken to create HDR environment map don't influence proportional to the knowledge about position and shape of all light sources in the real scene
- After proper selecting sequence of Low Dynamic Range images (LDR)[\[1\]](#), the knowledge about position and shape of all light sources in the real scene

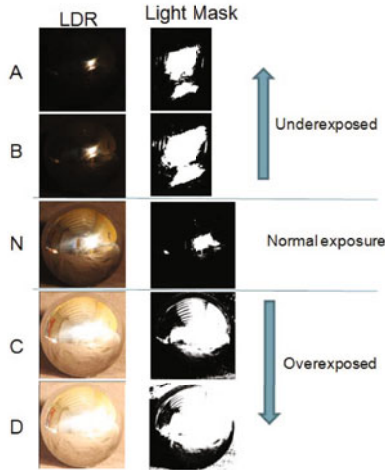


Fig. 13. Image taken to experiment

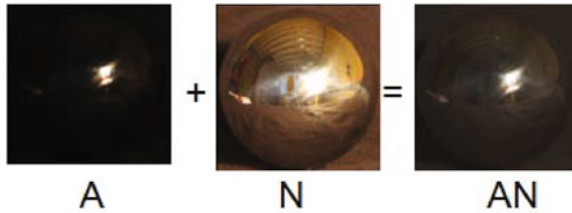


Fig. 14. HDR created from two LDR images

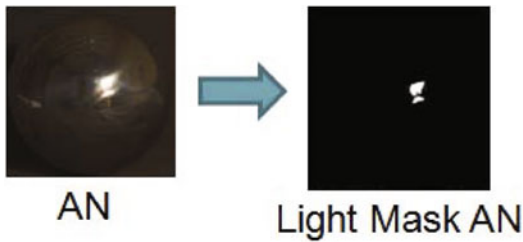


Fig. 15. Light Mask

gotten from HDR created from 2 LDR images is comparable to the knowledge got from HDR created from 3 or more LDR images

Depending on kind of image taken to create HDR, knowledge about position and shape of all light sources in the real scene is significantly different

- Knowledge about position and shape of all light sources in the real scene got from HDR created from 2 or more LDR is biggest as one of image is significantly underexposed

Benefits and Limitations of Methods. Benefits of methods:

- Increase of knowledge about position and shape of all light sources in the real scene

Limitations of methods and problems with its application:

- Using multi-camera systems (2 camera) - Calibration (Cameras observe ball from different angle)
- It is indispensable for obtainment of maximum efficiency of method of three camera setup individually cameras exposition for scene

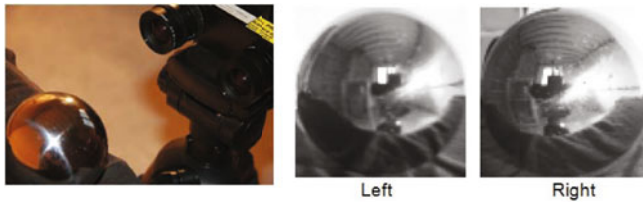


Fig. 16. Different camera angle observation

4 Conclusions and Future Work

New method was presented which give better light sources prediction in real environment. Usefulness of new method was experimentally confirmed. However, some problems with its application have appeared related to using multi-camera systems (calibration). Next work are planned with improvement this method at employment of other technology:

- Apply adaptive light filters
- Apply artificial intelligence to light sources prediction
- Using one camera to create HDR with using a automatic system to change light filter

References

1. Akyüz, A.O., Reinhard, E.: Color appearance in high-dynamic-range imaging. *SPIE Journal of Electronic Imaging* 15(3) (2006)
2. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. *IEEE Computer Graphics and Applications* 21(6), 34–47 (2001)

3. Azuma, R.T.: A survey of augmented reality. In: Computer Graphics, SIGGRAPH 1995 Proceedings, Course Notes no.9: Developing Advanced Virtual Reality Applications, pp. 1–38 (August 1995)
4. Chan, E., Durand, F.: Rendering fake soft shadows with smoothies. In: Proceedings of the Eurographics Symposium on Rendering, pp. 208–218. Eurographics Association (2003)
5. Cohen, J., Tchou, C., Hawkins, T., Debevec, P.E.: Real-time high dynamic range texture mapping. In: Proceedings of the 12th Eurographics Workshop on Rendering Techniques, London, UK, pp. 313–320. Springer, Heidelberg (2001)
6. Haller, M., Supan, P., Stuppacher, I.: Image based shadowing in real-time augmented reality. *International Journal of Virtual Reality* (2006)
7. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems* E77-D(12) (December 1994)
8. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: a class of displays on the reality-virtuality continuum. In: Das, H. (ed.) Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference. Proc. SPIE Telemanipulator and Telepresence Technologies, vol. 2351, pp. 282–292 (1995)
9. Supan, P., Stuppacher, I.: Interactive image based lighting in augmented reality. In: *Digital Media*. Upper Austria University of Applied Sciences, Hagenberg/Austria (2006)
10. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the AFIPS 1968 (Fall, part I), fall joint computer conference, part I, December 9–11, pp. 757–764. ACM, New York (1968)
11. Vonikakis, V., Andreadis, I.: Fast automatic compensation of under/over-exposed image regions. In: Mery, D., Rueda, L. (eds.) *PSIVT 2007*. LNCS, vol. 4872, pp. 510–521. Springer, Heidelberg (2007)
12. Witte, K.: *How to shoot a chrome ball for hdri* (2009)
13. Zlatanova, S.: *Augmented reality technology* (2002)

The Context-Sensitive Grammar for Vehicle Movement Description

Jan Piecha and Marcin Staniek

Informatics Systems Department of Transport, Faculty of Transport
Silesian University of Technology Krasińskiego 8, 40-019 Katowice, Poland
jan.piecha@polsl.pl, marcin.staniek@polsl.pl

Abstract. The paper introduces a method allowing to spot image of vehicles, registered by a video camera. The object's manoeuvres on and between traffic lanes are one of the most important task being under consideration in this contribution. The elaborated algorithms of the vehicle's movement analysis were presented in the paper. The description language, with its syntactic primitives, was introduced and implemented for the movement assignment, description and analysis.

1 Introduction

The traffic control systems are based on different technologies concerning the data registration, computing techniques and the data distribution methods [1]. The most fashionable and encouraging techniques are based on digital camera interfaces for roads traffic monitoring and registration.

The images pre-processing and processing, for properly carried on classification are key problems being under consideration recently.

The collected data is recorded in memory units of the controlling systems. The specifically assigned objects, with their characteristic shape descriptors and the objects location, on the observation field scenery, are considered in the paper. The obtained data set simplifications allow us describing the fast moving objects within the camera observation field, by remarkable reduced size of the data set, covering the full dimension of the camera observation area.

The real time controlling process is put into frames of the data sampling rate that is defined by the time window, indicating the control data grid [1] satisfactory for the controlling process quality assumptions.

The transportation model description demands were already discussed in several works; among them [2], [4]. The time limit has to be matched with the controlling unit speed, covering not only the demands of the traffic smooth flow but also of all calculations, fulfilling the algorithms computing time limit.

The digital cameras, available on a market today, are sampling the image stream 25 times per second, recording each car; driving 50 km/h, at 100 m passage (of the road) 182 times [3]. That is why the clocking rate produces immense set of the data, much bigger than it is needed for this slow process description.

These troublesome image sequences observation and description (in a real time mode) expects necessary simplifications of these movie data stream [2], [3].

In each data frame the objects class with its localisation on the scenery has to be distinguished.

The analysed image sequences, with a necessary background, can be assigned by adequate descriptors of the syntactic language L. The language symbolic variable identifiers, analyse and assign the vehicle’s localisation, with their movement histogram.

The objects description symbols define locations and moving directions of the vehicle. Moreover, the language components allow identify the movement on the transportation process scenery, by means of specific syntactic grammar. The grammar items generate sequences of symbols, defining the traffic objects and their manoeuvres.

The elaborated grammar generates words and sequences of words, allowing describe each manoeuvre on and between the traffic lanes, overtaken vehicles, driving back and towards, etc. The defined symbols composition indicate all regular and not illegible manoeuvres. The specific sequences of symbols, combined with parsers library, indicate the manoeuvre type.

2 The Vehicles Trajectories Syntactic Descriptors

2.1 The Syntactic

Main assumptions to the description method, provide an efficient assignment of the traffic scenery with objects and their coordinates, using several symbols only, instead of using traditional bitmaps.

Definition 1. *The syntactic symbols identification method allows as assigning the vehicles trajectory, used for recognition process of its movement, by means of the defined primitives placement analysis. The used primitives indicate the geometrical characteristic parts of the analysed objects, with their localisation on the observation field.*

The traffic route recognition concerns identification of the vehicles primitive descriptors that are used for the object assignment [2], [3]. The analysis procedures indicate several principles, implemented in the specified language.

In Fig.1 the trajectory primitives are indicated. The vehicles trajectory was divided into several units, showing the object localisation, during its movement

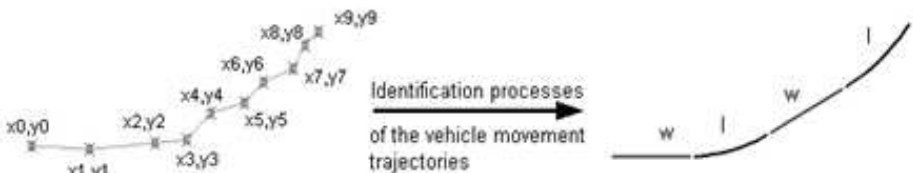


Fig. 1. The syntactic primitives assignment

along the traffic lane [3]. On a left side of the figure geometrical coordinates were assigned. On the right side, the syntactic symbols were presented. The trajectory description area is limited into the camera observation field and its projection onto a road surface, assigning the trajectory under analysis.

The distinguished shapes of the elementary units allow us indicate the vehicle’s movement using the defined symbols (Fig.2).

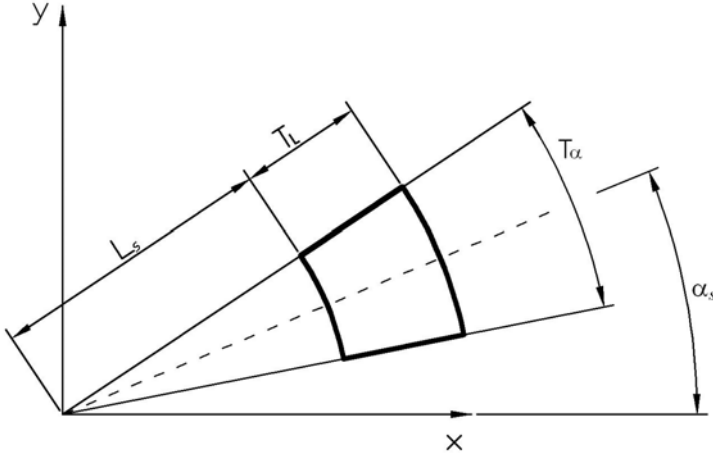


Fig. 2. The vehicles trajectory primitives assignment

The vehicle’s movement assignment primitives, were defined (1) by four attributes, according to the illustration in Fig.2.

$$S_P(L_S, T_L, \alpha_S, T_\alpha) \tag{1}$$

where: L_S - length of the primitive from axis centre $(0, 0)$, α_S - angle between the primitive and axis X, T_L i T_α - the length tolerance L_S and the primitive angle tolerance α_S ,

The length T_L tolerance and the angle T_α tolerance, define differences margins for the vehicle’s movement trajectories identification.

The analysis algorithm distinguishes the not eligible manoeuvres on the road, in the specified road zone. They are indicated by an evidence of traffic law, called the traffic incidents [3]. The analysis parameters are provided by an angle coordinates, like [2], [3]:

1. Driving ahead: w (1.8, 0.4, 0, 0.16),
2. Turning left: l (1.8, 0.4, 0.16, 0.16),
3. Turning right: p (1.8, 0.4, -0.16, 0.16),
4. Reverse: c (1.8, 0.4, π , 0.16).

The above symbols describe the vehicle’s movement trajectories [2], [3] building up the language primitives, indicating coordinates of the vehicle’s placement.

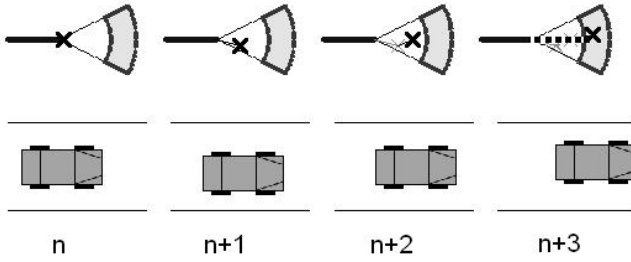


Fig. 3. The vehicles trajectory assignment in an image sequence

In case the current coordinates of the descriptor are close to any symbol, of the description language, the subsequent fragment of the vehicles trajectory is indicated (Fig.3). In the analysed video frames, the vehicle’s localisation is appointed by a sign x .

When the localisation vectors are close to any of the language primitives, the dotted line indicates movement trajectory of the vehicle (Fig.3). There are direction symbols illustrating: driving straight, turning left and right or changing the vehicle’s placement (for the vehicle’s speed $v=60\text{km/h}$).

Combining a data sampling rate, with the vehicle’s localisation, one can find their speed and various accidents evidences, with sample measures of the vehicle’s dynamics.

The recorded number of vehicle’s placement (in the video sequence) may indicate accelerations, slowing down or braking of vehicles.

The movement trajectories, observed on the traffic lane background, allow us indicating all illegible manoeuvres of vehicles, as well.

3 The Trajectories Description Language

The discussed primitives, used for the vehicles scenery description, create the description alphabet of the introduced language. The language grammar provides us with description items of the transportation network.

The language, with its alphabet, recognises every combination or subset of these symbols [5]. The symbols produce words and sentences of the language words. The undefined or unrecognised movement cases are not classified by the algorithm formulas.

The fundamental task of the syntactic assignment method, supports the vehicle’s movement and localisation recognition, based on the video data sequence.

The vehicle’s movement description and recognition mechanisms are encoded by the language grammar, with several combination of these elementary components.

The definitions of the language grammar [6] are expressed by several relations of:

$$G = (\Sigma_N, \Sigma_T, P, S) \tag{2}$$

where: Σ_N - defines a set of non-terminals, Σ_T - set of terminals, P - is a finite set of rules or productions, S - is the starting symbol $S \in \Sigma_N$.

The alphabet Σ calls the limited set of symbols, as: $\Sigma = \Sigma_N \cup \Sigma_T$ and $\Sigma_N \cap \Sigma_T = \emptyset$. The set of terminals Σ_T contains the defined {w-driving ahead, l-turning left, p-turning right, c-reverse} of the trajectories descriptors. The non-terminals Σ_N consists of variables, used in words construction of the language body, then the production set P is used for the words construction by operations:

$$\eta \rightarrow \gamma \tag{3}$$

According to the boundaries, given by means of these productions [8], four grammar types were distinguished: unrestricted, context-sensitive, context-free and regular.

The description of the road traffic manoeuvre in the context-sensitive grammar, uses productions expressed by the following relations:

$$\eta_1 A \eta_2 \rightarrow \eta_1 \gamma \eta_2 \tag{4}$$

where: $\eta_1, \eta_2 \in \Sigma^*$, $\gamma \in \Sigma^+$, $A \in \Sigma_N$,

The symbol A can be replaced by an empty sequence of symbols γ , when A appears in context of symbols η_1 and η_2 .

3.1 The Traffic Context-Sensitive Grammar

The context-sensitive grammar assigns the rules of membership, within a level of words formation, of transportation objects' free modelling. It allows describing the manoeuvres as: a vehicle's lane turnover, overtaking and U-turning (back).

The turning manoeuvre, in the grammar construction of the turning radius, is observed in a range of 5 to 25 meters. It allows identify the manoeuvre observed on every crossroad type, in cities.

The grammar expression for the manoeuvre of the vehicle's turning - left or right is expressed as:

$$G_{turn_left} = (\Sigma_N, \Sigma_T, P, S)_{turn_left} \tag{5}$$

where: $\Sigma_N = \{S, A, B, C\}$, $\Sigma_T = \{l, w\}$, $P = \{S \rightarrow All, All \rightarrow lBl, Al \rightarrow Awl, Awl \rightarrow wwlB, Awl \rightarrow wlB, lBl \rightarrow llC, Bl \rightarrow Bwl, Bwl \rightarrow wwlC, Bwl \rightarrow wlC, lC \rightarrow lwC, lwC \rightarrow lwwC, C \rightarrow l\}$

By the defined rules of membership of P , 27 words of the alphabet Σ were obtained. The largest lock radius contains the lock primitives l of the vehicle, separated by two primitives of a direct movement w .

A graphical interpretation of this analysis, concerning the grammar words of the lock G_{turn_left} , has been shown in Fig. 4. The lock symbol, visible in the figure, has the radius $R=5m$.

In the turning trajectories algorithm implementation, the L lengths primitives in range of 1.8m to 2.2m were selected. It means that the vehicle is detected twice, driving 90 km per hour, or once for speed above 200km/h in the sequence of 25 images per second.

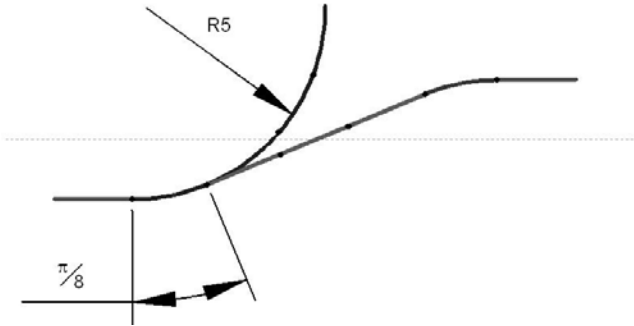


Fig. 4. Illustration of the turn primitive angle

Changes of the vehicle’s position, between the traffic lanes, are also an important manoeuvre, needed for the transportation network states complete description. Finding the words that belong to the manoeuvres description grammar rules, are one of the formal language description key items.

The description language structure contains units for: driving ahead, back and the vehicle’s locks. The sequence of symbols for vehicles movement on the traffic-lanes, consists of combinations of the lock primitives.

The turning to the left manoeuvre grammar (on the traffic lane) consists of the following symbols:

$$G_{change_left} = (\Sigma_N, \Sigma_T, P, S)_{change_left} \tag{6}$$

where: $\Sigma_N = \{S, A\}$, $\Sigma_T = \{l, w, p\}$, $P = \{S \rightarrow ww l A p w w w, A p w \rightarrow A p w w, A p \rightarrow A w p, A w p \rightarrow A w w p, A w w p \rightarrow A w w w p, l A \rightarrow A l, A l w \rightarrow A l l w, A l p \rightarrow A l p, A l \rightarrow w l\}$

The belonging rules P , for the traffic lane changes, allow us finding up to 16 words of these states description, using the alphabet Σ . The vehicle’s manoeuvre shape, on the traffic lane was illustrated in Fig 5.



Fig. 5. The traffic lane change illustration

The grey area shows an example of an eligible change zone (surface) on the traffic lane: G_{change_left} .

The grammar constructions for further traffic lane changes descriptors are defined similarly to the above manoeuvre of turning left.

4 The Movement Trajectory Analysis and Acceptation

The automaton for the language formulas analysis, allows us defining the input words that define the description language formulas [4], [5]. The calculations made by the automaton work, on checking the current movements, were defined by the transitions function δ .

Reading the symbol in the sequence and analysing the table of states, of the control unit, the next state of the automaton is found [6]. The word identification means the trajectory symbols finding, with characteristic relations between them. The analysis concerns relations to the language L with its all grammar G_n relations (Fig. 6).

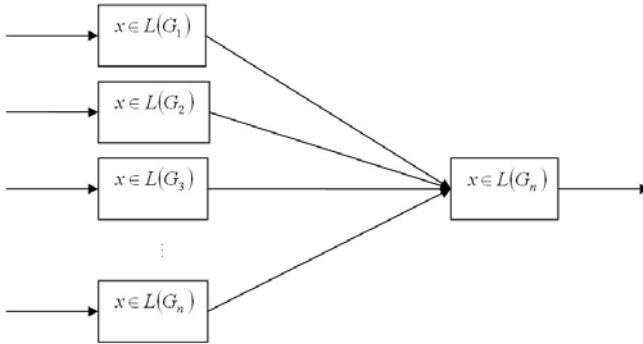


Fig. 6. The block diagram of words acceptance mechanisms

For more precise (complex) description of the vehicle manoeuvres the description symbols have to be reduced into smaller fragments of the vehicle’s shape. In spite of the language complexity increase the separate analysis of the descriptors combination will simplify remarkable the language grammar s and the analysis algorithms.

4.1 The Automaton for Vehicles Trajectory Movement Description

The automaton used for the description words analysis and accepting [6] (of the description language), allows also identify the scenery. The Turing machine and its versions, as: a finite-state automaton, push-down automaton and linear-bounded automaton [7] support the recognition algorithms. This machines are counting the function values for the given arguments [8]. The linearly bounded automaton is called single-bend Turing machine, with a stop mode. Its alphabet uses two special symbols; the ending delimiters reducing a direct entrance into cells saving an input data of the automaton. The work introduces some programming procedures of the automata theories, constructed as a parser that corresponds to syntactic descriptors of the traffic accidents, in transportation networks.

The linearly-bounded automaton defines a not deterministic model with the parser constructions used for identification of the network states that can be expressed by an equation:

$$P = (Q, \Sigma, \delta, q_0, F) \tag{7}$$

where: Q – finite, not empty set of states, Σ – finite inputs alphabet, δ – the transition function (for mapping), q_0 – starting delimiter $q_0 \in Q$, F – ending delimiter $F \subseteq Q$.

The defined parsers of the road incidents identifiers [1] concern manoeuvres: turn of vehicle, traffic lane change, overtaking and turning back the vehicle.

The parser construction for a turn left manoeuvres is assigned by the expression:

$$P_{turn_left} = (Q, \Sigma, \delta, q_0, F)_{turn_left} \tag{8}$$

where: $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$, $\Sigma = \{w, l\}$, δ – is defined by table 1, $q_0 = \{q_0\}$, $F = \{q_9\}$.

Table 1. The transition function δ of the vehicle’s turning left parser

	<i>l</i>	<i>w</i>
<i>q0</i>	(<i>q3</i> , <i>P</i>)	(<i>q1</i> , <i>P</i>)
<i>q1</i>	(<i>q3</i> , <i>P</i>)	(<i>q2</i> , <i>P</i>)
<i>q2</i>	(<i>q3</i> , <i>P</i>)	
<i>q3</i>	(<i>q6</i> , <i>P</i>)	(<i>q4</i> , <i>P</i>)
<i>q4</i>	(<i>q6</i> , <i>P</i>)	(<i>q5</i> , <i>P</i>)
<i>q5</i>	(<i>q6</i> , <i>P</i>)	
<i>q6</i>	(<i>q9</i> , 0)	(<i>q7</i> , <i>P</i>)
<i>q7</i>	(<i>q9</i> , 0)	(<i>q8</i> , <i>P</i>)
<i>q8</i>	(<i>q9</i> , 0)	

The linearly-bounded automaton P_{turn_left} allows us finding twenty seven words of the alphabet Σ for the vehicles turning manoeuvres.

The manoeuvre of the traffic lanes changing, for a left sided traffic, will be widening the traffic description symbols and a traffic network assignment. Its localisation will identify a parser P_{change_left} , as:

$$P_{change_left} = (Q, \Sigma, \delta, q_0, F)_{change_left} \tag{9}$$

where: $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}\}$, $\Sigma = \{w, p, l\}$, δ – is defined by graph (in Fig. 7), $q_0 = \{q_0\}$, $F = \{q_{14}\}$.

The construction P_{change_left} allows us accepting all sixteen words of the alphabet Σ ; being the language of manoeuvres, assigning the movement of the vehicle. The parser P_{change_left} of the traffic states was introduced in Fig 7.

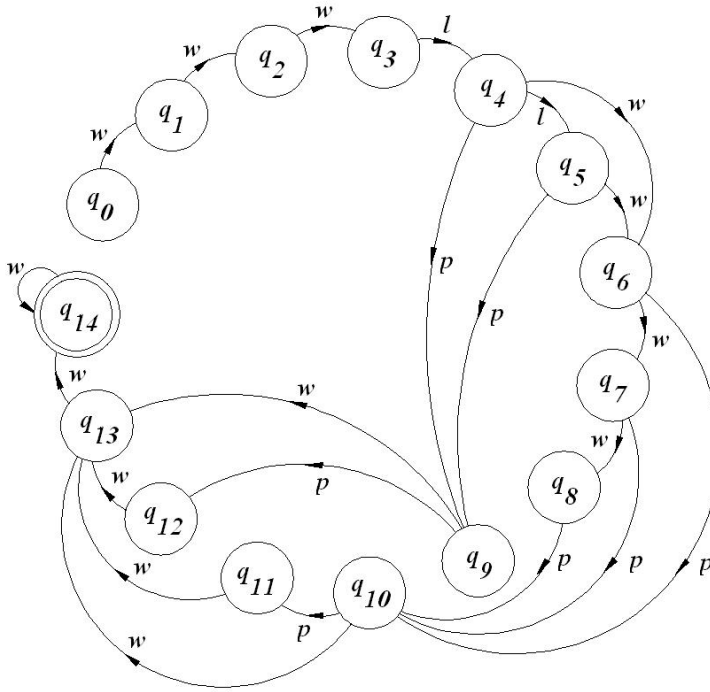


Fig. 7. The parser states graph P_{change_left} for vehicles changing manoeuvres

5 Conclusions

The real-time control machine, using the data recorded by digital camera, indicate a limited time-window for all processing units execution. That is why the calculation algorithms have to be simplified into a necessary level defined by the assumed data resolution.

Although the syntactic symbols language provide us with remarkable simplifications of the transportation objects description, number of tasks and the computing algorithms complexity can still be a challenge for further works searching fast methods of additional applications offering a video camera data file applications (as vehicles classes finding).

The carried on field testing experiments provided us with very satisfying results for further applications finding.

The context grammar of the language, allows us defining complex descriptors of the transportation vehicles states. The conclusions making automata identify and assign various manoeuvres of vehicles.

The implementation of the control machine has been done by a very fast hardware integrated processing unit based on the field programmable gate arrays - SPARTAN [9].

References

1. Roess, R., McShane, W., Prassas, E.: Traffic Engineering. Prentice-Hall, New York (2004)
2. Staniek, M., Piecha, J.: Vehicles trajectories movement description by means of syntactic method. International Scientific Journal: Transport Problems, Gliwice 4(4) (2009)
3. Staniek, M.: Metoda syntaktycznego opisu trajektorii ruchu pojazdów w sieci transportowej, PhD work, Gliwice (2009) (in Polish)
4. Fu, K.S.: Syntactic Pattern Recognition and Applications. Prentice-Hall, New Jersey (1982)
5. Fu, K.S.: Syntactic Methods in Pattern Recognition. Academic Press, New York (1974)
6. Aho, A.V., Ullman, J.D.: The theory of parsing, Translation and Compiling, Prentice-Hall Series in Automatic Computation, Vol. 1 (1972)
7. Chomsky, A.: Syntactic structure. Walter de Gruyter, Berlin (2002)
8. Hopcroft, J., Motwain, R., Ullman, J.: Introduction to Automata Theory, Languages and Computation, 2nd edn. Person Education (2001)
9. Płaczek, B., Staniek, M.: Moduły wideo-detektorów pojazdów ZIR-WD do sterowania i nadzoru ruchu drogowego, Praca badawcza: 512/11/475/06/FS-11. Zad. 1.3 Miernik parametrów ruchu drogowego WD-M. Research Report No. 8, Katowice (2007) (in Polish)

The Method for Verifying Correctness of the Shape's Changes Calculation in the Melting Block of Ice

Maria Pietruszka and Dominik Szajerman

Institute of Information Technology, Technical University of Łódź, Poland
drs@ics.p.lodz.pl

Abstract. The article shows the method of verification the accuracy of processing geometry by comparing the contours of the real block of ice with its model, during the melting process. This method is common, and it can be used for testing visualization of melting, sublimation and ablation phenomenon and other processes of the material decomposition. So far the reliability of geometric changes of the visualized ice has been assessed on the eye.

1 Introduction

The visualization of physical processes is one of the important uses of computer graphics. Specificity of visualization is the need of mapping the three components of an object's appearance: shape, material model, illumination model. The last two components allow the viewer to identify whether an object is made of ice, wax, wood, or stone, etc. It consists of a material color, its directional features, e.g. glossy, shiny, transparent, and spatial features e.g. little bumps. The shape of the object is usually the same and informs us about the functional features of the object e.g. table, chair.

However, the shape of the deformable objects changes. Decomposition of solids includes the ablation and sublimation [13] and the melting of wax [1] and ice. While the wax during the melting becomes plastic material whereas the ice becomes water. The other examples of decomposition are the wood burning [7], the erosion of stone [3] and the interactive sculpting in the virtual reality [2]. The opposite to the decomposition is the creation of solids. For example the hardening of lava, cement, wax or ice [1] and crystallization or solidification [6]. The third group of phenomena includes reversible transitions between solids and liquids [5,8] or viscoelastic objects [4].

Our research focuses on visualisation of sublimation and melting of ice in real-time rendering applications. In order to achieve suitable time of calculation all the algorithms were implemented on the GPU. It concerns not only the model of material for ice visualisation [11,10], but also the model of the deformation of the ice block during the melting [10]. Both models are empirical. The idea of the

deformation is based on the relationship between the speed of the melting and the irregularity of the ice surface, called the inter-phase surface [9]. It has allowed the application of the surface objects representation instead of the volumetric one used previously.

The main problem appeared in verification of the results, because so far both the appearance of ice and the changes of the shape were evaluated on the eye [1,2,3,4,5,6,7,8,13].

Such evaluation is not enough especially for empirical models. The result of our research is the original quantitative method of the verification of calculation of an object shape during the phase transition. In this paper there is an idea of the method presented. It includes the data acquisition and processing of the real and virtual object and the method of comparing of the processed data.

Moreover, there is presented the application of the method for verification of calculation of melting of four objects made of the ice.

2 Verification Scenario

The main goal of results verification is the comparison of calculations results with the natural process. There is necessary to ensure a set of physical conditions for real phenomenon and a set of analogous conditions in the virtual scene 3D in order to do the comparison.

Presented method considers following conditions:

1. virtual object is visualized using the tested method and the real object is photographed,
2. virtual and real object have the same initial shape,
3. the processing cycles of tested method correspond to the adequate photographs,
4. the viewpoint in the virtual scene corresponds to the real position and orientation of the camera,
5. projection parameters in the virtual scene are analogous to real camera parameters,
6. examined features of the virtual object shape agree with the features of the real object shape.

In order to meet the above assumptions one has to prepare two pipelines of data processing - one for the real object and one for the virtual object (fig. 1). Both use the same cast to make gypsum and ice models.

Pipeline A - *acquiring and processing for the real object data.*

1. Ice model preparation (water freezing in the cast).
2. Taking a sequence of photographs of the block of ice during melting.
3. Obtaining a contour sequence with the help of image processing.

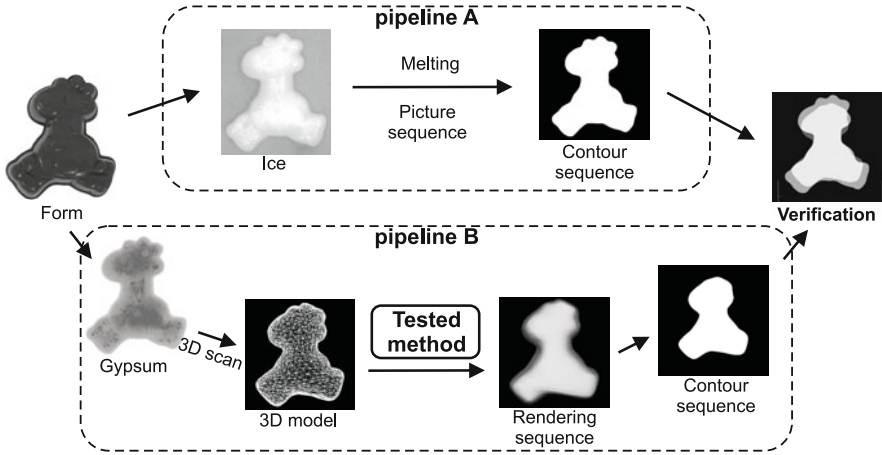


Fig. 1. Verification scenario

Pipeline B - acquiring and processing for the virtual object data.

1. Gypsum model preparation (in the same cast).
2. A 3D scan in order to obtain a triangle mesh model.
3. Rendering a sequence of images using the tested method of the ice melting calculation.

The revision of the results of geometry is done by comparing the sequences of contours generated from the both pipelines described above (see section 3). The term contour is understood as an object projection onto the observing plane.

2.1 Acquiring and Processing for the Real Object Data

The experiment described below, shows an example of implementing this part of the pipeline A, which gives the output sequence of photographs the real object (fig. 2).

Lump of ice was placed on the absorptive ground not to have to filter the image spilling water. The camera was above the object and the lens were pointing down. It corresponds to the contour rendering of a virtual object in the horizontal surface. The camera was removed as high as it was possible in the studio and how the optical zoom allows. Maximum optical zoom gives minimal viewing angle available in the camera. It is the best zoom of orthogonal projection offered by the camera.

The temperature of the surrounding estimated between $25^{\circ}\text{C} - 35^{\circ}\text{C}$. The initial of the block of ice estimated -10°C .

The sequence of single cubes melting in these conditions lasted over 2 hours. The photographs have been performed every 40 seconds, which gave about 200 photographs for the melting sequence.



Fig. 2. Configuration of photo position

In order to calculate the contours in the photographs are preprocessed (fig. 3). The ice changes the appearance during the melting. The thinner it is, the more clear it becomes. This makes it difficult to find the correct contours in a sequence of these photographs which are made in the later stages of melting (fig. 4).

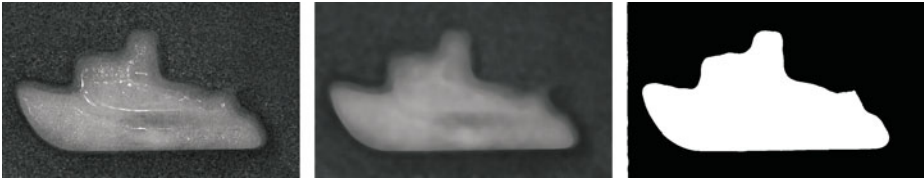


Fig. 3. The steps of picture's preprocessing: first there is a conversion from the colour to the gray-scale, then a median filtering removes a noise, and next the threshold makes a two-colour contour image



Fig. 4. Transparency of the outline of melting facility. The numbers of photographs: 1, 50, 100.

2.2 Acquiring and Processing for the Virtual Object Data

After obtaining a gypsum model from the same cast that was used for making the block of ice, the model is 3D scanned. The tested method is based on triangle



Fig. 5. Triangle mesh model of the battleship: a) mesh, b) two-colour rendering, c) the 3D look after rendering by the lighting

meshes, thus the chosen scanning method converts the gypsum model to the representation of its surface. Next the calculated triangle mesh model is processed and rendered by testing application (fig. 5). The application renders the model using only one colour without lighting and texturing. At the beginning of this section the rendering parameters which are set in the application, are discussed. Finally, the rendered images are the two-colour contour images (fig. 5b).

3 Contour Matching Function

The contour of the object refers to the object’s projection on the horizontal surface. The measure degree of the contour’s similarity (P) from synthetic picture and photograph takes place (fig. 6a):

$$P = \frac{S_A + S_B}{S_A + S_I + S_B}, \tag{1}$$

where:

S_I - the area of the common part of the contours,

S_A - the area of the contour from the photograph excluding the common part,

S_B - the area of the contour from the synthetic picture excluding the common part.

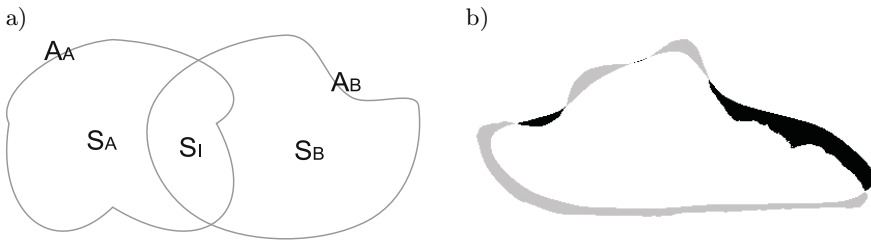


Fig. 6. Signs used in much of the degree of similarity between the contours: a) A_A - the area of the real object projection, A_B - the area of the virtual object projection, $S_I = A_A \cap A_B$ - number of pixels in the intersection, $S_A = A_A \setminus S_I$ - the area of the real object’s contour excluding the common part, $S_B = A_B \setminus S_I$ - the area of the virtual object’s contour excluding the common part, b) color code: S_A - grey, S_B - black, S_I - white

All areas are calculated as the number of image pixels; $P \in < 0, 1 >$, $P = 0$ for identical contours and $P = 1$ for separable contours. This measure is convenient in the interpretation. The similar measures are used in classification methods and in the image's segmentation [12].

There is accepted that the contours are sufficiently similar, to consider the quality of visualization of a sufficiently good, if the test sequence of photographs and images of synthetic value of P shall not exceed 0.20. Figure 6b) shows two examples of the contours positioned relatively to each other in the way to get values of $P = 0.20115$. Therefore, it shows that for such values of P the outlines of objects are similar.

4 Tests

The proposed method was used to verify the calculation of the ice melting by Szajerman [10]. The test was performed for objects derived from the four forms, called: giraffe, aircraft, ship, battleship. The tested blocks of ice weighed about 250 grams. The corresponding blocks of gypsum were approximated by a mesh of over 11000 triangles. Figure 7 and tab. 1 present the results of the comparison of contours for photography 1, 33, 67, 100. Similarity of the contours do not exceed $P_{100} = 0.20$. Given the similarity of input contours which amount to $P_1 = 0.07$ means the difference in calculated shape and photographed shape at around 10%.

Table 1. Similarity of contours (P) for selected images from a sequence of melting

model	P_1	P_{33}	P_{67}	P_{100}
giraffe	0.068818	0.081004	0.110698	0.137467
aircraft	0.027969	0.048929	0.079185	0.10399
ship	0.044926	0.073091	0.082767	0.120915
battleship	0.07131	0.102844	0.139845	0.178187

Deterioration of the similarity of the contours in the subsequent measure cycles is the accuracy of the calculation method for calculating the deformation of the object in the process of melting. However, it should be noticed, that the method of calculating the contour of the surface and also the experiment could enter some more mistakes. Contour of the input from the photographs and contour of the input from the synthetic image are different for several reasons:

1. water increases its volume coagulating in the form, and gypsum is not,
2. uncertainty of the 3D scanning of the gypsum block and approximation of its surface to the mesh of triangles,
3. lens imperfections, for example lens distortion,
4. uncertainty of mapping the dimensions, angles and other relationships between the environment and the three-dimensional scene



Fig. 7. The contour's similarity: a) P_1 , b) P_{33} , c) P_{67} , d) P_{100} ; grey - the real object's contour (S_A), black - the virtual object's contour (S_B), white - the common part of contours (S_I)

The rate of melting depends on the rate of heat flow, and thus from the geometrical features of the object. Therefore, the initial match between the contours impinges on the further comparison (tab. [1](#)). An additional reason for increasing the value of P is a contraction of the crop pixels constituting the contours. There are fewer pixels, so each disorder is becoming increasingly apparent.

5 Conclusions

There was presented a new approach to the results' verification of visualization and simulation algorithms. The tested algorithms are able to consider the changes in object's shape under the physical conditions, which are especially melting, sublimation and ablation phenomena, and also other process of the material's decomposition. The proposed method of verification allows not only

qualitative, but also quantitative assessment of visual results to be given by a tested method. It is ensured by a proposed measure of P .

The method was used in testing of melting and sublimation algorithms visualization [9,10]. It helped to prove that the tested algorithms give a plausible visual effect of melting and sublimation, and their results are similar enough to the real phenomena.

References

1. Carlson, M., Mucha, P.J., Van Horn III, R.B., Turk, G.: Melting and Flowing. In: Proceedings of the 2002 ACM SIGGRAPH (2002)
2. Chen, H., Sun, H.: Real-time Haptic Sculpting in Virtual Volume Space. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM, New York (2002)
3. Dorsey, J., Edelman, A., Jensen, H., Legakis, J., Pedersen, H.: Modeling and rendering of weathered stone. In: Proc. SIGGRAPH, pp. 225–234 (1999)
4. Goktekin, T.G., Bargteil, A.W., O'Brien, J.F.: A Method for Animating Viscoelastic Fluids. In: SIGGRAPH 2004 Papers, pp. 463–468 (2004)
5. James, D.L., Pai, D.K.: ArtDefo: Accurate Real Time Deformable Objects. In: SIGGRAPH 1999, pp. 65–72 (1999)
6. Kim, T., Lin, M.C.: Visual Simulation of Ice Crystal Growth. In: Eurographics/SIGGRAPH Symposium on Computer Animation, pp. 86–97 (2003)
7. Melek, Z., Keyser, J.: Multi-Representation Interaction For Physically Based Modeling. In: Proceedings of the 2005 ACM symposium on Solid and physical modeling, pp. 187–196 (2005)
8. Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., Alexa, M.: Point Based Animation of Elastic, Plastic and Melting Objects. In: Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp. 141–151 (2004)
9. Szajerman, D.: Mesh representation of simply connected 3d objects in visualisation of melting phenomena. *Machine Graphics & Vision* 15(3/4), 621–630 (2006)
10. Szajerman, D.: Visualisation of melting and sublimation phenomena. PhD Thesis. Technical University in Łódź, Poland (2010)
11. Szajerman, D., Pietruszka, M.: Real-time Ice Visualisation on the GPU. *Journal of Applied Computer Science* 16(2), 89–106 (2008)
12. Tomczyk, A., Szczepaniak, P.S.: Segmentation of Heart Image Sequences Based on Human Way of Recognition. LNCS, pp. 225–235. Springer, Heidelberg (2009)
13. Varadhan, H., Mueller, K.: Volumetric Ablation Rendering, pp. 53–60. The Eurographics Association (2003)

A Real Time Vehicle Detection Algorithm for Vision-Based Sensors

Bartłomiej Płaczek

Faculty of Transport, Silesian University of Technology
ul. Krasińskiego 8, 40-019 Katowice, Poland
bartlomiej.placzek@polsl.pl

Abstract. A vehicle detection plays an important role in the traffic control at signalised intersections. This paper introduces a vision-based algorithm for vehicles presence recognition in detection zones. The algorithm uses linguistic variables to evaluate local attributes of an input image. The image attributes are categorised as vehicle, background or unknown features. Experimental results on complex traffic scenes show that the proposed algorithm is effective for a real-time vehicle detection.

Keywords: vehicle detection, vision-based sensors, linguistic variables.

1 Introduction

The information concerning presence of vehicles in the predefined detection zones on traffic lanes is essential for adaptive traffic control at signalised intersections. This information can be effectively acquired using vision-based vehicle detectors. The idea is not new. However, fast and robust image processing algorithms are still being sought to provide low cost vehicle video-detectors and to improve their performance.

In recent years many vision-based algorithms have been developed for the vehicle detection. Most of them have been designed to perform image segmentation and categorise all pixels of the input image as pixels of vehicles, pixels of background, shadows, etc. [1], [5], [6]. The background subtraction is one of the most popular methods. Several adaptive background models have been proposed for the traffic images segmentation [9], [10]. Different methods have been intended for the counting of vehicles when passing through so-called virtual loops. In [7], the statistical models have been applied to extract image features that allow us to categorise each state of the detector into three categories (road, vehicle head and body) and to recognise passing vehicles. The methods proposed in [2], [8] use the time-spatial images analysis for the task of road traffic flow count. Also 3-D shape models have been dedicated for the vehicle detection [3], [4].

The paper introduces a fast algorithm for the vehicles presence recognition in detection zones. It operates on local image attributes instead of the particular pixels. The image attributes are evaluated for small image regions by using linguistic variables and the fuzzy sets theory. The recognition of vehicles is performed by using simple statistics on the values of the image attributes registered

in a video sequence. Occupancy of detection zones is then determined by counting the recognised vehicles features in appropriate image regions. The algorithm was applied for gray-scale video sequences. However, it can be easily extended to colour images. Robustness of the proposed algorithm was verified through extensive testing in various situations. Experimental results reported in this paper proves the real-time performance of the introduced algorithm.

2 Image Attributes

The pixels intensities and their spatial differences are the low-level image characteristics that can be effectively utilised for the vehicle detection. As the grey-scale images are taken into consideration, the colour attribute was introduced to categorise pixels into three classes: black (with low intensities), grey (medium intensities) and white (high intensities). This colour attribute allows to recognise some features of objects that are visible in traffic scenes, e. g.: black regions of image that usually correspond with bottoms of vehicles in daylight conditions, vehicles headlights having white colour, especially in night images and regions of grey colour which is typical of the road pavement.

The spatial differences of intensities allow us to evaluate contrast between the adjacent image regions. The fact that a high contrast correspond with sharp edges of a vehicle body, is particularly important for the task of vehicle detection. Therefore, the proposed method uses contrast measure as an relevant image attribute. The applied contrast description of two neighbouring image regions X and Y is based on distinction of three cases: (1) X is darker than Y, (2) X and Y have similar intensities, (3) X is brighter than Y.

The local attributes of t-th image for coordinates (x, y) are determined taking into account 7×7 square neighbourhood centred on a pixel (x, y) . Set of attributes for the pixel (x, y) consists of one colour attribute C and four contrast attributes: UR, UL, LL and LR. The contrast attributes describe intensity differences between centre of the analysed image region and its corners: upper right, upper left, lower left and lower right corner respectively. Values of the image attributes are determined by using linguistic variables. Sets of admissible values (linguistic terms) for the introduced attributes are defined as follows: $V_C = \{black, grey, white\}$ and $V_{UR} = V_{UL} = V_{LL} = V_{LR} = \{darker, similar, brighter\}$.

Fig. 1 presents the membership functions of fuzzy sets that are attached to the linguistic terms. These functions map values of pixels intensity onto membership degrees of linguistic terms. Membership degrees are computed taking into account the mean values of pixels intensities $IM(x, y, t)$ determined for the image regions of size 3×3 . The membership functions of linguistic terms describing the colour of a pixel are depicted in fig. 1 a). Parameters b_0 , b_1 and w_1 , w_2 of these functions are calculated by means of histogram analysis performed on the input image. Fig. 1 b) shows membership functions defined for terms that are used to describe contrast between two adjacent 3×3 image regions. The first of them is centred on the pixel (x, y) and the second one is centred on the pixel $(x + dx, y + dy)$. The selection of dx and dy values depends on the type

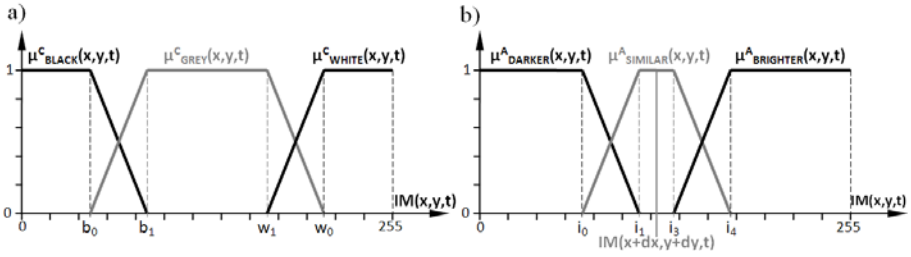


Fig. 1. Fuzzy sets definitions for linguistic variables: a) *C*, b) *UR*, *UL*, *LL* and *LR*

of attribute: $dx = 2, dy = -2$ for attribute *UR*, $dx = -2, dy = -2$ for *UL*, $dx = -2, dy = 2$ for *LL* and finally $dx = 2, dy = 2$ for *LR*. Parameters $i_0 \dots i_4$ of the membership functions are computed by a simple addition of predetermined offsets to the intensity value $IM(x + dx, y + dy, t)$.

As a result of a particular attribute *A* evaluation for the pixel (x, y) in *t*-th image a triple of membership degrees is computed:

$$A(x, y, t) = [\mu_{a_1}^A(x, y, t), \mu_{a_2}^A(x, y, t), \mu_{a_3}^A(x, y, t)], \tag{1}$$

where μ_a^A is the membership function of term *a* defined for attribute *A*, $a \in V_A, V_A = \{a_1, a_2, a_3\}$ is the set of values (terms) for attribute *A* and $A \in \{C, UR, UL, LL, LR\}$.

3 Vehicles Features

As it was discussed in the previous section, the values of image attributes are represented by triples of membership degrees. The proposed algorithm classifies each value of image attribute as a feature of vehicles, a feature of background or an unknown feature. The classification procedure is based on information regarding number of occurrences of particular attribute values in images that were previously analysed. It was assumed that the background features appear more frequently in the sequence of images than vehicles features do. Thus, a given attribute value can be recognised as a feature of vehicles if it has the low number of occurrences. When the occurrences number for a given value is high, it can be considered as a background feature. The attribute value will be categorised as an unknown feature if its number of occurrences is medium.

The accumulators arrays were applied in the introduced method to collect the occurrence information for respective values of image attributes. This information is necessary for the recognition of vehicles features in a video sequence. The accumulators arrays are updated after attributes evaluation for each image in the video sequence. Let $AC_a^A(x, y, t)$ denotes an accumulator that corresponds with value (term) *a* of attribute *A* evaluated for pixel (x, y) in *t*-th image. The count in the accumulator is incremented, if the value of attribute *A* for the analysed pixel conforms to the linguistic term *a*. This rule can be expressed by using following equation:

$$AC_a^A(x, y, t) = AC_a^A(x, y, t - 1) + \mu_a^A(x, y, t). \tag{2}$$

The accumulator count is decremented in the opposite situation, i. e. when the current value of attribute A is not a :

$$AC_a^A(x, y, t) = AC_a^A(x, y, t - 1) - \mu_{\bar{a}}^A(x, y, t). \tag{3}$$

When the standard definition of fuzzy set complement is taken into consideration, the membership degree of \bar{a} is calculated as follows:

$$\mu_{\bar{a}}^A(x, y, t) = 1 - \mu_a^A(x, y, t) \tag{4}$$

Therefore, equations (3) and (4) can be merged to give one formula for the accumulator update operation:

$$AC_a^A(x, y, t) = AC_a^A(x, y, t - 1) + 2\mu_a^A(x, y, t) - 1. \tag{5}$$

At the beginning, when the video sequence processing starts, there is no information available on the occurrences of particular attributes values. The accumulators are initialised at zero: $AC_a^A(x, y, 0) = 0$. A low absolute value of the count in an accumulator correspond with insufficient information, which does not allow to categorise the image features unambiguously. The features classification is possible only when the absolute value of the counts in accumulators is appropriately high. Moreover, the vehicle features are recognised if relevant accumulators counts are negative and background features are recognised in case of positive counts.

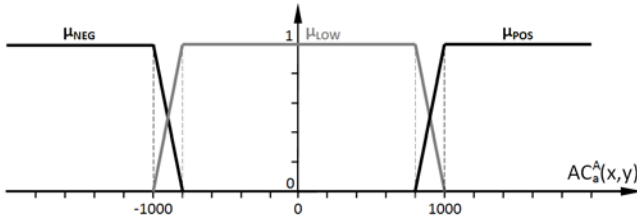


Fig. 2. Fuzzy sets definitions for linguistic terms: negative, low and positive

The terms used above to describe accumulators counts (low, negative, positive) have their own membership functions (fig. 2). The introduced membership functions are used to categorise image features. Fuzzy reasoning method with product t-norm and maximum t-conorm was applied to perform the features classification task. For an recorded image attribute $A(x, y, t)$ the degree of its membership in the vehicle features class (VF) is calculated as follows:

$$\mu_{VF}^A(x, y, t) = \max_a \{ \mu_{neg} [AC_a^A(x, y, t) \cdot \mu_a^A(x, y, t)] \}. \tag{6}$$

When the classes of background features (BF) or unknown features (UF) are taken into consideration, the membership degrees can be similarly computed. The only difference is in use of μ_{pos} or μ_{low} functions instead of μ_{neg} for BF and UF classes respectively.

4 Occupancy of a Detection Zone

The task of the proposed algorithm is to provide information on an occupancy of each detection zone. The occupancy $O_D(t)$ is a binary property of the detection zone D , determined for t -th image in a video sequence. The values of occupancy have the following interpretation: $O_D(t) = 1$ indicates that there is a vehicle occupying zone D in t -th image and $O_D(t) = 0$ denotes that the detection zone is empty. Procedure of the occupancy determination is based on vehicle features counting in detection zones. The underlying concept is that the number of vehicle features is high when the detection zone is occupied. The number of vehicle features recognised in a detection zone D for t -th image is calculated according to the following formula:

$$S_D(t) = \sum_{(x,y) \in D} \sum_A \mu_{VF}^A(x, y, t). \tag{7}$$

In the next step the resulting quantity is thresholded to determine a binary value of occupancy. The hysteresis thresholding method is applied here with two threshold values. The high threshold is used to formulate a condition of detection zone activation (switching occupancy to 1). The low threshold is taken into account when the detection zone is deactivated (occupancy changes from 1 to 0). Let $T_D^H(t)$ and $T_D^L(t)$ denote high and low threshold for detection zone D in the t -th image respectively. The occupancy is determined as follows:

$$O_D(t) = \begin{cases} 1, & S_D(t) \geq T_D^H(t) \\ 0, & S_D(t) \leq T_D^L(t) \\ O_D(t-1), & \text{else} \end{cases} \tag{8}$$

Appropriate determination of the thresholds values is crucial for the effective vehicle detection. Constant thresholds cannot ensure a good detection performance because significant variations of $S_D(t)$ range are often encountered for traffic video sequences. The variations are caused by the changes of ambient lighting conditions, weather phenomena, camera vibrations, accuracy limitations of the vehicles features recognition, etc. For the discussed algorithm an adaptive method of threshold determination was introduced to deal with the aforementioned problems. The thresholds values are determined for each image in the analysed video sequence. The following computations are performed to calculate them:

$$T_D^H = \max\{p_D S_D^{MAX}(t) + (1 - p_D) S_D^{MIN}(t), 100p_D\}, \quad T_D^L(t) = \alpha T_D^H(t), \tag{9}$$

where $p_D \in [0; 1]$ is the configuration parameter, which allows to adjust „sensitivity” of the detection zone D . The current range of $S_D(t)$ values is determined by the interval $[S_D^{MIN}(t); S_D^{MAX}(t)]$. The endpoints of this interval are calculated according to:

$$S_D^{MIN}(t) = \begin{cases} S_D^{MIN}(t-1) + \beta^{MIN}, & S_D(t) > S_D^{MIN}(t-1) \\ S_D(t), & S_D(t) \leq S_D^{MIN}(t-1) \end{cases},$$

$$S_D^{MAX}(t) = \begin{cases} S_D^{MAX}(t-1) - \beta^{MAX}, & S_D(t) < S_D^{MAX}(t-1) \\ S_D(t), & S_D(t) \geq S_D^{MAX}(t-1) \end{cases}. \tag{10}$$

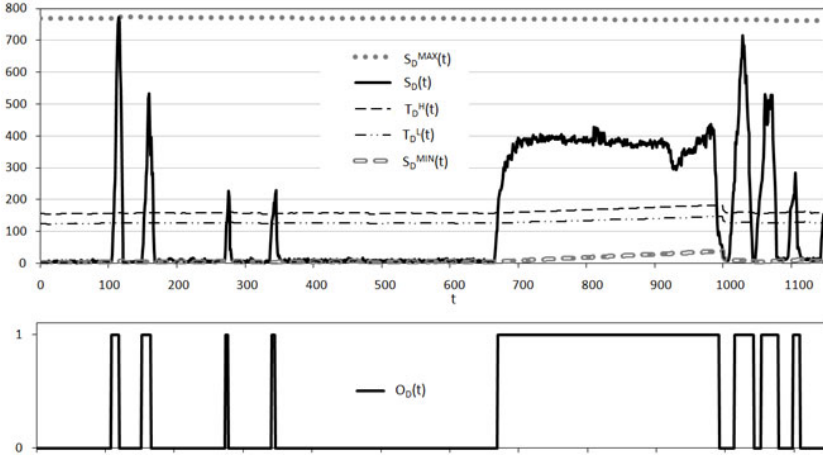


Fig. 3. An example of occupancy determination for a single detection zone

It was experimentally verified that for $\alpha = 0.8$, $\beta^{MIN} = 0.1$ and $\beta^{MAX} = 0.01$ the algorithm gives the correct results. In most cases the sensitivity parameter p_D was set to 0.2. Higher values need to be used when visible edges exist in the background of a detection zone.

Fig. 3 illustrates the occupancy determination procedure for a single detection zone. A sequence of 1150 images was analysed in this example (46 seconds) and eight vehicles were detected. The registered numbers of vehicle features along with thresholds values are presented in the upper chart. The lower plot represents binary values of the detection zone occupancy.

The basic method of occupancy determination, using thresholds defined above, was experimentally tested and some modifications were formulated to improve the algorithm performance. The first modification was motivated by a simple observation that the occupancy value for a detection zone may be changed only when a vehicle is entering into the zone or it is leaving the zone. In both cases the motion of the vehicle can be recognised by video sequence analysis. Thus, an additional condition was introduced for the occupancy determination, which takes into account the results of the motion detection. This modification was found to allow for errors reduction of the vehicle detection, especially when vehicles stops in detection areas for long time-periods. The simple motion detection method was implemented, based on subtraction of successive images in the video sequence.

The further improvement of the vehicle detection was achieved by occupancy-dependent update of the accumulators. The aim of this modification is to prevent vehicles features registering as the features of image background within occupied detection zones. According to this modification, when the detection zone is occupied, the accumulators update is executed in a different way than that described in section 3. For each pixel (x, y) in the occupied detection zone D the accumulator $AC_a^A(x, y, t)$ is updated as defined by (5) only if the current count of

this accumulator is not low. The update is skipped in an opposite situation. The term „slow” in the above condition is interpreted by using membership function of the fuzzy set presented in fig. 2.

5 Experimental Results

The proposed algorithm was tested on video sequences captured at several cross-roads in various lighting and weather conditions. The summarised length of all tested sequences was about 20 hours. They include day and night time scenes of the congested as well as free traffic flow. The test conditions were complicated with rain, snow, slight camera vibrations and significant light reflections.

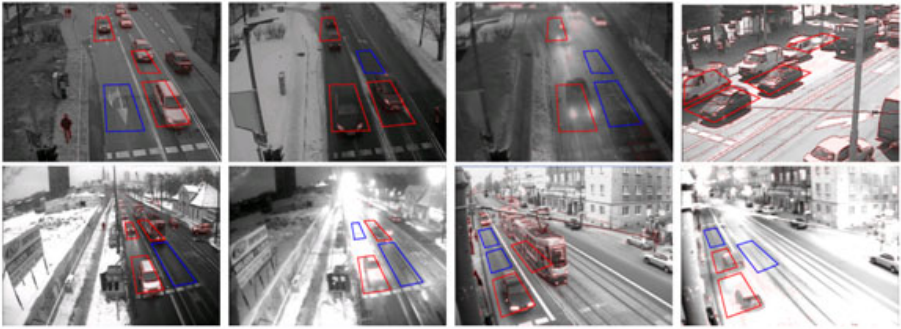


Fig. 4. Sample images of video sequences used for the experiments

Fig. 4 shows some examples of the analysed video with displayed boundaries of detection zones. More examples of the vehicle detection results are available at www.science-tech.pl. The size of experimental images was 768 x 512 pixels with 8-bit greyscale. A non-optimized implementation of the algorithm achieved processing speed of 30 fps on a Windows PC with Intel Core 2 2,5 GHz. Thus, it fulfils the real-time performance requirement for the video streams operating at 25 fps frame rate.

The accuracy of the vehicle detection was evaluated by the results verification for each time interval of 2 seconds. Two categories of errors were taken into account: a false negative detection was registered if a vehicle was present in a detection zone and the occupancy remained zero during all the time interval. When the detection zone was empty and the occupancy value was one for the time interval then the false positive detection was recognised. The error rate of vehicle detection was lower than 1% for sequences recorded by the cameras installed directly above traffic lanes, in good weather during the day time, when the lighting conditions were stable. This result was obtained also for congested traffic, when vehicles queues were observed. In the worst case, for a night time video sequence with very poor lighting and reflections from the pavement, the error rate rose to 11%

6 Conclusions

The experimental results demonstrate that the proposed algorithm is feasible and promising for the applications in the vision-based vehicle detection. It fulfils the real-time requirement and provides the robust vehicle detection under the complex conditions like lighting transitions, traffic congestion, vehicles headlight impact, etc. The accuracy of detection can be further enhanced by extending the algorithm to colour images. A hardware (FPGA) implementation of this algorithm is possible due to its low computational complexity. It will enable the vision sensors to be more cost-effective in the construction of the vehicle detection systems for road traffic control purposes.

References

1. Kim, Z., Malik, J.: Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In: IEEE Int. Conf. Comp. Vision, pp. 524–531 (2003)
2. Liu, A., Yang, Z.: Video Vehicle Detection Algorithm through Spatio-Temporal Slices Processing. In: IEEE Int. Conf. Mechatronic Embedded Systems, pp. 1–5 (2006)
3. Płaczek, B., Staniek, M.: Model Based Vehicle Extraction and Tracking for Road Traffic Control. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems, Advances in Soft Computing, vol. 2, pp. 844–851. Springer, Heidelberg (2007)
4. Płaczek, B.: Vehicles Recognition Using Fuzzy Descriptors of Image Segments. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems, Advances in Soft Computing, vol. 3, pp. 79–86. Springer, Heidelberg (2009)
5. Wang, Y., Ye, G.: Joint random fields for moving vehicle detection. In: British Machine Vision Conf., vol. 1, pp. 13–22 (2008)
6. Xu, S., Zhao, Y., Yu, C., Shen, L.: Vehicle Detection Algorithm Based on Shadow Feature. In: IEEE Int. Colloquium ISECS, vol. 1, pp. 105–109 (2008)
7. Yin, M., Zhang, H., Meng, H., Wang, X.: An HMM-Based Algorithm for Vehicle Detection in Congested Traffic Situations. In: IEEE ITS Conf., pp. 736–741 (2007)
8. Yue, Y.: A Traffic-Flow Parameters Evaluation Approach Based on Urban Road Video. Int. J. of Intelligent Engineering and Systems 2(1), 33–39 (2009)
9. Zhang, G., Avery, R.P., Wang, Y.: Video-based Vehicle Detection and Classification System for Real-time Traffic Data Collection Using Uncalibrated Video Cameras. TRB Transportation Research Record, no. 1993, pp. 138–147 (2007)
10. Zhang, W., Wu, J., Yin, H.: Moving vehicles detection based on adaptive motion histogram. Digital Signal Processing, 1–13 (2009), doi:10.1016/j.dsp.2009.10.006

Sequential Reduction Algorithm for Nearest Neighbor Rule

Marcin Raniszewski

University of Łódź, Faculty of Physics and Applied Informatics, Łódź, Poland
`marcin.raniszewski@uni.lodz.pl`

Abstract. An effective training set reduction is one of the main problems in constructing fast 1-NN classifiers. A reduced set should be significantly smaller and ought to result in a similar fraction of correct classifications as a complete training set. In this paper a sequential reduction algorithm for nearest neighbor rule is described. The proposed method is based on heuristic idea of sequential adding and eliminating samples. The performance of the described algorithm is evaluated and compared with three other well-known reduction algorithms based on heuristic ideas, on four real datasets extracted from images.

1 Introduction

The 1-NN rule is a very popular, effective and simple method of sample classification in Pattern Recognition (see [4] and [11]). For sufficiently large training set, 1-NN classification error is never beyond the double classification error of the Bayesian classifier (the smallest, theoretically possible error rate) (see [4]). The 1-NN rule offers usually worse performance than the k -NN rules with $k > 1$. On the other hand, the 1-NN rule does not require a training phase and operates much faster than the k -NN rules with $k > 1$.

The original k -NN classifiers have serious disadvantage: they operate with all samples from training set. To classify a sample x distances between the sample x and all training set samples are counted and the k closest samples are chosen. If a great number of samples has to be classified, the classification phase can last very long, what may be not satisfactory, mainly in applications of image analysis (classification of pixels).

The one of well-known solutions of the above problem is a reduction of a training set. A reduced training set should fulfil two conditions: provide similar fraction of correct classifications as that, obtained with a complete training set and contain the possibly smallest amount of samples.

Among many reduction methods there are a lot of techniques based on heuristic ideas. Some of them, like MC1, RMHC-P (see [10]), GA (see [7], [8] and [9]) and TS (see [3]), are more precisely described in Sect. 2. In spite of good results they provide (high fraction of correct classifications and strong reduction level), these algorithms are not parameter-free (the correct values of parameters should be established during a validation phase, what can be difficult due to the

randomness of these methods). Moreover, some of them (GA, TS) can last very long with datasets consisting thousands of samples. The aim of the presented research was the creation of simple, parameter-free, random algorithm, much faster than GA and TS, providing similarly high fraction of correct classifications and reduction level as described in Sect. 2 well-known reduction procedures.

2 Well-Known Reduction Procedures Based on Heuristic Ideas

Skalak in [10] proposed two heuristics: MC1 and RMHC-P. The former uses the *Monte Carlo* method, while the latter: *Random Mutation Hill Climbing* procedure. Both algorithms have three parameters: k - the number of nearest neighbors in k -NN rule, m - the desired number of samples in a reduced training set, n - the number of reduced set generations in MC1 and the number of replacements (called mutations) in reduced set in RMHC-P. Results of both algorithms are unexpectedly good: high classification quality and strong reduction simultaneously. Both algorithms proposed by Skalak are simple and very fast in comparison with other well-known heuristic methods.

Kuncheva in [7], [8] and [9] described heuristic based on Genetic Algorithms (GA), which rates as editing technique (editing technique's aim is to find the subset of original training set, which provide better classification quality; a reduction level is not important). Every subset of training set is represented as a binary string called "chromosome" (i -th bit is set to one if i -th sample is in reduced set and to zero, otherwise). Each bit in chromosome is initially set to one with a predefined probability, called the reduction rate. Crossover and mutation rates are predefined probabilities used in crossover and mutation operations in a reproduction phase: two offspring chromosomes are produced by every couple of parent chromosomes (each pair of corresponding bits of the parent chromosomes are swapped with crossover rate) and then each bit of each offspring chromosome alternates (mutates) with the mutation rate. In [8], Kuncheva presented GA based on new fitness function with a penalty term (the number of samples in reduced set) as a reduction method and this implementation is considered in the paper. The main disadvantages of GA are: seven parameters and a long training phase for datasets with above 5000 samples with dozens of attributes. GA provide good classification quality and a satisfactory reduction level.

In [3] authors proposed heuristic based on *Tabu Search* method (TS). TS operates on subsets represented by binary strings, equivalently to GA's chromosomes. Subsets that differ from actual subset S by just one element create a new neighborhood of S . Cerveron and Ferri consider the same objective function as Kuncheva's fitness function (with a penalty term) and two different methods of creating a initial subset: condensed or constructive. The former uses CNN (see [5]) to obtain a consistent reduced set, the latter uses TS with disabled sample deletion (in creating a neighborhood of actual solution) starting from a subset of randomly selected samples, a single sample from each class, until a consistent set is obtained. TS results are alike to Skalak's algorithms. TS's training phase

lasts very long for thousands of samples with dozens of attributes. TS has three parameters.

3 Sequential Reduction Algorithm - The Proposed Reduction Technique

The proposed reduction algorithm, called *Sequential Reduction Algorithm* (SeqRA) is based on the heuristic idea of sequential adding and eliminating samples. The procedure lasts until classification performance of created subset is locally the highest: each sample no matter added or eliminated decreases a number of correctly classified samples from complete training set.

Sequential Reduction Algorithm consists of the following steps (X_{red} denotes reduced training set - initially empty, X - complete training set, $f(X_{red})$ - fraction of samples from X correctly classified using 1-NN rule operating with X_{red} and c denotes the number of classes in X):

1. Select randomly c samples from different classes and add them to X_{red} .
2. Compute $f(X_{red})$. If $f(X_{red}) = 100\%$ go to step 7, otherwise, let $f_{max} = f(X_{red})$.
3. Mark all samples as "unchecked".
4. For $j = 1..c$:
 - (a) Add random "unchecked" j -class sample from $X \setminus X_{red}$ to X_{red} .
 - (b) Compute $f(X_{red})$. If $f(X_{red}) = 100\%$ go to step 7, if $f(X_{red}) \leq f_{max}$ eliminate from X_{red} the sample added in the step 4(a) and mark it as "checked", otherwise, let $f_{max} = f(X_{red})$ and mark all samples as "unchecked".
5. For $j = 1..c$:
 - (a) Eliminate random "unchecked" j -class sample from X_{red} .
 - (b) Compute $f(X_{red})$. If $f(X_{red}) = 100\%$ go to step 7, if $f(X_{red}) \leq f_{max}$ add again to X_{red} the sample eliminated in the step 5(a) and mark it as "checked", otherwise, let $f_{max} = f(X_{red})$ and mark all samples as "unchecked".
6. Repeat the steps 4–5 until all samples will be "checked".
7. X_{red} is a reduced reference set.

If, in iteration j in step 4 or 5 there are no "unchecked" samples, the iteration is skipped.

After each adding or eliminating sample, new X_{red} offers better classification quality. Thus, the algorithm is finite.

The SeqRA is a parameter-free and random method (each time the algorithm is run for training set X , it creates the different X_{red}).

4 Datasets Used in Experiments

Four real datasets were used to tests (see [\[1\]](#)):

- Optical Recognition of Handwritten Digits Dataset (OPTDIGITS) – 10 classes, 64 attributes, 5620 samples (training and test set were united). Description of the dataset: "We used preprocessing programs made available by NIST to extract normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into nonoverlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range 0..16. This reduces dimensionality and gives invariance to small distortions",
- Statlog (Landsat Satellite) Dataset (SAT) – 7 classes (one empty), 36 attributes, 4435 samples in the training set and 2000 samples in the test set. Description of the dataset: "One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. Each image contains 2340 x 3380 such pixels. The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label of the central pixel. The number is a code for the following classes: (1) red soil, (2) cotton crop, (3) grey soil, (4) damp grey soil, (5) soil with vegetation stubble, (6) mixture class (all types present), (7) very damp grey soil. There are no examples with class 6 in this dataset. In each line of data the four spectral values for the top-left pixel are given first followed by the four spectral values for the top-middle pixel and then those for the top-right pixel, and so on with the pixels read out in sequence left-to-right and top-to-bottom",
- Statlog (Image Segmentation) Dataset (SEGMENTATION) – 7 classes, 19 attributes, 2310 samples (training and test set were united). Description of the dataset: "The instances were drawn randomly from a database of 7 outdoor images (brickface, sky, foliage, cement, window, path, grass). The images were handsegmented to create a classification for every pixel. Each instance is a 3x3 region",
- Breast Cancer Wisconsin (Diagnostic) Dataset (WDBC) – 2 classes, 30 attributes, 569 samples. Description of the dataset: "Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image".

5 Experimental Results

Stratified ten-fold cross-validation was used for experiment (see [6]), except of reduction of SAT datasets (authors do not agree to cross-validate this dataset, hence the training set was reduced and test once).

The 1-NN rule was used to estimate the classification quality.

All tests were made on Intel Core 2 Duo, T7250 2.00 Ghz processor with 2 GB RAM.

Datasets were reduced by four heuristic methods: RMHC-P, GA, TS and proposed SeqRA. In GA the number of iterations was set to 200 for SAT, SEGMENTATION and WDBC datasets and to 50 for OPTDIGITS dataset due to very long training phase, the reduction rate to 0.1, number of chromosomes to 20, the crossover rate to 0.5, the mutation rate to 0.025 and the number of nearest neighbors k to 1. Implementation of GA is equivalent to that proposed in [2] with a fitness function with penalty term (cardinality of reduced set weighted by coefficient $\alpha = 0.01$; the higher is α , the higher gets penalty). TS parameters are set to values proposed in [3] with one difference: the condensed initialization was used for OPTDIGITS and SAT datasets due to very long phase of constructive initialization for those training sets. In RMHC-P: $k = 1$, m is equal to reduced set size obtained by the SeqRA (for better comparison of the algorithms) and n was set to 2500 for OPTDIGITS, 2000 for SAT 1500 for SEGMENTATION and 600 for WDBC dataset.

All algorithms were implemented in Java.

6 Discussion

Experimental results (average fractions of correct classifications and reduction levels) are visualized in Fig. 1.

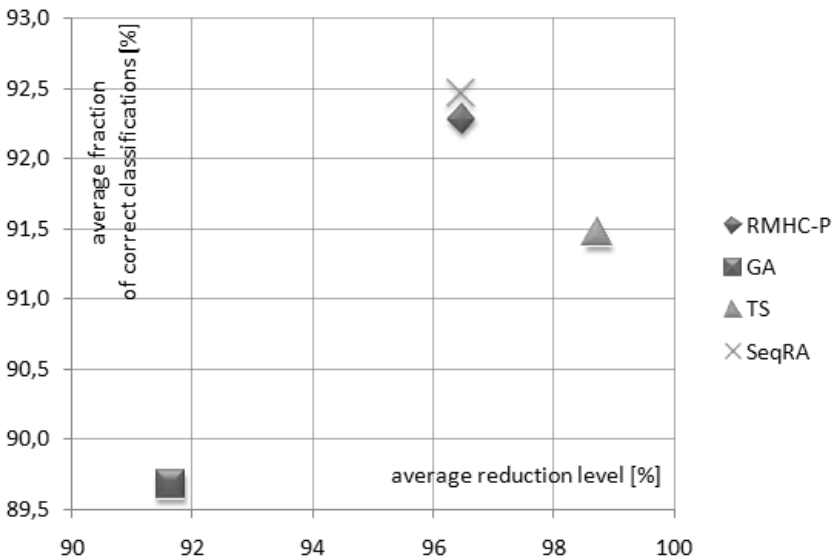


Fig. 1. Experimental results

Table 1. The test results: reduction levels (fractions of discarded samples in percentages). The number under mean value for specific dataset is a standard deviation (except of SAT dataset). The last row "avg" presents average values in columns from all datasets.

	RMHC-P	GA	TS	SeqRA
OPTDIGITS	96.34 0.27	90.67 0.16	99.33 0.03	96.34 0.27
SAT	96.34	90.64	99.50	96.34
SEGMENTATION	94.61 1.25	91.24 0.42	97.61 0.10	94.61 1.25
WDBC	98.61 0.60	94.06 0.41	98.52 0.23	98.61 0.60
avg	96.48 0.71	91.65 0.33	98.74 0.12	96.48 0.71

Table 2. The test results: classification qualities (in percentages). The number under mean value for specific dataset is a standard deviation (except of SAT dataset). Column called "compl." means the classification quality on complete training set using 1-NN rule, the last row "avg" presents average values in columns from all datasets.

	compl.	RMHC-P	GA	TS	SeqRA
OPTDIGITS	98.79 0.32	96.78 0.56	96.00 0.61	94.77 1.26	97.17 1.04
SAT	89.40	89.05	86.60	87.25	89.75
SEGMENTATION	96.36 1.00	90.87 2.58	85.06 2.84	90.74 2.37	91.04 2.88
WDBC	91.19 4.16	92.42 4.33	91.03 4.63	93.14 2.47	91.89 3.21
avg	93.93 1.83	92.28 2.49	89.67 2.69	91.47 2.03	92.46 2.38

TS, RMHC-P and SeqRA offer similar and very strong reduction level (average from 96.5% to 98.7% – see Table 1) and slightly worse classification qualities than that obtained on complete training sets (see Table 2). GA results in the lowest, but still satisfactory strong reduction level (approx. 91.7% – see Table 1) and classification quality (see Table 2).

All the heuristics offer very good results, especially TS method, which perfectly reduced training sets (almost 99% of reduction) and SeqRA and RMHC-P,

which provide (except of SEGMENTATION dataset) almost identical fraction of correct classifications as that, obtained with a complete training set.

The main problems connected with heuristics are the number of parameters and long duration of training phase for datasets with thousands of samples. The different parameter settings in GA, RMHC-P and TS can lead to slightly better or slightly worse results. Correct setting of parameters requires more validation and more validations require more time. Duration of training phase of GA and TS is very long for middle dataset (like OPTDIGITS - see Table 3). For large datasets duration of training phase of GA and TS can grow to serious limitation. Therefore, the use of SeqRA, which is parameter-free, has acceptable duration of training phase for middle datasets (approx. 2 min. – see Table 3) and offers similar results, should be taken into consideration.

Table 3. The test results: duration of training phase (reduction of complete training sets). In GA the number of iterations was set to 200. In TS condensed initialization was used.

	RMHC-P	GA	TS	SeqRA
OPTDIGITS	~0.5 min.	~6.5 hrs	~7 hrs	~2 min.
SAT	~8 s	~2 hrs	~2.5 hrs	~1 min.
SEGMENTATION	~2 s	~20 min.	~6 min.	~5 s
WDBC	~0.5 s	~1 min.	~1 min.	~1 s

7 Conclusions

Presented Sequential Reduction Algorithm is heuristic based on the idea of sequential adding and eliminating the samples. The algorithm has the following advantages:

- high fraction of correct classification of reduced training set (similar to other well-known heuristics),
- satisfactory high reduction level of training set (similar to other well-known heuristics),
- very low duration of training phase in comparison with other well-known heuristics like GA and TS,
- lack of parameters, in opposite to other well-known heuristics.

However, the following disadvantages (common for all considered heuristics) must be also taken in the consideration:

- problem with unacceptable long training phase for large datasets,
- solution is unequivocal due to randomness in choosing samples.

Acknowledgements

The author is a scholarship holder of project entitled „Innovative education...” supported by European Social Fund.

References

1. Clarke, F., Ekeland, I., Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: an experimental study. *International Journal of Intelligent Systems* 16(12), 1445–1473 (2001)
3. Cerveron, V., Ferri, F.J.: Another move towards the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule. *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics* 31(3), 408–413 (2001)
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. John Wiley & Sons, Inc., Chichester (2001)
5. Hart, P.E.: The condensed nearest neighbor rule. *IEEE Transactions on Information Theory* IT-14(3), 515–516 (1968)
6. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proc. 14th Int. Joint Conf. Artificial Intelligence*, pp. 338–345 (1995)
7. Kuncheva, L.I.: Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters* 16, 809–814 (1995)
8. Kuncheva, L.I.: Fitness functions in editing k-NN reference set by genetic algorithms. *Pattern Recognition* 30(6), 1041–1049 (1997)
9. Kuncheva, L.I., Bezdek, J.C.: Nearest prototype classification: clustering, genetic algorithms, or random search? *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 28(1), 160–164 (1998)
10. Skalak, D.B.: Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: *11th International Conference on Machine Learning*, New Brunswick, NJ, USA, pp. 293–301 (1994)
11. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*, 3rd edn. Academic Press/Elsevier, USA (2006)

GPU-Supported Object Tracking Using Adaptive Appearance Models and Particle Swarm Optimization

Bogusław Rymut* and Bogdan Kwolek

Rzeszów University of Technology
W. Pola 2, 35-959 Rzeszów, Poland
bkwolek@prz.rzeszow.pl
<http://www.prz.edu.pl>

Abstract. This paper demonstrates how CUDA-capable Graphics Processor Unit can be effectively used to accelerate a tracking algorithm based on adaptive appearance models. The object tracking is achieved by particle swarm optimization algorithm. Experimental results show that the GPU implementation of the algorithm exhibits a more than 40-fold speed-up over the CPU implementation.

1 Introduction

While the central processing unit (CPU) is a general purpose microprocessor, which carries out the instructions of a computer program and is capable of processing a wide range of instructions, a graphics processor unit (GPU) is a dedicated microprocessor for the performing of graphical operations of the program. Modern GPUs are designed to operate in a SIMD fashion, which is a natural computational paradigm for graphical tasks. Recent research demonstrates that they are capable of accelerating a much broader scope of applications than the real-time rendering applications for which they were originally designed. GPUs offer potential for considerable increase in computation speed in applications that are data parallel.

Object tracking is an important problem in computer vision. It is a prerequisite for analyzing and understanding visual data, and has been an active research topic in the computer vision community over the last two decades. The goal of the object tracking is to automatically find the same object in an adjacent frame from an image sequence once it is initialized. Tracking algorithms are now employed in a wide variety of domains, such as robotics, human-computer-communication, vehicular traffic and surveillance. The challenge is to track the object irrespective of scale, rotation, perspective projection, occlusions, changes of appearance and illumination. Therefore, reliable vision-based object tracking is typically time-consuming process. However, it should be fast enough to maintain transparent interaction with the user.

* B. Rymut is currently a student, doing his MSc thesis on GPU-based object tracking

Bayesian filtering techniques are often employed to achieve reliable tracking. For example, the Kalman filter has been used to track object in [1]. Unfortunately, object tracking in real-world environment rarely satisfies Kalman filter’s requirements. Particle filtering [2] is superior to Kalman filtering without being constrained to linear models and Gaussian observations. However, particle filter (PF) being a sequential Monte Carlo method is time-consuming tracking technique. One of the major drawbacks of particle filters is that a huge number of particles are usually required for accurate estimation of state variables lying in a high dimensional space.

One way to achieve object tracking is searching for the best match of the pre-defined object model in the image. Recently, particle swarm optimization (PSO), a population based stochastic optimization technique has received considerable attention. Unlike the independent particles in the PF, the particles in a PSO interact locally with one another and with their environment in the course of searching for the best solution. Each particle in the swarm represents a candidate solution to the optimization problem. The most time consuming operation in PSO-based object tracking is evaluation of the fitness function. Since multiple candidate solutions are evaluated in each iteration, PSO-based tracking algorithms are computationally demanding for real-time applications.

Adaptive appearance models have acknowledged their great usefulness in visual tracking. In [3], the appearance model is based on phase information derived from the image intensity. Similar to this work, the appearance models that are utilized in [4][5] consist of three components, W, S, F , where the W component models the two-frame variations, the S component characterizes temporally stable images, and the F component is a fixed template of the target to prevent the model from drifting away. The algorithms mentioned above produce good tracking results, but are quite time-consuming. This motivated us to develop a GPU implementation of the tracking using particle swarm optimization with adaptive appearance models. Since in adaptive appearance model based tracking the objects are represented as 2D arrays of pixels data, our algorithm takes the advantage of SIMD architecture effectively.

2 Visual Appearance Modeling Using Adaptive Models

Our intensity-based appearance model consists of three components, namely, the W -component expressing the two-frame variations, the S -component characterizing the stable structure within all previous observations and F component representing a fixed initial template. The model $A_t = \{W_t, S_t, F_t\}$ represents the appearances existing in all observations up to time $t - 1$. It is a mixture of Gaussians [3] with centers $\{\mu_{k,t} \mid k = w, s, f\}$, their corresponding variances $\{\sigma_{k,t}^2 \mid k = w, s, f\}$ and mixing probabilities $\{m_{k,t} \mid k = w, s, f\}$.

Let $I(p, t)$ denote the brightness value at the position $p = (x, y)$ in an image \mathcal{I} that was acquired in time t . Let \mathcal{R} be a set of J locations $\{p(j) \mid j = 1, 2, \dots, J\}$ defining a template. $Y_t(\mathcal{R})$ is a vector of the brightness values at locations $p(j)$ in the template. The observation model has the following form:

$$p(z_t|x_t) = \prod_{j=1}^J \sum_{k=w,s,f} \frac{m_{k,t}(j)}{\sqrt{2\pi\sigma_{k,t}^2(j)}} \exp \left[-\frac{1}{2} \left(\frac{Y_t(j) - \mu_{k,t}(j)}{\sigma_{k,t}(j)} \right)^2 \right] \quad (1)$$

where z_t is the observation corresponding to template parameterization x_t . In the object likelihood function we utilize a recursively updated appearance model, which depicts stable structures seen so far, two-frame variations as well as initial object appearance. Owing to normalization by subtracting the mean and dividing by standard deviation, the template becomes invariant to global illumination changes.

The update of the current appearance model A_t to A_{t+1} is done using the Expectation Maximization (EM) algorithm [6]. For a template $\hat{Y}_t(\mathcal{R})$, which was obtained from the image \mathcal{I} using the estimated parameterization \hat{x}_t , we evaluate the posterior contribution probabilities as follows:

$$o_{k,t}(j) = \frac{m_{k,t}(j)}{\sqrt{2\pi\sigma_{k,t}^2(j)}} \exp \left[-\frac{1}{2} \left(\frac{\hat{Y}_t(j) - \mu_{k,t}(j)}{\sigma_{k,t}(j)} \right)^2 \right] \quad (2)$$

where $k = w, s, f$ and $j = 1, 2, \dots, J$. The posterior contribution probabilities (with $\sum_k o_{k,t}(j) = 1$) are utilized in updating the mixing probabilities in the following manner:

$$m_{k,t+1}(j) = \gamma o_{k,t}(j) + (1 - \gamma)m_{k,t}(j) \quad | \quad k = w, s, f \quad (3)$$

where γ is accommodation factor. Then, the first and the second-moment images are determined as follows:

$$M_{1,t+1}(j) = (1 - \gamma)M_{1,t}(j) + \gamma o_{s,t}(j)\hat{Y}_t(j) \quad (4a)$$

$$M_{2,t+1}(j) = (1 - \gamma)M_{2,t}(j) + \gamma o_{s,t}(j)\hat{Y}_t^2(j) \quad (4b)$$

In the last step the mixture centers and the variances are calculated as follows:

$$\mu_{s,t+1}(j) = \frac{M_{1,t+1}(j)}{m_{s,t+1}(j)}, \quad \sigma_{s,t+1}(j) = \sqrt{\frac{M_{2,t+1}(j)}{m_{s,t+1}(j)} - \mu_{s,t+1}^2(j)} \quad (5)$$

$$\mu_{w,t+1}(j) = \hat{Y}_t(j), \quad \sigma_{w,t+1}(j) = \sigma_{w,1}(j) \quad (6)$$

$$\mu_{f,t+1}(j) = \mu_{f,1}(j), \quad \sigma_{f,t+1}(j) = \sigma_{f,1}(j) \quad (7)$$

In order to initialize the model A_1 the initial moment images are set using the following formulas: $M_{1,1} = m_{s,1}Y_{t0}(\mathcal{R})$ and $M_{2,1} = m_{s,1}(\sigma_{s,1}^2 + Y_{t0}^2(\mathcal{R}))$.

3 PSO-Based Object Tracking

PSO is a population based algorithm that exploits a set of particles representing potential solutions of the optimization task [7]. The particles fly through the n -dimensional problem space with a velocity subject to both stochastic and deterministic update rules. The algorithm seeks for the global best solution through adjusting at each time step the location of each individual according to personal best and the global best positions of particles in the entire swarm. Each particle keeps the position $pbest$ in the problem space, which is associated with the best fitness it has achieved personally so far. Additionally, when a particle considers all the population as its topological neighbors, each particle employs $gbest$ location, which has been obtained so far by any particle in the swarm. The new positions are subsequently scored by a fitness function f . The velocity of each particle i is updated in accordance with the following equation:

$$v_j^{(i)} \leftarrow wv_j^{(i)} + c_1r_{1,j}(pbest_j^{(i)} - x_j^{(i)}) + c_2r_{2,j}(gbest_j - x_j^{(i)}) \quad (8)$$

where $v_j^{(i)}$ is the velocity in the j -th dimension of the i -th particle, c_1 , c_2 denote the acceleration coefficients, $r_{1,j}$ and $r_{2,j}$ are uniquely generated random numbers in the interval [0.0, 1.0]. The new position of a particle is calculated in the following manner:

$$x_j^{(i)} \leftarrow x_j^{(i)} + v_j^{(i)} \quad (9)$$

The local best position of each particle is updated as follows:

$$pbest^{(i)} \leftarrow \begin{cases} x^{(i)}, & \text{if } f(x^{(i)}) > f(pbest^{(i)}) \\ pbest^{(i)}, & \text{otherwise} \end{cases} \quad (10)$$

and the global best position $gbest$ is defined as:

$$gbest \leftarrow \arg \max_{pbest^{(i)}} \{f(pbest^{(i)})\} \quad (11)$$

The value of velocity $v^{(i)}$ should be restricted to the range $[-v_{max}, v_{max}]$ to prevent particles from moving out of the search range. In some optimization problems the local best version of PSO, where particles are influenced by the best position within their neighborhood, as well as their own past experience can give better results. While such a configuration of the PSO is generally slower in convergence than algorithm with $gbest$, it typically results in much better solutions and explores a larger part of the problem space.

At the beginning of the optimization the PSO initializes randomly locations as well as the velocities of the particles. Then the algorithm selects $pbest$ and $gbest$ values. Afterwards, equations (8)-(11) are called until maximum iterations or minimum error criteria is attained. After that, given $\hat{x}_t = gbest$ we calculate \hat{Y}_t , and then update of the object model using formulas (2)-(7).

In the simplest solution the object tracking can be realized as deterministic searching of window location whose content best matches a reference window

content. PSO allows us to avoid such time consuming exhaustive searching for the best match. It provides an optimal or sub-optimal match without the complete knowledge of the searching space. In PSO based tracking, at the beginning of each frame in the initialization stage, an initial position is assigned to each particle

$$x_t^{(i)} \leftarrow \mathcal{N}(gbest, \Sigma) \quad (12)$$

given the location $gbest$ that has been estimated in the previous frame $t - 1$. In the evaluation phase the fitness value of each particle is determined by a predefined observation model according to the following formula:

$$f(x_t^{(i)}) = p(z_t^{(i)} | x_t^{(i)}) \quad (13)$$

4 Programming of GPU

In this section we outline the architectural properties of G80 [8], which are the most relevant to our implementation. CUDATM is a new language and development environment developed by NVIDIA, allowing execution of programs with thousands of data-parallel threads on NVIDIA G80 class GPUs [9]. Such threads are extremely lightweight and almost no cost for creation and context switch is needed. In CUDA, programs are expressed as kernels and GPU is viewed as a device, see Fig. 1, which can carry out multiple concurrent threads. Each kernel consists of a collection of threads arranged into blocks. A thread block is a group of threads, which can cooperate jointly through efficiently sharing data via some fast shared memory, and synchronizing their execution to coordinate memory accesses. A kernel should have enough blocks to simultaneously utilize all the multiprocessors in a given GPU. Many thread blocks can be assigned to a single multiprocessor, which are executed concurrently in a time-sharing fashion to keep GPU as busy as possible.

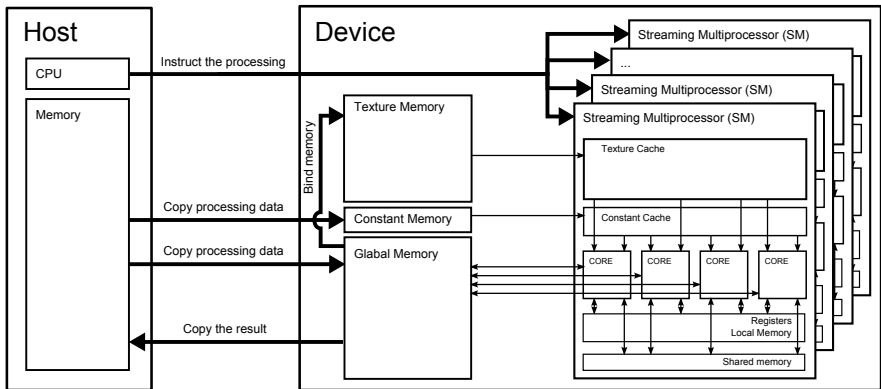


Fig. 1. G80 GPU architecture

As is illustrated in Fig. 11 both the host and the device maintain their own DRAM, referred to as host memory and device memory, respectively. On the device side the data structures can be mapped to the texture memory to take advantages of special hardware on the GPU, which supplies a small amount of on-chip caching and a little more efficient access to off-chip memory. The device memory is accessible by all multiprocessors. Each multiprocessor has a set of registers, shared memory, constant cache, and texture cache. Constant/texture cache are read-only and have faster access than shared memory. Because only threads within the same block can cooperate via shared memory and thread synchronization, the user must partition the computation into multiple blocks. The number of threads per block is restricted by the limited memory resources of a multiprocessor core. On current GPUs, a thread block may contain up to 512 threads. In general, the more threads per block, the better the performance because the hardware can hide memory latencies. High arithmetic intensity also hides the memory latency. Storing the data locally reduces the need to access off-chip memory, thereby improving the performance.

5 Implementation Details

Writing effective computer vision applications for GPU is not a trivial task. One should make careful decisions according the data layout, data exchange and synchronization to ensure that the program can take the full advantages of the available resources. In order to decompose the algorithm into GPU we should identify data-parallel portions of the program and separate them as CUDA kernels. The decomposition was done with regard to the main steps of the algorithm:

1. Assign each particle a random position in the problem hyperspace.
2. Evaluate the fitness function for each particle by applying (13).
3. For each particle i compare the particle's fitness value with its $f(pbest^{(i)})$.
If the current value is better than the value $f(pbest^{(i)})$, then set this value as the $f(pbest^{(i)})$ and the current particle's position $pbest^{(i)}$ as $x^{(i)}$.
4. Find the particle that has the best fitness value $gbest$.
5. Update the velocities and positions of all particles according to (8) and (9).
6. Given $gbest$, update of the object model using formulas (2) - (7).
7. Repeat steps 2 – 6 until maximum number of iterations is attained.

At the beginning of each frame we generate pseudo-random numbers using the Mersenne Twister [10] kernel provided by the CUDATM SDK. From uniform random numbers we generate a vector of normal random numbers using Box Mueller transform based on trigonometric functions [11] in order to initialize the particle's positions in the problem hyperspace. The positions are generated using equation (12). The initialization is executed in 32 blocks and each block consists of 128 threads, where each thread generates two random numbers. The evaluation of the fitness function is done in two separate kernels. The first kernel performs the normalization of the pixels in the template to the unit variance, whereas the second thread is executed after the first one and calculates the fitness

score. In both kernels each thread processes a single column of the template, and the parallel reduction technique [9] is used to obtain the final results. The results achieved by the first kernel are stored in the shared memory. Both kernels operate on textures. After the computation of the fitness score the calculation of the *pbest* locations and their corresponding fitness values takes place. Identical number of threads and blocks is used in this and in the initialization stage. Afterwards, the *gbest* value is calculated. Finally, the algorithm updates in parallel the velocities and the locations.

6 Experimental Results

The experiments were conducted on a PC with 1 GB RAM, Intel Core 2 Quad, 2.66 GHz processor with NVIDIA GeForce 9800 GT graphics card. The graphics card has 14 stream multiprocessors with 1.5 GHz, each with 8 cores, see Fig. 1. It is equipped with 1024 MB RAM, 64 KB constant memory and 16 KB common memory. We implemented the algorithm in CUDA and compared the runtimes with its counterpart that was implemented in C and executed on the CPU. The CPU code was compiled with Visual Studio 2005 with the SSE2 (Streaming SIMD Extensions 2) option and O2 optimization turned on. Table 1 shows the running times of the tracking algorithm both on CPU and GPU as well as the speed-up. The communication delays for copying images from CPU to GPU and vice versa have not been taken into account. The most time-consuming operation of the tracking algorithm is calculation of the fitness function (13). This operation amounts to 0.9 of the whole processing time.

Table 1. Tracking times [sec.] and speed-up obtained on CPU (Intel Core 2, 2.66 GHz) and a GPU (NVIDIA GeForce 9800 GT)

#particles	32	64	128	256
CPU(2.66 GHz)	30.6 ms	60.0 ms	117.9 ms	234.2 ms
GPU(GF 9800 GT)	1.4 ms	1.9 ms	3.4 ms	5.6 ms
CPU/GPU	22.4	31.5	38.8	41.5

In object tracking experiments we employed various number of particles, see Table 1. We can notice that for larger number of particles the speed-up of the GPU algorithm is larger. The tracking is done in three dimensional space, i.e. we track the location of the template as well as its scale. The scaling is achieved via bilinear interpolation, which is extension of the linear interpolation for interpolating functions of two variables on the regular grid of pixels. Figure 2 depicts some tracking results, which were obtained on color images 1. The first image

¹ Thanks Dr. Birchfield for this sequence, obtained from <http://robotics.stanford.edu/~birch/headtracker>

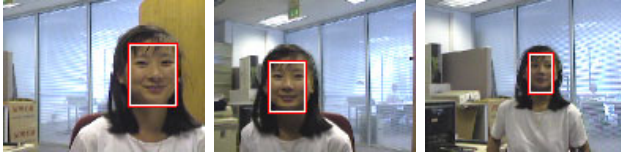


Fig. 2. GPU-based face tracking using adaptive appearance models

shown in Fig. 2 contains a face in the front of the background with colors that are similar to skin color. The size of the reference frame is 32×42 . The change of template size between successive frames is ± 1 pixel. The experimental results depicted on Fig. 2 were obtained using 32 particles.

7 Conclusions

In this paper, we have shown how the adaptive appearance based tracking algorithm can be accelerated significantly using programmable graphics hardware. The results showed that our algorithm using GPU is about 40 times faster than a CPU implementation. As a result the tracking algorithm runs at frame-rates exceeding 60 frames per second.

References

1. Weng, S., Kuo, C., Tu, S.: Video object tracking using adaptive Kalman filter. *J. Vis. Commun. Image Represent.* 17, 1190–1208 (2006)
2. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. *Int. J. of Computer Vision* 29, 5–28 (2006)
3. Jepson, A.D., Fleet, D.J., El-Maraghi, T.: Robust on-line appearance models for visual tracking. *IEEE Trans. on PAMI* 25, 1296–1311 (2003)
4. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: *IEEE Int. Conf. on CVPR*, pp. 1–8 (2008)
5. Kwolek, B.: Particle swarm optimization-based object tracking. *Fundamenta Informaticae* 95, 449–463 (2009)
6. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society. Series B* 39, 1–38 (1977)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942–1948. IEEE Press, Piscataway (1995)
8. Wasson, S.: Nvidia's GeForce 8800 graphics processor. Technical report, *PC Hardware Explored* (2006)
9. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable parallel programming with CUDA. *ACM Queue* 6, 40–53 (2008)
10. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation* 8, 3–30 (1998)
11. Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. *The Annals of Mathematical Statistics* 29, 610–611 (1958)

GFT: GPU Fast Triangulation of 3D Points

Jairo R. Sánchez, Hugo Álvarez, and Diego Borro

CEIT and Tecnun (University of Navarra), Manuel de Lardizábal 15
20018 San Sebastián Spain
{jrsanchez,halvarez,dborro}@ceit.es

Abstract. Traditionally triangulating 3D points from image features is a complex task that involves non-linear optimization techniques that are computationally very expensive. This work proposes an algorithm based on Monte Carlo simulations that fits well on the graphics hardware and can perform the triangulation in real time. Results are compared against the well known Levenberg-Marquardt using real video sequences, showing that it achieves the same precision but in much less time.

1 Introduction

Structure From Motion (SFM) problem is a subset of the Simultaneous Localisation And Mapping (SLAM) problem, where the objective is to solve the 3D structure of the scene and/or the camera motion relying on image data. This work addresses only the reconstruction part, and extends the work of Escudero et al. [1] that solves the motion estimation using similar principles.

Traditionally SFM is addressed using recursive estimations (probabilistic maps) or batch reconstructions (minimization techniques). The first option is fast but not as accurate as one would wish. The second option is very accurate but not suitable for real time operation.

Recursive estimators try to compute the reconstruction using Bayesian estimators, typically Extended Kalman Filters (EKF) [2]. We are interested in predicting the sequence of the state of our world $X_k = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ having a set of observations $Y_k = \{\mathbf{y}_1, \dots, \mathbf{y}_k\}$. The problem is formulated as the conditional probability of the current state given the entire set of observations: $P(\mathbf{x}_k | Y_k)$. The key is that the current state probability can be expanded using the Bayes's rule. Usually the state \mathbf{x}_k is composed by the camera pose and the 3D structure, and the observation \mathbf{y}_k is image measurements. Now the problem is how to approximate these probability densities. Authors like [3] use sequential Monte Carlo methods. The difference with the method proposed here is that they only take into account the pose information of the last frame (Markov sequence).

Global minimization is used when real time operation is not needed. The problem is solved using non-linear minimization like Levenberg-Marquardt (LM) [4] and Bundle Adjustment (BA) [5]. The objective is to minimize the reprojection error of the 3D points given the measurements on the image. These methods produce good results if the initial value of the parameters is close to the global

optimum. Authors, like [6], use the L_∞ norm and convex optimization. However, as the author states, the method is very sensitive to outliers.

Recent works [7] are capable to reconstruct huge scenes using BA, but not in real time. Citing from [7]: “3D reconstruction took 27 hours on 496 compute cores”. Authors like [8] use local BA to get the optimization work in real time, but only use the last five frames.

In this work, an alternative minimization technique is proposed based on Monte Carlo simulations [9] that can run on parallel systems, like GPUs. The advantage of using the GPU is that it has a lot of computing units, called shader processors, that can run simultaneously the same task on different data following the SIMD (Single Instruction Multiple Data) computational model [10]. The benefit of using the GPU comes when the size of the input data is large, making it necessary to adapt existing algorithms to fit this computational model.

The main contribution of this paper is the design of a 3D triangulation algorithm based on this paradigm and an implementation that runs on middle-end graphics adapters. Other authors have used Monte Carlo simulations on GPUs for other topics, like quantum mechanics [11]. However, these algorithms are very specific for each application domain and cannot be compared between them.

Section 2 describes the proposed method and Section 3 its implementation on the GPU. Section 4 evaluates performance and precision of the GPU implementation vs. the widely used LM algorithm given in [12].

2 Problem Description

Let I_k be the image of the frame k . Each image has a set of features associated to it given by a 2D feature tracker $Y_k = \{\mathbf{y}_1^k, \dots, \mathbf{y}_n^k\}$. Feature \mathbf{y}_i^k has (u_i^k, v_i^k) coordinates. The 3D motion tracker calculates the camera motion for each frame as a rotation matrix and a translation vector $\mathbf{x}_k = [\mathbf{R}_k | \mathbf{t}_k]$, where the set of all computed cameras up to frame m is $X_m = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$. The problem consists of estimating a set of 3D points $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ that satisfies:

$$\mathbf{y}_i^k = \Pi(\mathbf{R}_k \mathbf{z}_i + \mathbf{t}_k) \quad \forall i \leq n, \forall k \leq m \quad (1)$$

where Π is the pinhole projection function. The calibration matrix can be obtained if 2D points are represented in normalized coordinates.

The proposed triangulation performs a global minimization over all available images using a probabilistical approach based on Monte Carlo simulations. This consists of sampling inputs randomly from the domain of the problem and weighting them using a goal function that depends on the measurement obtained from the system. As the number of samples increases, the triangulated point will be closer to the optimum. This strategy leads to very computationally intensive implementations that makes it unusable for real time operation. Unlike other minimization methods, each possible solution is computed independently from others, making it optimal for parallel architectures, like GPUs.

In this work, a bottom-up SFM approach is used, but the proposed triangulation algorithm is valid for top-down approaches as well. In the first stage

the feature tracker finds new features and tracks them using optical flow and Kalman filtering [13]. Each feature has a SIFT descriptor [14] used to detect lost features. These features are used by a particle filtering based 3D motion tracker that uses the capabilities of the GPU as well [1]. Finally, all this information is used to reconstruct new 3D points. This last phase is the one covered in this work. The triangulation algorithm follows three steps for each point to be triangulated: initialization, random sampling and weighting.

2.1 Initialization

New points are initialized via linear triangulation. The results are usually far from the optimum, but it is a computationally cheap starting point. For each image k , a 2D feature point \mathbf{y}_i^k can be described as $\mathbf{y}_i^k \times (\mathbf{x}_k \mathbf{z}_i) = \mathbf{0}$, being \mathbf{x}_k the pose for the frame k and \mathbf{z}_i the coordinates of the wanted 3D point (points are expressed in homogeneous coordinates). Having two of these realations, an equation in the form $\mathbf{A} \mathbf{z}_i = \mathbf{0}$ can be written and solved using DLT [15].

This stage is implemented in the CPU since it runs very fast, even when triangulating many points.

2.2 Sampling Points

For each point \mathbf{z}_i , a set of random samples $S_i = \{\mathbf{z}_i^{(1)}, \dots, \mathbf{z}_i^{(q)}\}$ is generated around its neighborhood using a uniform random walk around its initial position:

$$\mathbf{z}_i^{(j)} = f(\mathbf{z}_i, \mathbf{n}_j) = \mathbf{z}_i + \mathbf{n}_j, \quad \mathbf{n}_j \sim U_3(-s, s) \quad (2)$$

where \mathbf{n}_j is a 3-dimensional uniform distribution having minimum in $-s$ and maximum in s that is chosen to be proportional to the prior reprojection error of the point being sampled. In this way, the optimization behaves adaptively avoiding local minimums and handling well points far from the optimum. It is important to scale and translate points before doing the sampling so that the centroid is at the origin and the RMS distance of points to the origin is $\sqrt{3}$. If not, the parameter s should be conveniently scaled for each point. Moreover, floating point units work better with centered and scaled points.

Another good choice for the sampling function would be a Gaussian distribution with mean \mathbf{z}_i and the uncertainty of the point as the standard deviation. However we will leave this as future work.

2.3 Evaluating Samples

All the set S_i for every point \mathbf{z}_i is evaluated in this stage. The objective function is to minimize the L_2 norm of the Equation [1] applied to every hypothesis for every available frame. In other words, the objective is to solve:

$$\operatorname{argmin}_j \left(w_i^{(j)} \right) \quad \text{where } w_i^{(j)} = \sum_{k=1}^m \sqrt{\Pi \left(\mathbf{R}_k \mathbf{z}_i^{(j)} + \mathbf{t}_k \right) - \mathbf{y}_i^k} . \quad (3)$$

The computational complexity of the problem is $O(n * m * q)$, where m is the number of frames, n is the number of 3D points and q is the number of hypotheses per point. However, Equation 3 satisfies the independence needed in stream processing, since each hypothesis is independent from others.

3 GPU Implementation

General Purpose GPU (GPGPU) programming consists of using the programmable 3D graphics pipeline to perform general purpose computing tasks.

The algorithm has been implemented using three shader programs written in GLSL 1.2 [16]. These programs run sequentially for each point to be triangulated. The first shader will generate all the hypotheses for a single point location, the second shader will compute the function Equation 3 and the third shader will choose the best hypothesis. Figure 1 shows an overview of the proposed method.

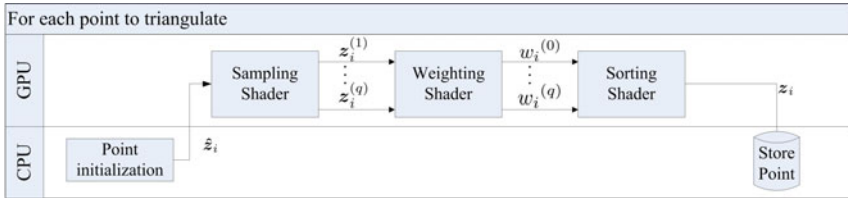


Fig. 1. Overview of the proposed algorithm

Since the GPU is a hardware designed to work with graphics, the data is loaded using image textures. A texture is a matrix of RGBA values, thus each element (s, t) of the matrix, called texel, can be up to a 4-vector of floating point values. The output is the image to be rendered, so the way to obtain the results from it is using a special texture called framebuffer and the render-to-texture mode. It is very important to choose good memory structures since the transfers between the main memory and the GPU memory are very slow.

3.1 Sampling Points

In this stage hypotheses are generated from the 3D point to be optimized using Equation 2. The data needed are the initial position of the 3D point and a texture with uniformly distributed random numbers. Since GLSL lacks random number generating functions, this texture is computed in the preprocessing stage and due to performance optimizations remains constant, converting this method into a pseudo-stochastic algorithm [1]. Therefore, the only datum transferred is the 3D point. The output is a texture containing the coordinates for all hypotheses in the RGB triplet. The alpha channel is used as the value of the objective function, so in this stage, it is initialized to 0. It is not necessary to download the framebuffer

with the generated hypotheses to the main memory, because they are only going to be used by the shader that evaluates the samples. The computational cost of this function is $O(q)$, where q is the total hypotheses count. Currently available GPUs can execute more than 200 hypotheses simultaneously, so this stage runs very fast. Algorithm 1 summarizes the code executed by each stream j .

Algorithm 1. Point Sampling

Require: $\hat{z}_i \rightarrow$ Initial point location

Require: $randomTex \rightarrow$ Random numbers

$[s, t] = \text{GetStreamCoordinates}()$

$[dx, dy, dz] = \text{ReadFromTexture}(randomTex, [s, t])$

$z_i^{(j)} = \hat{z}_i + [dx, dy, dz]$

3.2 Evaluating Samples

All generated hypotheses are weighted applying Equation 3. This shader runs once for each projection \mathbf{y}_i^k using texture ping-pong [17]. This is a common resource in GPGPU that uses two textures as input and output. The texture that acts as input in a render pass, acts as output in the next pass of the shader, and the one that acts as output, acts as input. It is necessary to do it because a texture cannot act as an input and output at the same time due to hardware restrictions. These two textures are generated by the shader described in the previous section. The only data needed to be transferred are the camera pose and the projection of the 3D point for each frame. Another advantage of using one rendering pass per projection, is that in each pass only one point projection and one camera pose is loaded into the GPU, avoiding memory limitations in the algorithm. In this way, the only limitation in the number of frames to optimize is the computational power of the GPU, making this solution very scalable.

The computational cost of this function is $O(m * q)$ where m is the number of frames to be optimized and q the number of hypotheses per point. Algorithm 2 shows the program executed by each stream. This shader program must be executed m times for each 3D point.

Algorithm 2. Hypotheses Weighting

Require: $inputTex \rightarrow$ Texture with sampled points

Require: $\mathbf{x}_k \rightarrow$ Camera pose in frame k

Require: $\mathbf{y}_i^k \rightarrow$ Projection of \mathbf{z}_i in frame k

$[s, t] = \text{GetStreamCoordinates}()$

$[x, y, z, weight] = \text{ReadFromTexture}(inputTex, [s, t])$

$testPoint = \mathbf{R}_k \cdot [x, y, z]^T + \mathbf{t}_k$

$testPoint = testPoint / testPoint.z$

$output.xyz = [x, y, z]$

$output.a = \text{distance}(testPoint.xy, \mathbf{y}_i^k) + inputTex.a$

3.3 Choosing the Best Hypothesis

When all the passes involved in section 3.2 are rendered, the output texture will contain all the hypotheses weighted. The best hypothesis can be searched in the CPU or directly in the GPU. Experimentally, we concluded that GPU search is the fastest way if the size of the texture is big enough.

This search is performed in a parallel fashion using reduction techniques [17]. The ping-pong technique is used here as well, but in each pass the size of the output is reduced. Each output pixel is calculated in function of four texels of the input texture, until the output texture size is reduced to one. Figure 2 illustrates the reduction process in a 4×4 texture. This operation is done in $\log_4(q)$ passes. As seen in Figure 2 it is necessary to work with square textures.



Fig. 2. Minimum Reduction

4 Experimental Results

In order to validate the algorithm, we have ran tests on a real video recorded in 320×240 using a standard webcam. Figure 5 shows a virtual building augmenting the scene used in the tests. Results are compared with the implementation of the LM algorithm given by [12]. In our setup, GFT runs with a viewport of 256×256 , reaching a total of 2^{16} hypotheses per point. The maximum number of iterations allowed to the LM algorithm is 200.

The precision is measured as the mean value of the objective function 3 after the triangulation of 25 points using 15 frames. Figure 3 (in logarithmic scale) shows that GFT gets on average 1.4 times better results than LM. There are no outliers in the data, therefore as seen in the figure, the algorithm depends on a good initial approximation of the point.

The PC used for performance tests is an Intel C2D E8400 @ 3GHz with 4GB of RAM and a nVidia GeForce GTX 260 with 896MB of RAM. Following tests show the performance comparison between GFT and LM. CPU implementations of GFT are not given since they are very slow, even when using multicore processors. Figure 4(a) shows a test running with 10 frames incrementing the point count in each time step. In Figure 4(b), 15 points are optimized, incrementing the number of frames used in each time step. Figures are in logarithmic scales.

These tests show that GFT runs approximately 30 times faster than LM, thus being capable to run at 30fps getting peaks of 100 times better performance when incrementing the number of tracked frames. However, one advantage of using LM

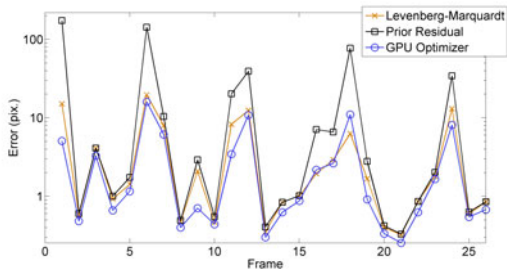


Fig. 3. Residual error on real images

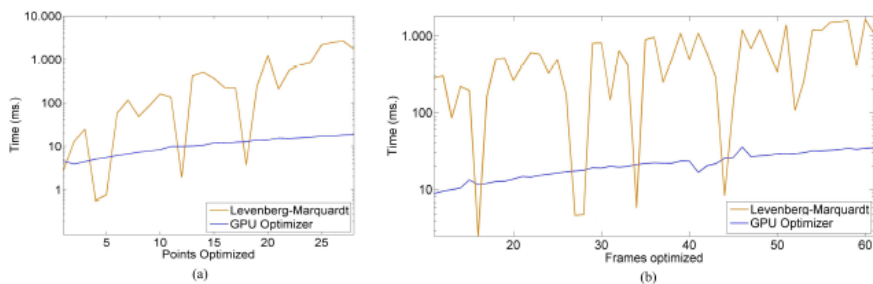


Fig. 4. Comparison with constant point count (a) and with constant frame count (b)

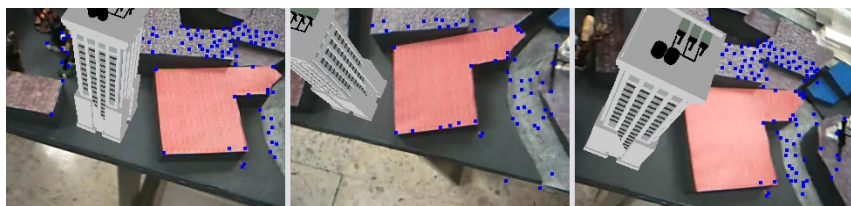


Fig. 5. Augmented video sequence. Blue points are the reconstructed 3D points.

is that it can finish the optimization when the parameters are near the optimal value, as can be seen in some abrupt downwards peaks. GFT could achieve this behaviour running the optimization in various annealing levels, finishing the annealing loop at the desired level of precision.

5 Conclusions

This work presents a method for 3D structure estimation that runs in real time on a GPU. The main contribution is a GPU based parameter estimation framework that can be used to solve many computer vision problems, not only 3D

reconstruction. Since the method relies on Monte Carlo simulations, it is very robust to outliers and highly adaptable to different level of errors on the input.

For validating it, the algorithm is compared against the popular LM algorithm. Tests on real data demonstrate that the GPU optimizer can achieve better results than LM in much less time. This gain of performance allows to increment the number of data used on the optimization, obtaining better precision without sacrificing the real time operation. Moreover, the GPU implementation leaves the CPU free of computational charge so it can dedicate its time to do other tasks. In addition, the tests have been done in a standard PC configuration using a webcam, making the method suitable for commodity desktop hardware.

References

1. Eskudero, I., Sánchez, J., Buchart, C., García-Alonso, A., Borro, D.: Tracking 3d en gpu basado en el filtro de partículas. In: Congreso Español de Informática Gráfica, pp. 47–55 (2009)
2. Welch, G., Bishop, G.: An introduction to the kalman filter. In: SIGGRAPH, Los Angeles, CA, USA (2001)
3. Qian, G., Chellappa, R.: Structure from motion using sequential monte carlo methods. *International Journal of Computer Vision* 59(1), 5–31 (2004)
4. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics* 2(2), 164–168 (1944)
5. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment a modern synthesis. *Vision Algorithms: Theory and Practice* (2000)
6. Kahl, F.: Multiple view geometry and the l infinity norm. In: International Conference on Computer Vision, vol. 2, pp. 1002–1009 (2005)
7. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: International Conference on Computer Vision, Kyoto, Japan (2009)
8. Klein, G., Murray, D.W.: Parallel tracking and mapping for small ar workspaces. In: International Symposium on Mixed and Augmented Reality (2007)
9. Metropolis, N., Ulam, S.: The monte carlo method. *Journal of the American Statistical Association* 44(247), 335–341 (1949)
10. Kapasi, U.J., Rixner, S., Dally, W.J., Khailany, B., Ahn, J.H., Mattson, P., Owens, J.D.: Programmable stream processors. *IEEE Computer*, 54–62 (2003)
11. Anderson III, A.G., Goddard III, W.A., Schröder, P.: Quantum monte carlo on graphical processing units. *Computer Physics Communications* 177(3), 298–306 (2007)
12. Lourakis, M.: levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++ (July 2004), <http://www.ics.forth.gr/~lourakis/levmar>
13. Sánchez, J., Borro, D.: Non invasive 3d tracking for augmented video applications. In: IEEE Virtual Reality 2007 Conference, Workshop Trends and Issues in Tracking for Virtual Environments (IEEE VR 2007), Charlotte, USA, pp. 22–27 (2007)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2(60), 91–110 (2004)
15. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2000)
16. Kessenich, J.: The opengl shading language. Technical report, 3Dlabs (2006)
17. Pharr, M.: GPU Gems 2. Programming Techniques for High-Performance Graphics and General-Purpose Computing (2005)

Methods for Visualization of Bone Tissue in the Proximity of Implants

Hamid Sarve¹, Joakim Lindblad¹, Carina B. Johansson²,
and Gunilla Borgefors¹

¹ Centre for Image Analysis, SLU, Uppsala, Sweden
{[hamid](mailto:hamid@cb.uu.se),[joakim](mailto:joakim@cb.uu.se),[gunilla](mailto:gunilla@cb.uu.se)}@cb.uu.se

² School of Health and Medical Sciences, Örebro University, Sweden
carina.johansson@oru.se

Abstract. In this work we present two methods for visualization of SR μ CT-scanned 3D volumes of screw-shaped bone implant samples: *thread fly-through* and *2D unfolding*. The thread fly-through generates an animation by following the thread helix and extracting slices along it. Relevant features, such as bone ratio and bone implant contact, are computed for each slice of the animation and displayed as graphs beside the animation. The 2D unfolding, on the other hand, maps the implant surface onto which feature information is projected to a 2D image, providing an instant overview of the whole implant. The unfolding is made area-preserving when appropriate. These visualization methods facilitate better understanding of the bone-implant integration and provides a good platform for communication between involved experts.

1 Introduction

Bone anchored medical implants are important for an aging and increasingly osteoporotic population. Our ultimate aim is to improve the understanding of the mechanisms of bone-implant integration. To enable good communication between the people involved in development, production, and use of medical devices — computer scientists, material scientists, and medical doctors — each with their own special knowledge, it is very important to provide a common visual platform for a mutual understanding of the problems of implant integration.

Until recently, implant integration was only evaluated from 2D microscopic images. Such images provide high resolution information, but they only represent a small part of the sample and have a high degree of variability depending on where the 2D slice is taken. Reliable analysis of bone-implant integration requires 3D imaging as the whole sample should be included in the analysis. Once the sample is imaged in situ in 3D, there is a need to visualize the data in a way which highlights the relevant information. The trabecular bone has a sponge-like structure with a huge surface-to-volume ratio. Available software for 3D visualization, which are designed for rendering of solid objects, generally do not provide good visualizations of the porous bone. Such renderings are therefore often difficult to interpret. Visualization techniques that are specially designed

for the task, which provide a simple overview of the bone-implant integration for the whole sample, and in an appealing and easily understandable way display the most relevant information, are therefore of great interest.

In this work, we present two new methods for visualization of imaged volumes of the bone-tissue in the proximity of screw-shaped implants. We image the samples using a Synchrotron Radiation-based micro CT (SR μ CT) device (Sect. 3) and we segment the volumes using a method from our previous work (Sect. 4.1). The first visualization method, *thread fly through* (Sect. 4.3), follows the helix-shaped implant thread from the top of the implant to the bottom and extract slices from the image volume which are assembled into an animation. Along with the slices, traditional features revealing information about the bone-implant integration are computed and presented. The second method, *2D unfolding* (Sect. 4.4), makes an area preserving mapping of the implant surface, with projected feature information, to a 2D image. The results of these visualizations are presented in Sect. 5.

2 Background

Micro Computed Tomography (μ CT) is a commonly used tool for 3D imaging of bone [1, 3, 9]. However, when the samples include metallic implants the imaging becomes complicated [3, 11, 14] as the metal absorbs a considerably higher amount of X-rays than the bone and results in image artifacts.

In recent years, SR μ CT has evolved as a powerful imaging technique for biomedicine and bone studies [3, 4, 5, 7, 14]. This technique yields more accurate tomographic reconstructions due to the parallel beam acquisition and higher resolution compared to standard μ CT [5]. A number of works use SR μ CT for 3D bone quantification and visualization [4, 12, 15].

Today's methods for visualizing this type of data consist almost exclusively of 2D slices, 3D surface renderings, or 3D volume renderings. The 2D slices renderings are relatively easy to interpret, but give a very limited view (see Fig. 1a,b). Existing 3D rendering methods, albeit giving a better overview, give images that are in general difficult to interpret (see Fig. 1c).

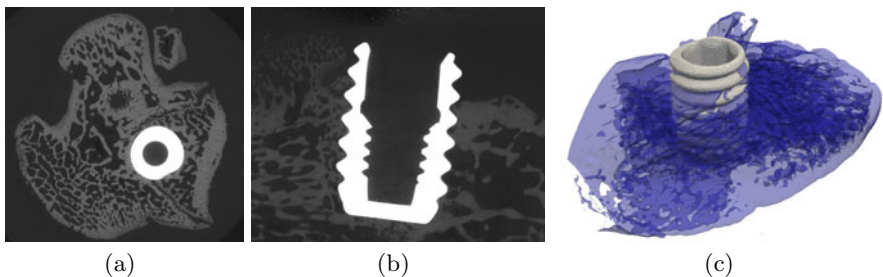


Fig. 1. (a) A horizontal slice from an SR μ CT volume. (b) A vertical slice from the volume. (c) A rendering of the volume showing the implant (*gray*) and bone tissue (*blue*).

3 Materials and Imaging

Pure titanium screws (diam. 2.2 mm, length 3 mm), inserted in the femur condyle region of 12-week-old rats for four weeks, are imaged using the SR μ CT device of GKSS¹ at DESY², in Hamburg, Germany, at beamline W2 using a photon energy of 50keV. 1440 equally stepped radiograms are acquired between 0° and 360°. A filtered back projection algorithm is used to obtain the 3D data of X-ray attenuation for the samples. The field of view of the X-ray detector is set to 6.76mm \times 4.51mm with a pixel size of 4.40 μ m providing a reconstructed image volume of 14.3 \times 14.3 \times 5.5mm with a measured spatial resolution of about 11 \times 11 \times 11 μ m.

4 Method

4.1 Segmentation

The segmentation and artifact correction methods presented in [14] are used to separate the image volumes into three regions: implant, V_I , bone tissue, V_B , and soft tissue, V_{ST} . The implant is a low-noise high-intensity region in the image volume and is segmented by thresholding. The bone and soft tissue are segmented by supervised linear discriminant analysis [8]. The method compensates for shading artifacts, appearing close to the implant boundary.

4.2 Modeling of the Thread Path

As a first step in modeling of the thread path, we calculate the implant symmetry axis, *ISA*, by Principal Component Analysis; each voxel of the V_I is considered a data point and its coordinates (x, y, z) are stored in a matrix M . The axis, where the variance of the segmented implant is the largest, is computed as the normalized eigenvector with the highest corresponding eigenvalue of the covariance matrix of M . For convenience, the image volume is subsequently rotated and translated so that *ISA* is along the z -axis and the xy -plane is at the lowest point of the implant thread. Secondly, we calculate the thread root radius, R , as the minimum distance between *ISA* and the implant surface averaged over all z -levels.

We model the implant thread as a helix, H , following the thread root:

$$H(\phi) = (R \cos(\phi), R \sin(\phi), Z \frac{\phi}{\phi_{max}}), \quad (1)$$

where $\phi \in [0, \phi_{max}]$, $\phi_{max} = 10\pi$, i.e., five full turns, and Z is the height of the implant.

¹ Gesellschaft für Kernenergieverwertung in Schiffbau und Schifffahrt mbH

² Deutsches Elektronen-Synchrotron

4.3 Thread Fly-through

The thread fly-through is visualized as an animation compiled from a volume, $V_{FT}(x, y, t)$. Each frame, t , includes one 2D slice extracted perpendicular to the helix H at an angle $\phi(t)$, where $\phi \in [0, 10\pi]$. For each slice, features described below are calculated and presented together with the animation.

Extraction of the 2D slices. The extraction of the slices utilizes the Graphics Processing Unit (GPU). The sample volume is copied to the 3D texture memory of the GPU and for each t , a 2D quadrilateral is extracted. The vertices are calculated as one point at $H(\phi)$ and $H(\phi + 2\pi)$ and two other points at a distance of $td + d$ away from $H(\phi)$ and $H(\phi + 2\pi)$ respectively, where d is a distance away from the convex hull, CH , of the implant and td is the thread depth, i.e., the distance between the thread root and CH (see Fig. 2a). CH is computed using the *qhull algorithm* [2]. For every turn, we extract n_t slices, i.e., $\phi(t) = \frac{2\pi}{n_t}t$.

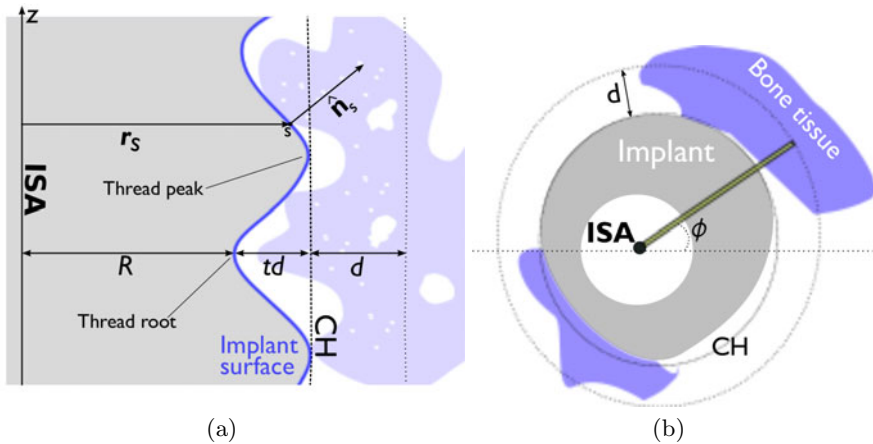


Fig. 2. Illustration of a vertical (left) and a horizontal (right) cross-section. The latter illustrates the radial sampling performed in the 2D unfolding.

Features. The regions of interest (ROIs) are the implant-interface and the region proximate to it. We present two features: bone ratio, BR , and bone contact, BC , which are related to the ROIs. BR is measured as the percentage of area of bone tissue surrounding a thread from its root to a distance d away from CH . BC is the estimated length of the bone to implant contact at the interface, expressed in percentage of the total length of each thread valley. The BR feature is calculated by summing the voxels classified as bone in the ROI. The BC -feature is estimated as described in [13].

4.4 2D Unfolding

Assume an implant volume surface, V_S , with feature information projected onto it. A rendering of V_S visualizes the feature information but requires a

360°-rotation to display the whole surface. To facilitate an immediate overview, we do a cylindrical unfolding of V_S to a 2D image, I , using a mapping $u : \mathbf{Z}^3 \rightarrow \mathbf{Z}^2$, $u(x, y, z) = (\text{atan}(\frac{y}{x}, z))$. The function u unfolds each slice of V_S to a row in I .

The 2D unfolding is performed by a radial projection of the relevant feature information onto the implant surface, followed by an angular sampling (see Fig. 2b). The sampling is made from the ISA as origin for angles $[0, 2\pi]$ by creating an angle histogram with n_b number of bins using the method introduced in [6]. The pixel in row z of I is set to the corresponding bin's value in the angle histogram of slice z . The contribution of each voxel in the sampling is weighted by its coverage of the specific angle.

Features. The features described in Sect. 4.3 are visualized as follows. BC is visualized by generating a volume, $V_{BC} = (V_B \oplus SE) \cap V_I$, where SE is a small structural element and \oplus denotes dilation. The mapping u unfolds the V_{BC} to I_{BC} . If a bin of the angle histogram is non-zero, the corresponding pixel in I_{BC} is considered to be bone tissue in contact with the implant. BR is visualized by generating a volume, V_{BR} , where each surface voxel contains the sum of the voxels of V_B within $td + d$. Analogous to the unfolding above, u unfolds the volume to I_{BR} by radial sampling. To normalize the measure, the value of each bin is divided by $td + d$.

Stretching. For the BR measure, the cylindrical mapping to the 2D image is intuitive. However, for the BC feature, which is strongly connected to the implant surface, it is desirable to have an area preserving mapping. The area variations arise, on one hand, from the difference in surface area depending on the distance between the surface and the ISA (the thread peak has a larger circumference than the thread root) and, on the other, slope of the thread surface. To correctly handle these variations, pixels in I_{BC} are stretched according to a compensation map, C . The pixels of I_{BC} are stretched vertically away from the middle line of I_{BC} by the factors in C .

The compensation map is calculated as $C = C_T \cdot C_A$, where C_T is the distance-to- ISA compensation and C_A the gradient compensation computed as follows: let s be a voxel at the implant surface, then,

$$C_T = \frac{|\mathbf{r}_s|}{R + \frac{td}{2}}, \quad C_A = \frac{1}{\hat{\mathbf{n}}_s \cdot \hat{\mathbf{r}}_s},$$

where \mathbf{r}_s is a vector orthogonal to ISA from ISA to s , $\hat{\mathbf{r}}_s = \frac{\mathbf{r}_s}{|\mathbf{r}_s|}$ and $\hat{\mathbf{n}}_s$ is the unit normal vector of s . The normal vector is calculated using a method by Luo et al. present in [10]. This method uses spatial moments of a ball-shaped window with the diameter w to calculate the normal of the surface.

5 Results

The following settings were used: $d = 2td$, SE is set to a voxel and its face neighbors (a 3D '+'-shape), $n_b = 1/(R + td)$, giving on the average one bin for

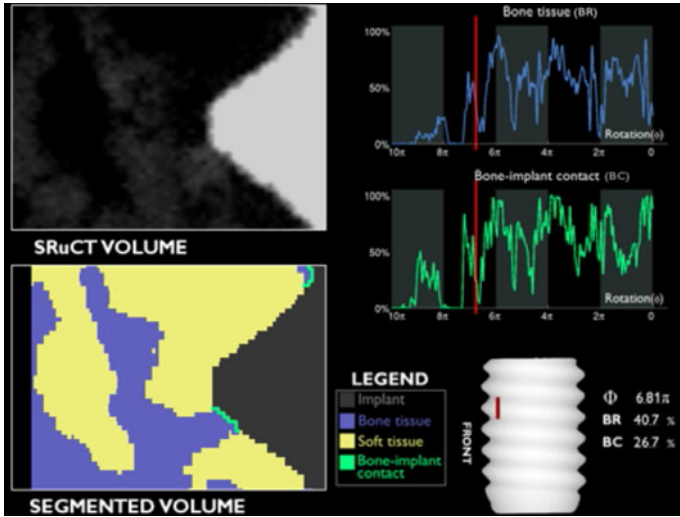


Fig. 3. One frame of the animation, that shows an extracted quadrilateral from the SR μ CT volume and its corresponding segmentation within the ROI. Graphs of BR and BC are shown in the top right. An indicator showing the current position of the extracted quadrilateral is shown in the bottom right.

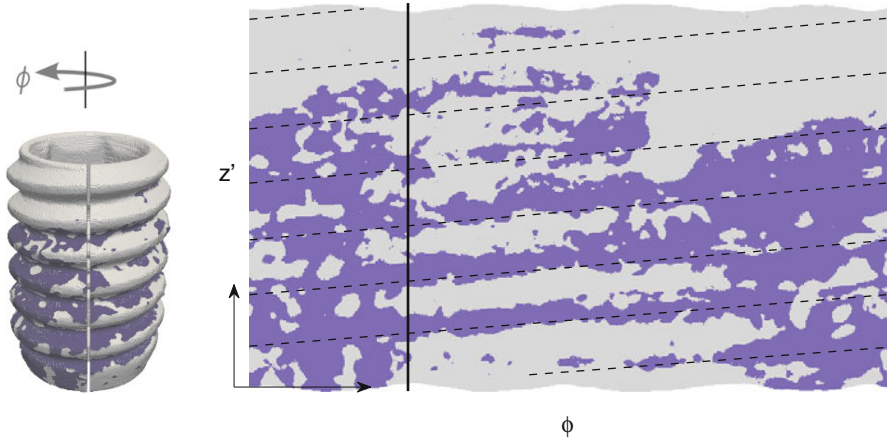


Fig. 4. (Left) Rendered surface of the implant (V_I) with bone-implant contact regions (V_{BC}) superimposed. (Right) The unfolded surface I_{BC} . Black dashed lines show the approximate location of the peaks of the threads. The vertical line indicates the corresponding angles in the two images.

each surface voxel. Analogously, we set $n_t = 2\pi R$ to allow all voxels to be included in V_{FT} . Furthermore w is set to 7 to avoid having noise affecting the compensation map.

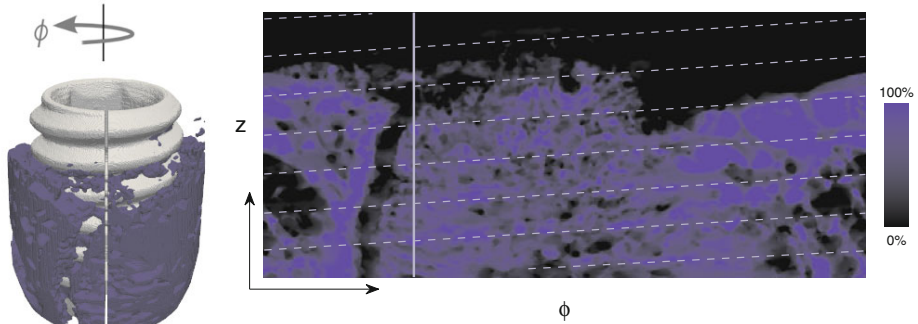


Fig. 5. (Left) Rendered surface of the implant (V_I) with bone tissue volume (V_{BR}) in the region of interest superimposed. (Right) The unfolded surface I_{BR} . White dashed lines show the peaks of the threads. The vertical line indicates the corresponding angles in the two images.

Five implant samples are visualized. For each sample we compute two 2D unfoldings (one for each feature) and one thread fly-through animation. The animations are available at <http://www.cb.uu.se/~hamid/flythrough>. A screenshot of one animation is shown in Fig. 3. The animations show the extracted quadrilateral from the SR μ CT volume and its corresponding segmentation. Furthermore, graphs of BR and BC and an indicator showing the current position of the extracted quadrilateral are shown. Note that the fly-through is focused on the thread peak in the current setting. It is very easy to shift the focus to the thread valley or multiple threads instead, if desired.

The result of the 2D unfoldings for one of the implants is shown in Fig. 4 and Fig. 5. Thanks to the unfolding, it is much easier to notice the canyon with low bone ratio (left side of the image) in Fig. 5 than the surface rendering in Fig. 1c.

6 Conclusion

In this work we present two new methods for visualizing SR μ CT-scanned volume samples of bone implants. These visualization methods provide an improved insight in bone-implant integration. The animations provides information about the bone-implant integration over the whole sample in an understandable way. The 2D unfoldings give a direct overview of the bone-implant contact of the surface of the implant and the bone concentration in the proximity of the implant. It now remains to evaluate the methods to see which material scientists and medical doctors prefer and what extra information they can extract from them, compared to present standard visualizations.

This work is performed in close collaborative with and according to requests and feedback from material scientists and medical experts at Astra Tech AB and Örebro University.

Acknowledgment. We would also like to acknowledge Astra Tech AB for their support. This work was supported by grants from The Knowledge Foundation, project 2008/0716.

References

1. Balto, K., et al.: Quantification of Periapical Bone Destruction in Mice by Micro-computed Tomography. *Journal of Dental Research* 79, 35–40 (2000)
2. Barber, C.B., Dobkin, D., Huhdanpaa, H.: The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software* 22(4), 469–483 (1996)
3. Bernhardt, R., et al.: Comparison of Microfocus- and Synchrotron X-ray Tomography for the analysis of Osteointegration Around Ti6Al4V-Implants. *European Cells and Materials* 7, 42–50 (2004)
4. Bernhardt, R., et al.: 3D analysis of bone formation around titanium implants using micro computed tomography. In: *Proc. of SPIE*, vol. 6318 (2006)
5. Candecca, R., et al.: Bulk and interface investigations of scaffolds and tissue-engineered bones by X-ray microtomography and X-ray microdiffraction. *Biomaterials* 28, 2506–2521 (2007)
6. Gavrilovic, M., Wählby, C.: Quantification of Colocalization and Cross-talk based on Spectral Angles. *Journal of Microscopy* 234, 311–324 (2009)
7. Ito, M.: Assessment of bone quality using micro-tomography (micro-CT) and synchrotron micro-CT. *Journal of Bone Miner Metab* 23, 115–121 (2005)
8. Johnson, R.A., Wichern, D.W.: *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs (1998)
9. van Lenthe, G.H., Muller, R.: CT-Based Visualization and Quantification of Bone Microstructure In Vivo. *IBMS BoneKEy* 5(11), 410–425 (2008)
10. Luo, L.M., et al.: A moment-based three-dimensional edge operator. *IEEE Trans. Biomed. Eng.* 40, 693–703 (1993)
11. Numata, Y., et al.: Micro-CT Analysis of Rabbit Cancellous Bone Around Implants. *Journal of Hard Tissue Biology* 16, 91–93 (2007)
12. Peyrin, F., Cloetens, P.: Synchrotron radiation μ CT of biological tissue. *IEEE ISBI*, 365–368 (July 2002)
13. Sarve, H., et al.: Quantification of Bone Remodeling in the Proximity of Implants. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) *CAIP 2007*. LNCS, vol. 4673, pp. 253–260. Springer, Heidelberg (2007)
14. Sarve, H., et al.: Quantification of Bone Remodeling in SR μ CT Images of Implants. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) *SCIA 2009*. LNCS, vol. 5575, pp. 770–779. Springer, Heidelberg (2009)
15. Weiss, P., et al.: Synchrotron X-ray microtomography (on a micron scale) provides three-dimensional imaging representation of bone ingrowth in calcium phosphate biomaterials. *Biomaterials* 24, 4591–4601 (2003)

An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation^{*}

Adam Schmidt, Marek Kraft, and Andrzej Kasiński

Poznań University of Technology, Institute of Control and Information Engineering
Piotrowo 3A, 60-965 Poznań, Poland
{Adam.Schmidt,Marek.Kraft}@put.poznan.pl

Abstract. The detection and matching of feature points is crucial in many computer vision systems. Successful establishing of points correspondences between concurrent frames is important in such tasks as visual odometry, structure from motion or simultaneous localization and mapping. This paper compares the performance of the well established, single scale detectors and descriptors and the increasingly popular, multi-scale approaches.

1 Introduction

Many of today's computer vision applications – from mobile computers to robotic systems – require robust, markerless object detection. An increasingly popular approach to achieve this goal is the detection and matching or recognition of distinctive, natural image features across scene views. The developments in this field resulted in development of feature detectors and descriptors that can successfully cope with realistic application scenarios. In this paper, we investigate the usefulness of various detector-descriptor pairs in mobile robotics application. The motivation behind this is the fact, that in our opinion, characteristics of the features detected with a specific detector can significantly influence the matching process.

Specifically, we are interested in application of the feature detection and matching in mobile robot navigation (visual odometry [1] and SLAM [2]) in indoor environments. In this specific scenario, the inter-frame transformations of feature neighborhood used for description are relatively small, as the robot movement is mostly planar, and the environments is (in most cases) relatively feature-rich. The results we obtained can help to decide if the navigation process can (in the aforementioned circumstances) benefit from using more complex multiscale detectors and descriptors. The multi-scale approach, despite recent progress, is still time-consuming and the additional computational overhead may be unjustified in time-critical applications if it can be avoided.

^{*} This project is partially funded by the Polish Ministry of Science and Higher Education, project number N N514 213238.

2 Feature Detectors and Descriptors

This section contains short description of the investigated image feature detectors and descriptors.

2.1 Feature Detectors

Harris Corner Detector. The popular Harris corner detector is based on investigating local auto-correlation function of the signal [3]. Variations of the auto-correlation function over principal directions are investigated. This is done using the so called structural tensor:

$$A = \sum_u \sum_v w(p, q) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (1)$$

where I_x and I_y denote the partial image derivatives in x and y directions, respectively, and $w(p, q)$ denotes a weighting window over the area (p, q) . Most common choice for the weighting window is a Gaussian, as it makes the responses isotropic. The criteria for cornerness measure is:

$$M_c = \det(A) - \kappa \text{trace}^2 A \quad (2)$$

where κ is a tunable sensitivity parameter. Corners are the local maxima of M_c . Additional thresholding of the cornerness measure is applied to filter out the weak responses.

Shi Tomasi feature detector. The Shi Tomasi corner detector is strongly based on the Harris corner detector [4]. The difference lies in the cornerness measure M_c , which is computed as the minimum of the absolute values of the eigenvalues of the structural tensor (II).

$$M_c = \min(|\lambda_1|, |\lambda_2|) \quad (3)$$

Corners are the local maxima of M_c . Such cornerness measure is said to be more robust than (2) while the image patches that surround features undergo affine transformations.

FAST Corner Detector. FAST stands for Features from Accelerated Segment Test [5]. The corner detector performs two tests. At first, candidate points are being detected by applying a *segment test* to every image pixel. Let I_p denote the brightness of the investigated pixel p . The test is passed, if n contiguous pixels on a Bresenham circle with the radius r around the pixel p are darker than $I_p - t$ ('dark' pixels), or brighter than $I_p + t$ ('bright' pixels), where t is a threshold value. The authors use a circle with $r = 3$, and $n = 9$ for best results [5]. The ordering of questions used to classify a pixel is learned by using the ID3 algorithm, which speeds this step up significantly. As the first test produces many adjacent responses around the interest point, additional criterium is applied to

perform a *non-maximum suppression*. This allows for precise feature localization. The cornerness measure used at this step is:

$$M_c = \max\left(\sum_{x \in S_{\text{bright}}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{\text{dark}}} |I_p - I_{p \rightarrow x}| - t\right) \quad (4)$$

where $I_{p \rightarrow x}$ denotes the pixels laying on the Bresenham circle. As the second test is performed to only a fraction of image points that passed the first test, the processing time remains short.

SURF Feature Detector. SURF (Speeded Up Robust Features) is a image feature detector and descriptor [6], inspired by the SIFT detector [7], known for its high performance. SURF is designed with emphasis on speed, being SIFT’s main weakness. SURF is said to be few time faster than SIFT with no performance drop. To achieve this, the detector uses the Haar wavelet approximation of the blob detector based on Hessian determinant. Haar wavelet approximations can be efficiently computed at different scales using integral images [10]. Accurate localization of features requires interpolation.

Star Keypoint Detector. Star keypoint detector is a part of OpenCV computer vision library. It was derived from CenSurE (Center Surround Extrema) feature detector introduced in [8]. The main motivation behind development of this detector was to achieve a full spatial resolution in a multiscale detector. According to [8], as SIFT and SURF detectors perform subsampling, the accuracy of feature localization is affected. CenSurE detector uses a bi-level approximation of the the LoG filter. The circular shape of the mask is approximated by shapes, which allow to combine simplicity and rotation invariance. CenSurE uses octagon shaped filter masks, while the star keypoint detector mask shape is made of two squares, one of which is rotated by 45 degrees (see figure 1).

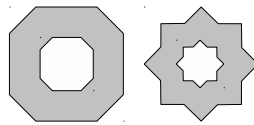


Fig. 1. Filters used by CenSurE (left) and star (right) keypoint detectors

Such filters, regardless of size, can be efficiently computed using integral images. Instead getting scale-space by subsampling, the scale-space is constructed by applying filters of different size, thus allowing for precise localization without interpolation.

2.2 Feature Descriptors

Correlation Based Similarity Measures. Matching by correlation based similarity is achieved by taking a square window of certain size around the feature found in the reference image to find the best matching feature in another

image – one with the highest or the lowest (depending on the measure used) similarity score. For our experiments we have chosen to use three common similarity measures: SAD (sum of absolute differences), SSD (sum of squared differences) and NCC (normalized cross-correlation) [9]. SAD and SSD are the simplest similarity measures, SSD being the more discriminative (image patch differences are amplified by squaring). NCC is more complicated, but is supposed to offer more robustness to lighting and contrast changes [9]. This is achieved by normalization.

SURF Family of Descriptors. The SURF family of descriptors encodes the distribution of pixel intensities in the neighborhood of the detected feature at the corresponding scale. The use of Haar wavelets in conjunction with integral images allows to decrease computation time [10]. The wavelets allow to determine the gradient values in x and y direction. Computation of the descriptor for a given feature can be divided into two distinct tasks.

The first task is the dominant orientation assignment. This step allows to achieve the rotation invariance. First, the Haar wavelet responses are computed for all points within the radius $6s$ of the detected interest point, where s is the coefficient denoting the scale, at which the interest point was detected. The size of the wavelets is also properly rescaled – the side length is set to $4s$. The responses are then weighted with a Gaussian with the $\sigma = 2s$. Each response is a point in the vector space with the x -responses along the abscissa and the y -responses along the ordinate. The dominant orientation is selected by rotating a circle segment covering an angle of $\frac{\pi}{3}$ rad around the origin. The responses in each segment are then summed and form a resultant vector. The orientation of the longest resultant vector is selected as the dominant orientation for the descriptor.

The construction of the descriptor itself starts with selecting a square window with the side size of $20s$ that is centered on the interest point and properly oriented (as computed in the previous step). The window is divided regularly into 4×4 square subregions. Each one of these subregions is divided into 5×5 regularly spaced sample points. Next, Haar wavelet response for each of the principal directions (dx, dy) is computed at each of these sample points. The absolute values of these responses are also incorporated into the descriptor. The directions are considered with respect to the dominant orientation. The complete form of the descriptor for each of the subregions is computed by summing the responses and their absolute values acquired at each of the sample points (see equation 5).

$$DESC_{sub} = [\sum dx, \sum dy, \sum |dx|, \sum |dy|] \quad (5)$$

The responses are weighted with a gaussian centered at the interest point to increase the robustness. Each subregion contributes with 4 elements of the descriptor vector. For 16 subregions, the descriptor size is therefore 64. Hence, the descriptor name is SURF64. The authors proposed also a simplified version of the descriptor, in which the window is divided into 3×3 square subregions. This reduced the descriptor's dimensionality and allowed for faster matching. The name

of this variant is SURF36. If the rotational invariance is not required, the upright version of the descriptor, called USURF can be used. Skipping the dominant orientation assignment step yields further reduction of the time required for computation.

In our experiments, we used an additional modification to the original SURF descriptor proposed in [8]. To eliminate the boundary effects, the subregions are extended and overlap. The overlap size is $2s$. This is called MSURF, and its upright version – MUSURF.

3 Experiment Description

The detector-descriptor pairs were evaluated on a sequence registered while driving a mobile robot around a lab room. The path was approximately 20 meters long. The sequence is one minute long and consists of 758 frames. The UI-1225LE-C uEye camera (IDS Imaging) with 1/3" CMOS sensor was used. The camera enables fast registration (up to 87 FPS) with the global shutter. The camera was equipped with a wide-angle lens ($f = 2.8mm$). No filtering and rectification of images was performed. To maintain stable framerate, exposure time was set to a constant value, but the automatic gain control function was used. The sequence displays many realistic effects – noise, distortions, contrast variations etc.

Each one of the described feature detectors was tested with each one of the described corner descriptors. No restrictions were made to the search area and no cross checking was performed – we always kept the feature in Image 2 that best matched the feature in Image 1. Matching was performed with five different threshold settings, to find the best compromise between the number of matched features and the match quality.

To evaluate the quality of matches, we used the 8-point algorithm for fundamental matrix computation [11], with robust estimation method based on the RANSAC (random sample consensus) algorithm [12]. RANSAC is an iterative method allowing for the estimation of a mathematical model (the fundamental matrix in our case) from a set of observations (matched points) containing outliers (false matches). The ratio of inliers (i.e. feature matches consistent with the estimated fundamental matrix) to all the detected matches is the indicator of the quality of the matching process. Additionally, any case in which the number of inliers was less than 10 was treated as a matching failure. The symmetric reprojection error [11] was used to check consistency of the data with mathematical model. To further examine the accuracy of the interest point location in images, the estimation using RANSAC was performed with two different thresholds: $\theta = 0.5$ pixel and $\theta = 1.0$ pixel. Example images from the sequence are given in figure 2.

4 Results and Discussion

Tables 1 and 2 show the results of running different detector-descriptor pairs on the test sequence with the threshold set to $\theta = 0.5$ and $\theta = 1.0$ pixels. The results



Fig. 2. Example frames from the registered sequence

are the ratios of inliers to all the detected matches averaged over the whole run. The detection and matching was performed for each consecutive frame pair. It's worth noting, that there were no failed matches (i.e. frames with fewer than 10 inliers) in this case.

Table 1. Ratio of inliers to all detected matches, $\theta = 0.5$, frame to frame matching

	SAD	SSD	NCC	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.60	0.67	0.61	0.84	0.88	0.83	0.86
Harris	0.61	0.66	0.64	0.85	0.87	0.84	0.85
Shi-Tomasi	0.71	0.79	0.68	0.77	0.80	0.90	0.90
SURF	0.54	0.56	0.63	0.78	0.80	0.79	0.81
Star	0.61	0.65	0.71	0.74	0.75	0.73	0.73

Table 2. Ratio of inliers to all detected matches, $\theta = 1.0$, frame to frame matching

	SAD	SSD	NCC	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.77	0.83	0.76	0.93	0.95	0.935	0.95
Harris	0.79	0.83	0.80	0.95	0.96	0.953	0.96
Shi-Tomasi	0.88	0.92	0.83	0.91	0.94	0.973	0.98
SURF	0.64	0.65	0.72	0.90	0.92	0.912	0.92
Star	0.76	0.79	0.86	0.92	0.92	0.914	0.92

The results show, that the MSURF descriptor family outperforms other descriptors. Another thing worth noting is that the differences in performance between MSURF descriptors with different dimensionality are relatively small. This leads to the conclusion, that for dense sampling it is safe to use the shorter descriptor. The multiscale detectors behave in general worse than Harris, FAST and Shi-Tomasi detectors while paired with the same descriptor. A few interesting phenomena were observed. The Shi-Tomasi feature detector, contrary to all the other detectors, performed better with the upright version of SURF family descriptors. The reason is probably the fact, that the detector responds also to edge-like features. Assigning dominant orientation is therefore harder than it is

the case with the other detectors, responding to corner-like features. False orientation assignment leads to mismatches. As there are no significant in-plane rotations, the standard and upright versions of MSURF perform similarly in terms of ratios of inliers to all the detected matches averaged over the whole sequence. However, its versions that detect dominant rotation tend to produce more matches. Interestingly, the Star detector based on CenSurE, which we expected to perform better than SURF in terms of accuracy (according to the claims in [8]) outperformed SURF only when paired with SAD, SSD and NCC matching schemes. However, it was outperformed by SURF when any of the MSURF descriptors was used for matching with $\theta = 0.5$, which is the case that requires more accurate feature localization. Despite the changes in illumination and contrast, NCC is not apparently better than SAD or SSD. The exception to this rule is the Star detector, which seems to work well combined with NCC. Running the RANSAC estimation with different thresholds for allowed reprojection error ($\theta = 0.5$ and $\theta = 1.0$) does not prove that any of the described detectors is more accurate than any other, as no general conclusions holding for all matching schemes can be drawn. The Star detector combined with MSURF descriptors is the only exception, as it was mentioned earlier.

In our next experiment, the detection and matching was performed between every 10th frame of the sequence. As the SURF descriptor family achieved the highest performance in the previous test, SAD, SSD and NCC matching was not performed in this case. The Shi-Tomasi feature detector was not used, because it was impossible to achieve accuracy (in terms of average inlier number to match number ratio) on par with the other detectors without introducing matching failures (less than 10 inliers among the matches). The evaluation criteria remained the same. The results are shown in tables 3 and 4.

The percentage of inliers is clearly lower when compared with tables 1 and 2. Relaxation of matching threshold from 0.1 to 0.2 was necessary, to avoid frames with too few inliers. The performance difference between multiscale and single

Table 3. Ratio of inliers to all detected matches, $\theta = 0.5$, every 10th frame matching

	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.51	0.57	0.53	0.57
Harris	0.52	0.58	0.54	0.56
SURF	0.47	0.52	0.49	0.52
Star	0.51	0.56	0.50	0.53

Table 4. Ratio of inliers to all detected matches, $\theta = 1.0$, every 10th frame matching

	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.67	0.73	0.72	0.75
Harris	0.70	0.75	0.73	0.75
SURF	0.65	0.69	0.66	0.69
Star	0.69	0.73	0.68	0.71

scale detectors is not as apparent as it is in the case of small inter-frame differences, but the single scale detectors still demonstrate better performance. An exception to this rule is the FAST detector combined with MSURF36. The the descriptor length seems to be more important as the baseline gets longer – 64-element descriptors offer better inlier ratio in this case. Interestingly, upright versions of the MSURF descriptors seem to perform better in terms of inlier ratio. However, the versions with dominant orientation detection return more matches. The results show, that also in the case of more significant displacements it is still safe to use single-scale detectors paired with the performant MSURF family descriptors. Multiscale algorithms would probably get better notes under longer displacements and severe view angle change, scaling or rotations. Such extreme conditions are however uncommon in most mobile robotics applications.

5 Conclusions

We have presented the results of performance evaluation of different feature detector-descriptor pairs. The performance was tested with focus on indoor mobile robot navigation applications. The results show, that for this application domain, using single-scale feature detectors paired with the MSURF family descriptors does not introduce any performance penalty when compared to state of the art multiscale detectors. The use of single-scale approach allows to reduce the computational overhead. Further reduction in processing time us possible by using the upright family of MSURF descriptors, as indoor applications usually do not introduce significant in-plane rotations in successive frames. Matching can be sped up by using the reduced version of the description vector. The effect on performance depends in this case on the inter-frame differences. All the tests were performed on unrectified, unfiltered images taken by a moving camera with a wide angle lens, without restrictions on matching area. Using additional criteria e.g. considering the ratio of distance from the closest match to the distance of the second closest [7] and reducing the search radius would improve the results.

References

1. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 652–659 (2004)
2. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on PAMI* 29(6), 1052–1067 (2007)
3. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (1988)
4. Shi, J., Tomasi, C.: Good Features to Track. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
5. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding* 110(3), 346–359 (2008)

7. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
8. Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center surround extremas for realtime feature detection and matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
9. Banks, J., Corke, P.: Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision. *The Int. J. of Robotics Research* 20(7), 512–532 (2001)
10. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: *Proc. of Int. Conf. on Image Processing*, vol. 1, pp. 900–903 (2002)
11. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn., vol. 24, pp. 381–395. Cambridge University Press, Cambridge (2004)
12. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24, 381–395 (1981)

The Visual SLAM System for a Hexapod Robot*

Adam Schmidt and Andrzej Kasiński

Institute of Control and Information Engineering
Poznań University of Technology, Poznań, Poland
{Adam.Schmidt, Andrzej.Kasinski}@put.poznan.pl

Abstract. The precise localization of the mobile robot plays a vital role in the autonomous operation of the mobile robot. The vision based simultaneous localization and mapping (SLAM) is a widely known technique for tracking the movement of the camera in the unknown environment. This paper presents a robot's movement model which is based on the reference trajectory of the robot. The proposed model was compared with the state-of-the-art model used in the successful MonoSLAM system [8] and provided good results.

Keywords: SLAM, hexapod, mobile robotics.

1 Introduction

The ability to work in an unexplored environment is crucial to the success of the autonomous robot's operation. The simultaneous localization and mapping (SLAM) is a technique used to build a map of the robot's surrounding while keeping track of its position. In recent years significant attention has been paid to the development of a vision based SLAM. The reasons behind this popularity are the low price and the availability of cameras, large amounts of information contained in images as well as relatively simple mathematical models of cameras. Moreover, the use of the visual information is intuitive and the visual SLAM system can be easily integrated with the tele-operation system. The main drawback of such systems is that cameras are bearing-only sensors i.e no information regarding the depth of the scene is available.

The work of Davison and Murray [1] is considered to be the first vision based, real-time system able to construct a map, track and control a mobile robot simultaneously. Sim et al [2] combined the SIFT feature descriptor with the FastSLAM filtering to desing a large-scale visual SLAM system. In his PhD dissertation Baczyk [3] introduced artificial landmarks spread over the environment which facilitated loop closing. Recently Gil et al. [4] proposed a multi-robot system for the cooperative exploration of the unknown environment. Mouragnon et al. [5] proposed a visual SLAM system for a mobile robot using holistic features

* This research was supported by the Polish Ministry of Science and Higher Education under Grant No. N514 294635 for the years 2008-2010, which is gratefully acknowledged.

matching between consecutive images, key frames selection and local bundle adjustment. Sturm and Visser [6] proposed a visual compass for the mobile robot enabling fast orientation estimate within any cylindrical space.

The MonoSLAM approach presented by Davison [8] is one of the most successful approaches to the visual SLAM. The robot is modeled as a rigid body which linear and angular velocities change smoothly as a result of random, external accelerations. The authors of this approach focus on developing a real-time system and strive to achieve as high frame rate as possible.

Unfortunately, these assumptions are generally not valid in case of the hexapod robot. Its construction allows fast and abrupt changes of the movement direction. Moreover, the movement of the robot is not random but determined by the orders sent to the robot’s controller (yet the results of those orders remain uncertain due to slipping, limited precision of servoes and the structure of the environment). Additionally, the onboard computer of most robots has limited resources. Therefore, instead of maximizing the frame rate (and the computational load) it is advisable to develop a model capable of precise localization at a low frame-rate. This paper presents the attempt to develop a movement model incorporating the knowledge of the robot’s expected trajectory and to integrate it with the state-of-the art methodology of the visual SLAM.

2 System

2.1 Probabilistic Map

The probabilistic, feature-based map approach adopted from [7, 8] was used to model the robot’s environment. The map provides the current estimates of the robot’s and features’ positions as well as the uncertainty of those estimates. The state vector x of the map consists of the robot’s state and the state vectors of all the features. The uncertainty of the state is modeled as a single, multivariate Gaussian and is represented by the covariance matrix P which can be partitioned in the following manner:

$$x = \begin{bmatrix} x_r \\ x_f^1 \\ x_f^2 \\ \vdots \end{bmatrix}, P = \begin{bmatrix} P_{x_r x_r} & P_{x_r x_f^1} & P_{x_r x_f^2} & \dots \\ P_{x_f^1 x_r} & P_{x_f^1 x_f^1} & P_{x_f^1 x_f^2} & \dots \\ P_{x_f^2 x_r} & P_{x_f^2 x_f^1} & P_{x_f^2 x_f^2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1}$$

The proposed system was designed for the hexapod robot (Fig. 1) presented in [10] equipped with the uEye camera. The robot’s coordinate system is situated in the center of the lower plate and the state vector consists of a 3D position r and a rotation quaternion q relative to the global frame:

$$x_r = [r \ q]^T = [r_x \ r_y \ r_z \ q_a \ q_b \ q_c \ q_d]^T \tag{2}$$

The inverse depth parametrisation of the features state was used [9]. Each feature’s state vector consists of six elements:

$$x_f^i = [x_i \ y_i \ z_i \ \theta_i \ \phi_i \ \rho_i]^T \tag{3}$$

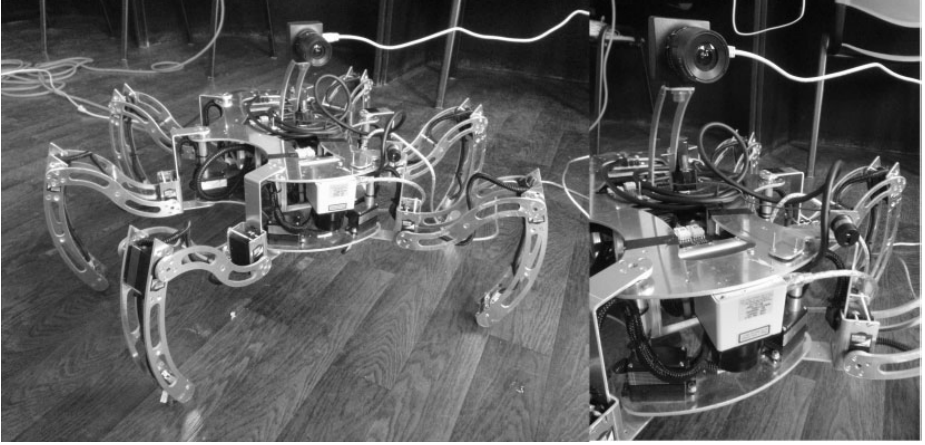


Fig. 1. The hexapod robot

The x_i, y_i, z_i represent the position from which the feature was observed at the first time, θ_i and ϕ_i are the angles encoding the ray between the camera and the feature, and ρ_i is the inverse of the distance between the camera and the feature. The 3D position of the feature is calculated as:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \begin{bmatrix} \cos(\phi_i) \sin(\theta_i) \\ -\sin(\phi_i) \\ \cos(\phi_i) \cos(\theta_i) \end{bmatrix} \quad (4)$$

Additional information regarding this representation can be found in [9].

The Extended Kalman Filter is used to update the state vector and the corresponding covariance matrix P with an assumption that the robot moves in the static surrounding. The dimension of the model may change during the operation as features can be added and deleted. The covariance matrix is sparse and many of its subblock are zero matrices. To facilitate adding and removing of features and to take the advantage of the covariance matrix' sparseness both the state vector and covariance matrix were implemented as hash tables of matrices.

2.2 Motion Model and Prediction

The proposed model is based on the notion that the commands sent to the robot can be used to predict its position. The robot's orders consist of the desired translation and rotation (parametrised by the XYZ Euler angles relative to the robot) expressed in the robot's local frame. Therefore, the inputs vector takes the form of:

$$u(t) = [t_x \ t_y \ t_x \ \alpha \ \beta \ \gamma]^T \quad (5)$$

where t_x, t_y, t_z are the expected displacements and α, β, γ are the rotation angles. The motion of the robot is modelled by the following prediction function:

$$x_r^{new} = \begin{bmatrix} r + R_W^R t \\ q \times q_\alpha \times q_\beta \times q_\gamma \end{bmatrix} \tag{6}$$

where R_W^R is the rotation matrix between the global and the robot’s frames, t is the translation vector, $q_\alpha, q_\beta, q_\gamma$ stand for the quaternions representing Euler angles’ rotations and \times is the Hamilton product.

As there is no direct feedback in the movement control the resulting robot’s displacement differs from the expected one. This inputs uncertainty (caused mainly by the robot’s leg slipping and servoes imprecision) is modeled as a multivariate gaussian and is represented by a covariance matrix P_i . Currently, it is assumed that the inputs are independent and the P_i is a diagonal matrix. Jacobians of the prediction function over the robot’s state and the robot’s inputs are used to increase the ucertainty of the system.

2.3 Measurement

The directional vector $h_i = [h_x^i \ h_y^i \ h_z^i \ 1]$ defines the line passing through the camera center and the i -th feature:

$$h_i^C = T_C^R T_R^W [X_i \ Y_i \ Z_i \ 1]^T \tag{7}$$

where T_C^R is a constant homogenous transformation between the camera’s and robot’s frames defined by the robot’s design and T_R^W is a homogenous transformation between the robot’s frame and the global coordinate system depending on the robot’s state.

This vector is used to calculate the projection of the feature onto the image plane:

$$h_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} u_0 - f_u \frac{h_x^i}{h_z^i} \\ v_0 - f_v \frac{h_y^i}{h_z^i} \end{bmatrix} \tag{8}$$

where u_0, v_0, f_u and f_v are the perspective camera intrinsic parameters. After calculating jacobians of h_i with respect to the feature’s and robot’s state the Kalman filter innovation matrix can be computed:

$$S_i = \frac{\delta h_i}{\delta x_r} P_{x_r, x_r} \frac{\delta h_i^T}{\delta x_r} + \frac{\delta h_i}{\delta x_r} P_{x_r, x_f^i} \frac{\delta h_i^T}{\delta x_f^i} + \frac{\delta h_i}{\delta x_f^i} P_{x_f^i, x_r} \frac{\delta h_i^T}{\delta x_r} + \frac{\delta h_i}{\delta x_f^i} P_{x_f^i, x_f^i} \frac{\delta h_i^T}{\delta x_f^i} + P_h \tag{9}$$

where P_h models imager noise covariance.

The S_i represents the 2D Gaussian PDF of feature’s projection in the image coordinates, which can be used to define the search ellipse. A given number of visible features with biggest search ellipses is selected as a measurement at a current iteration. The exhaustive search is performed within the ellipses with the normalized cross-correlation between the image patch and the feature template as a measure of similarity (Fig. 2).

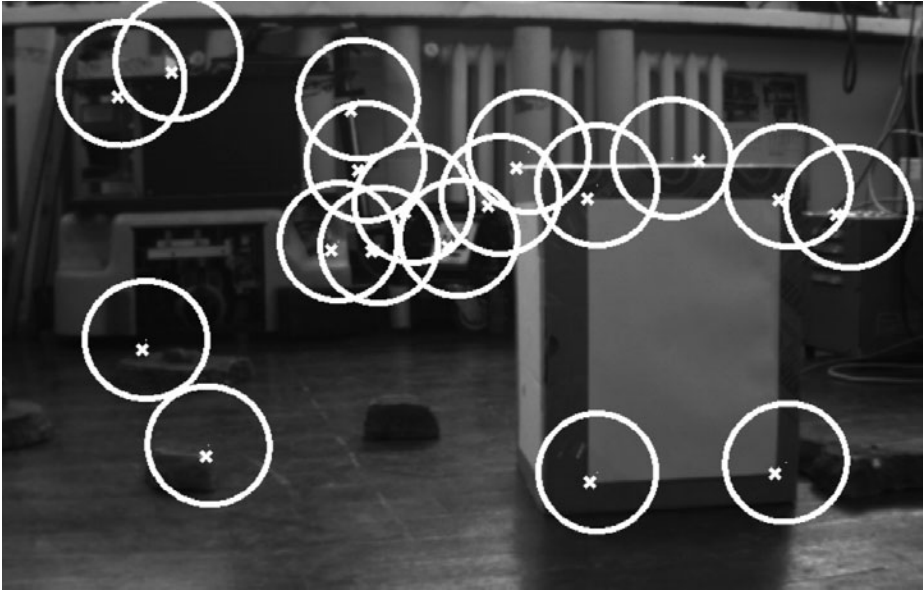


Fig. 2. The exemplary frame from the robot’s camera with the search ellipses

2.4 Map Management and Initialization

It is assumed that during the initialization the robot is placed in the origin of the global coordinate system with a zero uncertainty. To set up the scale of the map a rectangular marker is detected and its corners are added to the map as the first four features. The position of those features is calculated according to the analytical solution of the perspective 4-point problem with coplanar points presented in [11]. Additional feature points are detected with the FAST detector [12].

Whenever the number of observed features is lower than the number of expected features the interest points are detected with the FAST detector and new features are added to the map. If a feature cannot be detected a given number of times despite being theoretically visible it is removed from the map.

3 Experiments

The proposed model movement model was compared with the ‘agile camera’ model proposed by Davison [8]. Three sequences have been chosen to illustrate the features of the presented model:

- the smooth trajectory:

$$u_1(i) = [0 \ 2 \ 0 \ 0 \ 0 \ 1.5^\circ]^T \quad (10)$$

- the two-parts trajectory with a smooth start, abrupt change of direction and elevation changes:

$$u_2(i) = \begin{cases} [0 \ 2 \ 0 \ 0 \ 0 \ 1.5^\circ]^T & \text{if } i \in \langle 1, 10 \rangle \\ [-2 \ 1.25 \ (i - 10) \ 0 \ 0 \ -1.5^\circ]^T & \text{if } i \in \langle 11, 13 \dots \rangle \\ [-2 \ 1.25 \ -(i - 10) \ 0 \ 0 \ -1.5^\circ]^T & \text{if } i \in \langle 12, 14 \dots \rangle \end{cases} \quad (11)$$

- discontinuous movement to the sides and rotations around the robot's z axis:

$$u_3(i) = \begin{cases} [-4 \ 2.5 \ 0 \ 0 \ 0 \ 1.5^\circ]^T & \text{if } i \in \langle 1, 3 \dots \rangle \\ [4 \ 2.5 \ 0 \ 0 \ 0 \ -1.5^\circ]^T & \text{if } i \in \langle 2, 4 \dots \rangle \end{cases} \quad (12)$$

Images in the sequences were taken after each step of the robot which resulted in the frame rate of approx. 2 frames per second. The ground truth information was obtained by recording the robot's movement with the fish-eye camera placed under the ceiling. The images were undistorted using the algorithm presented by Baczyk [13] and the robot's position was manually tracked under the assumption that the robot's platform is parallel to the ground (Fig. 3).



Fig. 3. The exemplary view from the ceiling camera before (left) and after (right) undistortion

The first sequence simulated the robot's movement through flat terrain with a smooth reference trajectory. In case of the sequence u_1 the system using the proposed model was able to track the trajectory of the robot despite severe differences between the reference and the measured trajectories (Fig. 4, left column). The 'agile camera' model was not able to follow the exact trajectory, however the final estimate of the position was close to the measured position.

The second trajectory corresponds to the situation when the robot was forced to abruptly change the direction of its movement and started to traverse through the bumpy terrain which caused changes of the elevation. The system using the presented movement model was able to track the position of the moving robot despite its discontinuous movement (Fig. 4 middle column). The system based on the 'agile camera model' was able to track the smooth part of the trajectory but failed after the sudden change of the movement direction.

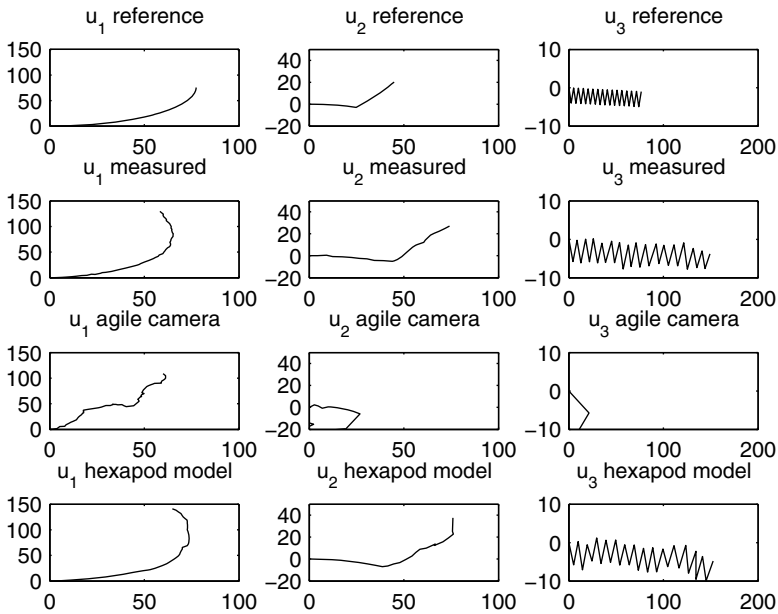


Fig. 4. Robot's trajectories

The third sequence illustrates the problem of the robot scanning its surrounding. The swaying movement increased the effective field of view. Unfortunately, the 'agile camera' was not able to track the movement of the robot and tracking process collapsed after the first step (Fig. 4 right column). Using the presented model allowed for tracking the robot's movement.

4 Conclusions

This paper presents the visual SLAM system using the movement model based on the robot's reference trajectory. The proposed model allows to track complicated and discontinuous trajectories of the robot despite the differences between the observed and the reference trajectory. The presented system requires approx. 2 frames per second to operate, which is desirable in case of the robot's onboard computer with limited resources.

The proposed model was compared with the state-of-the-art movement model used in the MonoSLAM system. The presented movement model proved to be superior in the task of the hexapod robot tracking. The features of the models were highlighted by using exemplary sequences of images gathered from the camera mounted on the hexapod robot.

The future work will focus on combining the proposed and the 'agile camera' models. Additionally, the information gathered from the onboard accelerometers

will be used to develop a hybrid, vision- and acceleration based SLAM system. Moreover, the performance of different feature detectors (e.g Shi-Tomasi, Harris, Star Keypoint, SURF) and descriptors (different variants of the SURF descriptor) will be evaluated.

References

1. Davison, A.J., Murray, D.W.: Simultaneous Localization and Map-Building Using Active Vision. *IEEE Trans. PAMI* 24(7), 865–880 (2002)
2. Sim, R., Elinas, P., Griffin, M., Little, J.J.: Vision-Based SLAM Using the Rao-Blackwellised Particle Filter. In: *Proc. IJCAI Workshop Reasoning with Uncertainty in Robotics* (2005)
3. Baczyk, R.: *Visual Navigation System for a Robot Based on the Modified SLAM Method*, PhD Dissertation, Poznan University of Technology (2007) (in Polish)
4. Gil, A., Reinoso, O., Ballesta, M., Juliá, M.: Multi-robot visual SLAM using a Rao-Blackwellized particle filter. *Robot. Auton. Syst.* 58(1), 68–80 (2010)
5. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Monocular Vision Based SLAM for Mobile Robots. In: *International Conference on Pattern Recognition (ICPR 2006)*, pp. 1027–1031 (2006)
6. Sturm, J., Visser, A.: An appearance-based visual compass for mobile robots. *Robotics and Autonomous Systems* 57(5), 536–545 (2009)
7. Smith, R., Self, M., Cheeseman, P.: Stochastic Map for Uncertain Spatial Relationships. In: *Proc. Fourth Int'l. Symp. Robotics Research*, pp. 467–474 (1987)
8. Davison, A.J., Reid, I., Molton, N., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. PAMI* 29(6), 1052–1067 (2007)
9. Civera, J., Davison, A.J., Montiel, J.M.M.: Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics* 24(5), 932–945 (2008)
10. Belter, D., Walas, K., Skrzypczynski, P.: Autonomous Hexapod Robot – Development of an Improved Mechanical Design and Control System. *Pomiary Automatyka Robotyka* 2, 249–258 (2009) (in Polish)
11. Fabrizio, J., Devars, J.: An Analytical Solution To The Perspective-n-point Problem For Common Planar Camera And For Catadioptric Sensor. *International Journal of Image and Graphics* 8, 135–155 (2008)
12. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
13. Baczyk, R., Skrzypczynski, P.: Mobile Robot Localization by Means of an Overhead Camera. In: *Automation 2001*, pp. 220–229 (2001) (in Polish)

Displacement Calculation of Heart Walls in ECG Sequences Using Level Set Segmentation and B-Spline Free Form Deformations

Andrzej Skalski¹, Paweł Turcza¹, and Tomasz Zieliński²

¹ Department of Measurement and Instrumentation
AGH University of Science and Technology
al. Mickiewicza 30, 30-059, Kraków, Poland
{skalski,turcza}@agh.edu.pl

² Department of Telecommunications AGH University of Science and Technology
al. Mickiewicza 30, 30-059, Kraków, Poland
tzielin@agh.edu.pl

Abstract. In the paper a problem of displacement calculation of the walls of left heart ventricle in echocardiographic (ECG) ultrasound sequences/videos is addressed. A novel method, which is proposed in it, consists of: 1) speckle reduction anisotropic diffusion (SRAD) filtration of ultrasonography (USG) images, 2) segmentation of heart structures in consecutive de-noised frames via active contour without edges method, 3) calculation of left ventricle frame-to-frame deformation vectors by B-Spline Free Form Deformation (FFD) algorithm. Results from method testing on synthetic USG-like and real ECG images are presented in the paper.

1 Introduction

Reports of the World Health Organization state that heart diseases are one of the most often and dangerous. Measurements of intima-media thickness of the common carotid artery in ultrasound B-mode examination and estimation of shape and motion parameters of the left heart ventricle in ultrasound echocardiography (ECG) are recognised as the best methods for evaluation of cardiovascular risk and heart events prediction. Estimation of left ventricle parameters involves its efficient segmentation, features tracking and displacement calculation. These problems represent very active research topics [1,4,7,10]. The special attention is paid to the left ventricle (LV). Malassoitis and Strintzis [10] applied Hough transform for ellipse model to obtain the initial approximation of the LV in the first frame of the sequence. Then active contour model, based on gradient image, is used for estimation of the LV boundary. Finally, estimated results are corrected by observing the shape deformations. The active contour model was also used [1] in conjunction with filtration in domain of discrete wavelet transform coefficients. Dydenko et al. [4] used registration with affine transformation of the prior shape, provided by a medical expert, together with level set segmentation.

Jacob et al. [7] also used a user intervention: a few contours have to be drawn on the data prior to running the tracker in order to learn the nominal deformation parameters and to make placement of the initial contour. Then, dynamic tracker based on Kalman filter was used.

This paper addresses the problem of displacement calculation of the walls of left heart ventricle in ECG ultrasound sequences/videos. A novel three-step method, already introduced in abstract, is proposed and tested in it. Presented results confirm the method efficiency.

2 Methodology

Idea of the proposed method can be summarized as follows: in the first step filtration algorithm for simple image correction is used. Then, walls of heart area are obtained by segmentation algorithm based on level set without edges technique [2]. Finally, segmentation results are exploited to find displacement vectors connecting corresponding pixels of heart walls in two images. Displacement vectors for each pixel are calculated using B-Spline Free Form Deformation [14]. On this basis calculation of mechanical tissue properties as strain, strain rate or local axial tissue velocity is possible [3].

2.1 Pre-processing and Filtering

Strong disturbances (speckles, specular reflections, echoes, shadows) visible in ultrasound images as multiplicative noise makes USG images segmentation a difficult task. Therefore, USG images are usually filtered before further processing. In this operation speckle reduction anisotropic diffusion (SRAD) filters based on Perona-Malik work [12] are frequently used.

$$\frac{\partial I}{\partial t} = \text{div}[c(|\nabla I|)\nabla I] \tag{1}$$

In (1) $\text{div}(\cdot)$ denotes divergence and the diffusivity coefficient $c(\cdot)$ depends on magnitude of image gradient ∇I which is also an edge estimator. If $c(|\nabla I|) \approx 0$ then diffusion stops and (1) represents all-pass filter preserving edges. In turn, $c(|\nabla I|) \approx 1$ leads to isotropic smoothing by means of 2-D Gaussian window. In our algorithm, following the work [5], the control of diffusion process is based on Kuan model [9] instead of the image gradient:

$$\frac{\partial I}{\partial t} = \text{div}[\kappa \nabla I], \quad I(t = 0) = I_0, \quad \kappa = 1 - \frac{1 - C_n^2/C_I^2}{1 + C_n^2} \tag{2}$$

where: $C_n^2 = \sigma_n^2/\bar{n}$ (noise), $C_I^2 = \sigma_I^2/\bar{I}$ (image) and I_0 is an initial image to be denoised.

In homogeneous image regions local standard deviation σ_I^2 is comparable to noise σ_n^2 . Since then $C_n^2 \approx C_I^2$ and $\kappa \approx 1$, (2) performs intra-region smoothing. On the other hand, near the edges $C_n^2 \ll C_I^2$ makes $\kappa \ll 1$ what slows down the diffusion process.

Using first forward and next backward differences denoted as $\delta_x^+ I_{i,j} = I_{i+1,j} - I_{i,j}$, $\delta_x^- I_{i,j} = I_{i,j} - I_{i-1,j}$ one can discretize the right side of the equation (2) as

$$\begin{aligned} \operatorname{div}(\kappa \nabla I)|_{i,j} \approx & \kappa_{i,j}(I_{i+1,j} - I_{i,j}) + \kappa_{i-1,j}(I_{i-1,j} - I_{i,j}) \\ & + \kappa_{i,j}(I_{i,j+1} - I_{i,j}) + \kappa_{i,j-1}(I_{i,j-1} - I_{i,j}) \end{aligned} \quad (3)$$

which leads to the *explicit scheme* having the following form:

$$\begin{aligned} I_{i,j}^{t+\Delta t} = I_{i,j}^t + \Delta t [& \kappa_{i,j}^t (I_{i+1,j}^t - I_{i,j}^t) + \kappa_{i-1,j}^t (I_{i-1,j}^t - I_{i,j}^t) \\ & + \kappa_{i,j}^t (I_{i,j+1}^t - I_{i,j}^t) + \kappa_{i,j-1}^t (I_{i,j-1}^t - I_{i,j}^t)] \end{aligned} \quad (4)$$

The *explicit scheme* is the most straightforward, but computationally inefficient. It requires quite small (less than 0.25 [16]) time step size to assure numerical stability. The other, so called *semi-implicit* scheme, based on (4) is unconditionally stable [5]. In our implementation, we have chosen the Jacobi approach proposed in [5] but instead of central differences as in [5] we applied the mixed image differences (between iterations t and $t + \Delta t$) to obtain additional computational speed-up. The final formula of the *semi-implicit* filter is:

$$I_{i,j}^{t+\Delta t} = \frac{I_{i,j}^t + \Delta t [\kappa_{i,j}^t I_{i+1,j}^t + \kappa_{i-1,j}^t I_{i-1,j}^t + \kappa_{i,j}^t I_{i,j+1}^t + \kappa_{i,j-1}^t I_{i,j-1}^t]}{1 + \Delta t [\kappa_{i,j}^t + \kappa_{i-1,j}^t \kappa_{i,j}^t + \kappa_{i,j-1}^t]} \quad (5)$$

The computational complexity of one time step in the semi-implicit scheme (5) is almost the same as in (4) but in this case Δt can be 10 (and more) times larger which results in 10 times lower number of iterations. This is especially important in processing large echocardiographic ultrasound images for which not only a large amount of pure arithmetic operations is required but also immense time-consuming memory access.

In fig. 1 efficiency of the described above modified SRAD filtration is presented. It is applied to two synthetic USG-like [13] test images from fig. 3 which were used for initial testing of the designed LV segmentation algorithm.

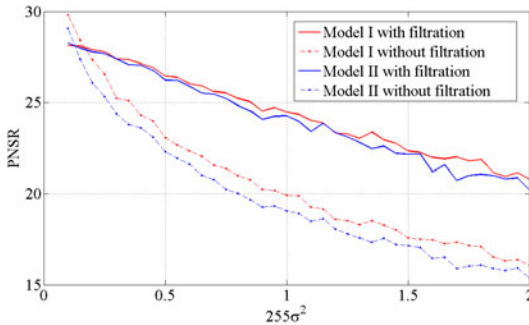


Fig. 1. PSNR of SRAD filtration performance for two artificial USG-like images (models I and II in fig. 3) disturbed by multiplicative speckle noise (as a function of noise level $255\sigma^2$)

2.2 Level Set Based ECG Image Segmentation

In the reported work the *Active Contour without edges algorithm*, proposed by Chan and Vese [2] has been used. The basic idea of classical active contour (*snakes*) approach [8] is to evolve a curve in 2D space, subject to constraints from a given image I . The snake model $\Gamma(s) : [0, 1] \rightarrow R^2$ should minimise the following equation:

$$E_{ac}(\Gamma) = \alpha \int_0^1 |\Gamma'(s)|^2 ds + \beta \int_0^1 |\Gamma''(s)|^2 ds - \lambda \int_0^1 |\nabla I(\Gamma(s))|^2 ds \quad (6)$$

where α, β, λ are positive coefficients.

The first two terms in equation (6) are known as internal energy and they are responsible for smoothness of the contour. The last one, external (image) energy, attracts the contour toward the edges existing in the image. Edges are usually indicated by classical gradient function. In order to remove noise from the image and increase capture range of the snake, the image is usually convolved with a Gaussian kernel before gradient computation. Formulation of the solution to (6) in terms of parametric curve Γ leads to difficulty with topology changing. The problem can be avoided [11] by representing the contour Γ as a zero level set of evolving function $\phi(x, y, t)$. In particular time instance, the contour Γ is defined as:

$$\Gamma = \{(x, y) | \phi(x, y) = 0\} \quad (7)$$

Segmentation of ECG image sequence is a particularly difficult task due to speckle noise and unsharp (not well defined) borders. The first problem has been solved in our approach by speckle reducing anisotropic diffusion (SRAD) filtering described in section 2.1 while the second one - by applying to ECG images a segmentation algorithm which is using information about image intensity values inside and outside the contour instead of the image gradient.

Using level set formulation the problem (6) can be formulated as [2]:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon [\mu \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|}) - v - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2] = 0 \quad (8)$$

where δ_ϵ is a regularised version of δ function. Application of this algorithm, additionally equipped with automatic ROI calculation, to the left ventricle segmentation is described in [15].

2.3 Displacement Calculation Based on Image Registration

The last algorithm step involves computation of displacement vectors for wall pixels obtained by segmentation of consecutive ECG images. Computation is based on B-spline Free Form Deformation (FFD) algorithm, described by Rueckert et al. [14]. As a similarity measure Sum of Squared Differences (SSD) is used. The method [14] makes use of two stage deformation model. In the first step, a rigid (affine) transformation is estimated in order to establish good initial alignment of images - it allows correcting heart position in ECG series. Then,

B-spline FFD algorithm is used to model the local deformation. Therefore, the overall transformation is composed of two parts, *global* and *local*, and can be denoted as:

$$\mathbf{T}(x, y) = \mathbf{T}_{global}(x, y) + \mathbf{T}_{local}(x, y) \quad (9)$$

where $\mathbf{T}_{local}(x, y)$ makes use of B-splines and it is defined as follows:

$$\mathbf{T}_{local}(x, y) = \sum_{m=0}^3 \sum_{n=0}^3 B_m(u) B_n(v) \xi_{i+m, j+n} \quad (10)$$

where ξ is nx -by- ny mesh of control points, $i = \lfloor x/nx \rfloor - 1$, $j = \lfloor y/ny \rfloor - 1$, $u = x/nx - i + 1$ and $v = y/ny - j + 1$. B_m represents the m -th basis function of the B-spline.

In order to receive transformation parameters, the registration process was defined in [14] as an optimization problem:

$$\arg \min_{\mathbf{T}} (-F_{similar}(I_{source}, \mathbf{T}(I_{target})) + \tau F_{smooth}(\mathbf{T}(\cdot))) \quad (11)$$

where τ is weighting parameter, I_{source} denotes a source image with regard to I_{target} , $F_{similar}$, F_{smooth} are a similarity measure and regularisation function, respectively.

In the reported research, the displacement calculation algorithm was implemented in multiresolution manner in order to reduce the computation time and improve the registration accuracy by avoiding local extremes of the cost functions. Exemplary results of displacement vectors calculation, limited to the selected region with heart walls, are presented in fig. 2.

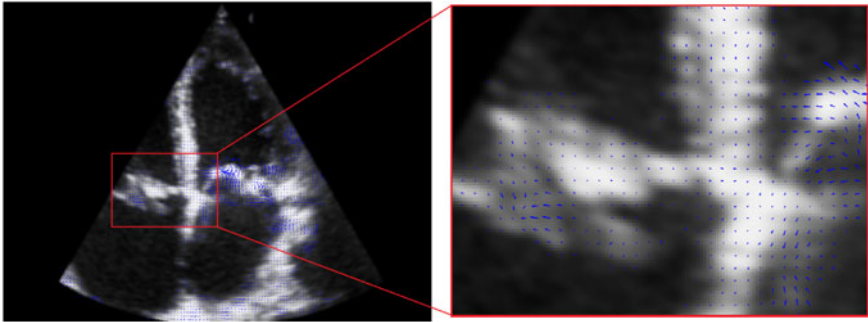


Fig. 2. Example of calculated displacement field between two ECG images limited to segmented area

3 Results and Verification

Modeling USG images is a subject discussed in literature [6, 13]. Typically, embedding the normalised $[0, 1]$ image in the multiplicative speckle noise is done by the following equation (among others implemented also in Matlab):

$$I_{noisy}(i, j) = \max\{0, \min[I(i, j) + \sqrt{12\sigma^2} \cdot I(i, j) \cdot rnd, 1]\} \quad (12)$$

where *rnd* denotes a random variable uniformly distributed in the range [-0.5, 0.5). However, for validation of the proposed method we have implemented in Matlab more precise speckle noise generation based on ultrasound B-mode echographic image acquisition model [13]. It encompasses simulation of the sectoral scan of a plane by an ultrasonic beam and introduces speckle noise in polar coordinates system by means of random walk in the plane of the image complex amplitude, according to the Burckhardt speckle formation taking into account origin of ultrasound echoes generation (see [13] for details). In reported simulations pixel values were normalized in the range [0, 1], $M(x, y)$ number of random walk steps was taken from the interval [1, 20], variance of a phase amplitude was equal [0.1, 0.5, 1.0, 1.3, 1.6, 2]/256, the scanning angle $\Theta = \pi/3$, origin height (offset) $y_0 = 16$ pixels. Images with 256x256 pixels were generated from USG grid pattern consisting of $m = 196$ arcs having $n = 64$ points each.

In figure 4 exemplary segmentation results for real ECG images are presented.

Authors did validation of the applied segmentation algorithm using the generation model of ECG images. In figure 3 two test synthetic USG-like images are shown, the clean (3a, 3d) and noisy one, i.e. embedded in simulated speckle noise (3b, 3e), as well as results of their segmentation (3c, 3f). In order to evaluate a segmentation efficiency, artificial images with different level of noise were used and as a segmentation quality measure a relative number of wrong pixel classifications was chosen:

$$FF = \left(1 - \frac{\sum_{x=1}^N \sum_{y=1}^M |I_{GT}(x, y) - I_S(x, y)|}{\sum_{x=1}^N \sum_{y=1}^M I_{GT}(x, y)} \right) \cdot 100\%, \quad (13)$$

where I_{GT} is a binary mask received from manual expert segmentation and I_S denotes a mask calculated by the proposed segmentation algorithm. Obtained results, presented in figure 5, confirms the method usefulness.

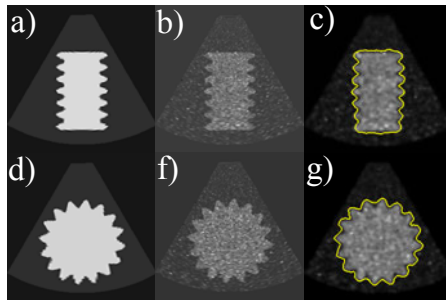


Fig. 3. Exemplary automatic segmentation results for two synthetic USG-like test images; Image/Model I: a) without noise; b) with speckle noise but without SRAD filtration; c) segmentation results after SRAD filtration; The same for test Image/Model II in d) e) f), respectively

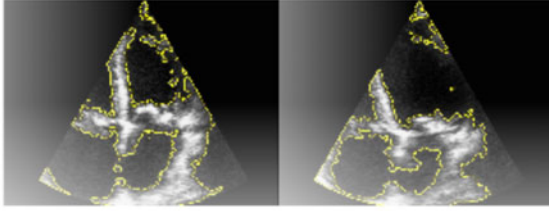


Fig. 4. Exemplary results of walls segmentation from real Echocardiographic images

Table 1. Errors of displacement calculation in pixels

	mean	std	min	max
Manually marked	2.55	1.86	0.0003	8.33
Algorithm	1.74	1.65	0.0001	6.82

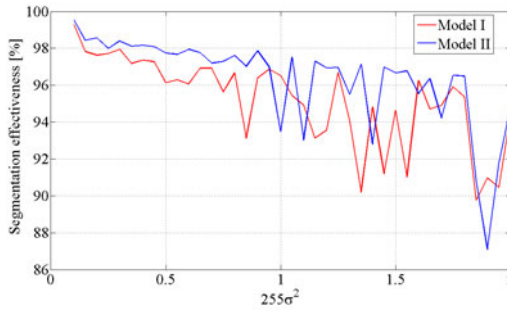


Fig. 5. Automatic segmentation efficiency for generated synthetic USG-like images (Model I and II in fig. 3) as a function of variance of speckle noise phase amplitude [13]

In order to verify displacement calculation, automatic measuring the local registration quality has been compared to 2 human examiners using real ECG data and selecting corresponding feature points. In this experiment 169 pairs of corresponding points have been manually marked. Obtained results, given in table 1, indicate supremacy of the automatic method.

4 Summary

In the paper the novel method for tracking the heart left ventricle movement dynamics in ECG videos is proposed. Obtained results for generated synthetic USG-like images and real ECG ones are promising and encourage the authors to think about future research focused on evaluation of medical characteristics of heart functions making use of the calculated LV motion parameters. The future work will include validation of segmentation process based on experts outline.

Acknowledgements

Authors would like to express their appreciation to the workers of the Department of General Medicine and Geriatrics, Collegium Medicum, Jagiellonian University, Krakow, Poland, specially to Prof. Tomasz Grodzicki, for making available USG images which were used in the research described in this paper. The work was supported by Polish-Italian grant, decision no 363/N-Wlochy/2008/0.

References

1. Bharali, U., Ghosh, D.: Cardiac Motion Estimation from Echocardiographic Image Sequence using Unsupervised Active Contour Tracker. In: Proc. ICARCV (2006)
2. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. on Image Process.* 10(2), 266–277 (2001)
3. D’hooge, J., et al.: Regional Strain and Strain Rate Measurements by Cardiac Ultrasound: Principles, Implementation and Limitations. *Eur. J. Echocardiogr.* 1, 154–170 (2000)
4. Dydenko, I., et al.: A level set framework with a shape and motion prior for segmentation and region tracking in echocardiography. *Med. Image Anal.* 10, 162–177 (2006)
5. Aja-Fernández, S., Lopez, C.A.: On the estimation of the coefficient of variation for anisotropic diffusion speckle filtering. *IEEE Trans. Image Process.* 15(9), 2694–2701 (2006)
6. Gilliam, A.D., Acton, S.T.: Echocardiographic Simulation for Validation of Automated Segmentation Methods. In: *IEEE ICIP 2007*, pp. 529–532 (2007)
7. Jacob, G., et al.: Shape-Space-Based Approach to Tracking Myocardial Borders and Quantifying Regional Left-Ventricular Function Applied in Echocardiography. *IEEE Trans. Med. Imag.* 21(3), 226–238 (2002)
8. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *Int. J. Comp. Vis.* 1, 321–331 (1988)
9. Kuan, D.T., Sawchuk, A.A., Strand, T.C.: Adaptive noise smoothing filter with signal-dependent noise. *IEEE Trans. Pattern Anal. Mach. Intell., PAMI* 7(2), 165–177 (1985)
10. Malassiotis, S., Strintzis, M.G.: Tracking the Left Ventricle in Echocardiographic Images by Learning Heart Dynamics. *IEEE Trans. Med. Imag.* 18(3) (1999)
11. Osher, S., Sethian, J.: Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *J. Comp. Phys.* 79, 12–49 (1988)
12. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI* 12(7), 629–639 (1990)
13. Perreault, C., Auclair-Fortier, M.-F.: Speckle Simulation Based on B-Mode Echocardiographic Image Acquisition Model. In: *4th Canadian Conf. CRV 2007*, pp. 379–386 (2007)
14. Rueckert, D., et al.: Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. *IEEE Trans. Med. Imag.* 18(8), 712–721 (1999)
15. Skalski, A., et al.: Left Ventricle USG image segmentation using Active Contour Model. In: *Int. Conf. on Comput. Sci., Amsterdam* (2010) (accepted)
16. Yu, Y., Acton, S.: Speckle reducing anisotropic diffusion. *IEEE Trans. Image Process.* 11, 1260–1270 (2002)

Fast and Accurate Machined Surface Rendering Using an Octree Model

Joanna Porter-Sobieraj and Maciej Świechowski

Warsaw University of Technology, Faculty of Mathematics and Information Science
Plac Politechniki 1, 00-661 Warsaw, Poland
{j.porter,m.swiechowski}@mini.pw.edu.pl

Abstract. This paper presents a novel method for reconstructing a machined surface from a voxel-based octree model to a polygonal mesh. A compact form of an octree is presented which allows unchanged topology after reconstruction. The algorithm considers an arbitrary shape of the milling cutter. Additional data contained in a node is stored for fast surface normal vector retrieval. Mesh vertices are pushed along the normal vector for more accurate surface approximation. Frustum culling, adaptive levels of detail and local sharp feature detection are included. This method is well suited for implementation on the newest programmable graphics units.

Keywords: 5C Milling Process Visualization, Volume Rendering, Machined Surface Reconstruction.

1 Introduction

The visualization of a milling process in a virtual environment before performing it on real machines has become a vital issue because it reduces both the risk and cost of manufacturing. The task is strongly limited by hardware capabilities, so most methods are developed and optimized for what is available at the time. Of particular interest is five-axis (5C) machining, where complex shapes need to be rendered in real time. The main features, which must be taken into account, are the speed and detail of visualization as well as the ability to reproduce special features such as sharp edges, highlights or transparent parts. The goal is to provide a resolution capable of showing small details and enabling early error verification while maintaining interactive frame rates.

The paper focuses on an algorithm for the visualization of shape changing during 5C cutting simulation. A machined shape decomposed recursively by an octree of voxels is considered. The main problem here was to design an algorithm, which enables fast triangulation of the surface of such a solid while meeting all the aforementioned requirements. To ensure the best fit of the visualized shape to the real one, an additional data structure for the cutter has been proposed. It turns out that the crucial information for storing and rendering the details of the shape of the machined material – in contrary to other existing methods – enables the rendering of the machining simulation with an arbitrary, not only APT-like, shape of the milling tool.

2 Related Work

One of the first methods of virtual machining visualization was described in a paper by Van Hook [1]. An arbitrary viewing-plane is chosen and represented by a set of basic geometric elements called dexels. Emanating from each dexel, a ray is cast orthogonally onto the viewing-plane. The whole scene is covered by rays with a fixed precision. Each dexel stores intersection information between a ray and a visualized part. For rendering purposes, only the nearest intersection is used. The purpose of such a method is to use a hardware Z-buffer to store intersection depth, and a hardware framebuffer – to store the pixel color of an object (if there is an intersection) or the background (if there is not an intersection). Although the whole process must be recalculated when the view changes, which makes it unsuitable for moving the camera interactively during the rendering, this method has been heavily used in commercial CAD/CAM packages and many modifications have been presented.

Saito and Takahashi [2] present a method based upon multiple viewing-planes and a blending formula between all of them, while Lischinski [3] shows how to discretize a 3D object into a set of two-dimensional slices. Various synthesis methods have been shown of how to construct the final image from the available slices.

A similar approach to visualization is performed by ray-casting. One of the first references to ray-casting in virtual machining can be found in Wang and Kaufman [4]. In most cases, ray-casting cannot be used in real-time rendering. Ray-casting gains special benefits in virtual machining when the part in question is represented as a CSG-tree, referenced in the Stewart et al. work [5]. Ray-casting with the use of voxels to represent the part is described in a paper [6] by Yagel et al. Voxels not only define the interior but also store the best, average normal vector of the surface.

There are more methods that can be performed on data stored in voxels. Yau et al. [7] employ a voxel octree and render all the visible cube faces. Many approaches are based on extracting an iso-surface from volume data. The most famous algorithm is called Marching Cubes and comes from Lorensen and Cline [8]. It is a very clean, fast and easy-to-implement method for creating gpu-friendly triangles from a three-dimensional scalar field. The original algorithm has some drawbacks, such as ambiguous cases, the impossibility of reconstructing sharp edges or corners and it sometimes generates more triangles than necessary. Without proper precision, space aliasing can also be noticed. The use of the Marching Cubes algorithm in virtual machining is presented in a variety of papers [9,10,11,12]. This algorithm does not make use of the specific data connected with the machined surface that can be obtained during the simulation. Therefore, surfaces machined with either a toroidal cutter with a small inner radius, or a flat-ended tool, which generates the same octree structure, both look the same.

Inui and Kakio [13] implement the concept of tessellating the workpiece in real time, computing it by subtracting the swept volume of a moving cutter from a solid model representing the stock shape. Regrettably, the swept volume can be generated speedily only for cutters of a simple shape.

3 The Proposed Method

3.1 Input Data

Representing the volume of solid geometry, an octree of voxels has been selected for the method described in this chapter. Each voxel is spatially defined by an axis aligned cube and can contain additional properties. An octree level contains voxels of the same size, beginning with one bounding cube for the root. Each node corresponding to a *grey* voxel (i.e. it intersects the boundary of the solid and cannot be classified as inside or outside the solid) contains up to 8 child voxels half its size. Such an octree is the most effective data structure for storing the voxels of the solid, because it efficiently describes big, homogenous (full or empty) cubes. Hence, the current shape is described by the number of leaves proportional to the solid surface, not to its volume [14]. The presented method of visualizing volume data can be generalized for non-virtual machining purposes. However, for rendering results from machining simulation, it is convenient to store information about the milling cutter and how it interacts with the geometry during the process.

As the rotating cutter is axis-symmetrical, the idea is to store only half its planar profile. The purpose of introducing a separate data structure for the cutter is to use it as a reference space for the voxels and retrieve precise information about the distance from the surface of the solid and the normal vector. The distance from the surface is defined as the nearest distance from the milling cutter in a time domain and a normal vector is defined as opposite to the normal of the cutter surface of revolution at the same moment of simulation. The idea is to discretize the section of the half-plane – defined by the polar and cylindrical axes and containing half the milling cutter profile – and store the scalar distance field between the milling cutter surface and the vector opposite the normal of the nearest point seen in this two-dimensional coordinate system.

The classification of voxels is done by their bounding spheres. Each voxel can have one of three possible states:

1. *White* – the bounding sphere was completely inside the milling cutter during simulation and the voxel was cut out.
2. *Grey* – the bounding sphere was damaged by the cutter. Such voxels may contain the surface of the solid.
3. *Black* – the bounding sphere was never intersected by the milling cutter. Voxel is completely inside the solid.

White voxels are removed from the octree immediately. *Black* voxels are removed during the octree preparation process described in Section 3.2.

The main idea is to have the position (x, y, z) of the voxel center in a three-dimensional tool system along with two angles α and β defining the orientation of the system. The third angle corresponds to the rotation axis and can be omitted because of axis-symmetry. These five scalar values are captured during simulation and stored in a voxel, the moment the distance between the machine tool and the voxel center is at its minimum.

3.2 Octree Preparation

Once a voxel octree is used for the visualization it must be converted to a form that meets a certain condition. Let n -compact voxel be a voxel which has n neighbors (adjacent voxels). The surface reconstruction algorithm requires that any voxel present in an octree must be *grey* or *black* not 1-compact and not 26-compact. As a result, the boundary of any compact solid is enclosed in voxels and this condition generates a very interesting topological property – the whole surface can be traversed in a similar way to moving in a space with the taxicab metric. In other words, every voxel of a solid piece is reachable from any other by making moves only between voxels sharing a common face. Figure 1 illustrates this important property.

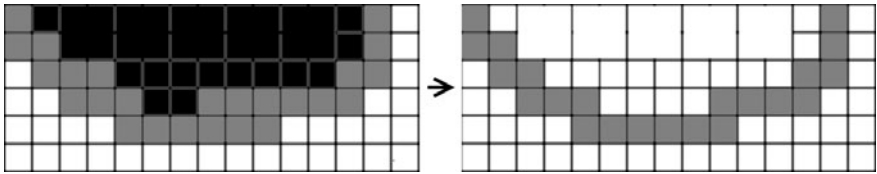


Fig. 1. The format an octree is converted to during the initial step

3.3 Surface Reconstruction

The surface reconstruction method explores the property derived in Section 3.2. The idea is to take every four adjacent voxels meeting at the common edge and create a square spanning their centers as presented in Figure 2. Such a procedure leads to the creation of a compact surface mesh built from squares of three different orientations. This mesh then becomes the basis for further refinement.

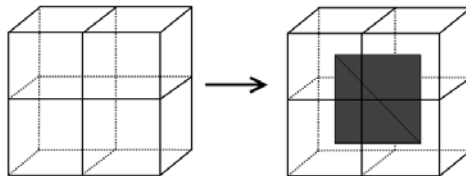


Fig. 2. Creation of polygons

The next step is to extract information about the cutter for each voxel center now associated with a vertex in the mesh. A voxel holds five scalar values (x, y, z, α, β) which define the voxel position \mathbf{P} in a fixed tool coordinate system and tool orientation.

The three coordinates (x, y, z) of the voxel position \mathbf{P} are then transformed to a two-dimensional coordinate system:

$$x' = \sqrt{x^2 + y^2} \quad z' = z \quad (1)$$

where x' represents a radial coordinate (on the radius axis) and z' represents a coordinate on the tool's own axis of rotation.

Distance d to the surface and a two-dimensional normal vector (n_x, n_z) are derived from the tool data structure. The normal becomes three-dimensional as $(n_x, 0, n_z)$. It is then rotated by angle $\gamma = \text{atan2}(y, x)$ along the Z-axis, angle α – along the X-axis and finally β – along the Y-axis. The normal obtained in this way is assigned to a vertex and used in a standard lighting and shading procedure. Vertex position \mathbf{P} is translated by retrieved distance d along normal vector \mathbf{N} :

$$\mathbf{P}' = \mathbf{P} + d\mathbf{N} . \quad (2)$$

3.4 Sharp Features Detection

The surface reconstructed by the above method may appear with sharp (C^0) areas. This is caused either by sharp tool paths or sharp features in the tool itself. In both cases, the points representing triangle vertices approximate the surface correctly, but normal vectors may be wrong. Then trilinear normal vector interpolation performed by graphics hardware will introduce artifacts and hide sharp features of the surface.

The idea is to detect sharp places on a triangle creation level, find a proper normal vector and use it for triangle vertices independently. This means that if a triangle is classified as being located on a sharp feature, any of its vertices will use a different normal vector for each triangle it is common with.

The solution is to compute three dot products for every pair of base normal vectors stored in vertices of a single triangle. If the value of any dot product falls in the $[-0.65, 0.65]$ interval, which is equivalent to a defining angle of at least 49° , a sharp feature is detected. In the next step, two tangent vectors spanning the triangle's edges are taken and their cross product becomes a natural triangle normal vector. Then, the normals stored in vertices are revisited, and the one closest in terms of a dot product with the computed triangle normal is chosen and used by all three vertices in a shading and lighting process.

3.5 Level of Detail

The introduction of levels of detail greatly increases rendering performance. Fortunately, it is easily adapted to an octree data structure. In the presented paper three different levels of detail are suggested but, in general, the method remains the same for any number of levels.

Each level corresponds to voxels of a particular depth in an octree. For example, voxels of levels 9, 8, or 7 in an octree may be used in 1, 2, or 3 levels of

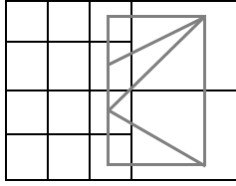


Fig. 3. Modification of a polygon creation method for adjacent different levels of detail

detail, respectively. Voxels of a level less than 7 are only used on a search-path starting from the root and searching for voxels designated for visualization.

To avoid gaps in the geometry, a slightly modified version is applied when creating polygons. Not four but six voxels are tested at the boundaries between different levels of detail. Combining voxel indexing with dictionaries (hashtables) is suggested, rather than the traditional way of storing an octree using memory pointers. Such an approach allows quick topological queries.

Culling invisible voxels is decided by testing their bounding spheres against a current camera viewing frustum. This can be performed either recursively, starting from the root or on a fixed octree level.

4 Results

The designed method has been tested against existing methods to evaluate both its visual and performance qualities. One of the major improvements of the presented algorithm is its use of a new, unique data structure which allows a flexible milling tool description of any arbitrary shape. In such cases, the quality of the reproduced surface is superior to the results obtained from previous methods, that do not make use of the tool shape or else just restrict it to a few common cutter shapes.

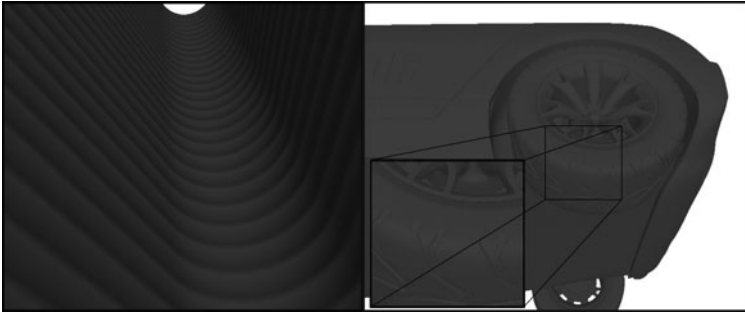
Another noteworthy aspect of the rendering algorithm is its integration with the simulation module. The octree used during the rendering process is interactively built upon the octree from the simulation, which can be of a higher precision. The precision can be arbitrarily high if required (e.g. for verification purposes) since the conversion process is completely dynamic. The only real limit is the available system memory. In contrast, most known solutions operate on fixed precision data structures (like octrees of limited depth) in the visualization module.

The goal of the experiment was to prove whether interactive frame rates can be achieved. Two general phases (*update* and *render*) can be distinguished which have an impact on the overall algorithm performance. The first phase includes octree preparation, visibility tests and triangulation. The second phase is the rendering of the final image on the screen.

A few tens of floating point operations are required to update one voxel so the cost is practically nothing for current processors. On a testing machine the update of 200 voxels lasted 1 ms. The number of voxels updated during one

Table 1. Performance tests on sample data. Performed on AMD Athlon 4800+, NVIDIA GeForce 8800GT and 1.5GB RAM with 1024x768 resolution.

Octree level	Leaf voxels	Triangles	Triangles per voxel	Fps	Triangulation time for the entire octree (sec)
8	48,832	99,872	2.05	1,053	0.243
8	110,608	219,680	1.99	580	0.546
9	278,048	556,744	2.00	240	1.119

**Fig. 4.** Visualizing the details of a machined surface stored in an 8-level octree (*left*) and a complex triangulated model (*right*) converted from a 10-level octree

frame depends on the chosen precision and is strictly related to the tool/voxel size ratio, but in real applications shouldn't be greater than 500. Statistical results show that the impact of the local update on performance is negligible.

The rendering phase can be reduced to a common problem of visualizing a triangle mesh using graphic programmable units. At present, over one million triangles can be rendered in a fully interactive way. Statistics show that this corresponds to trees with approximately half a million voxels.

5 Conclusion

A new approach to the visualization of volume data in virtual machining has been shown. The algorithm employs a separate data structure for storing information about the arbitrary machining tool and uses it to better approximate the polygonal surface and normal vector field. It guarantees the proper reconstruction of sharp shapes and the excellent rendering of details without increasing the depth of the octree. The developed algorithm produces a low average number of triangles; tests for various shapes prove that the average number of triangles per voxel is 2.0. It features frustum culling and levels of detail. It also enables future enhancements – easy implementation for parallel environments and is suitable for real time rendering.

References

1. Van Hook, T.: Real-time Shaded NC Milling Display. *ACM SIGGRAPH Computer Graphics* 20(4), 15–20 (1986)
2. Saito, T., Takahashi, T.: NC Machining with G-buffer Method. *ACM SIGGRAPH Computer Graphics* 25(4), 207–216 (1991)
3. Lischinski, D., Rappoport, A.: Image-Based Rendering for Non-Diffuse Synthetic Scenes. In: *9th Eurographics Workshop on Rendering*, pp. 301–314. Springer, Wien (1998)
4. Wang, S.W., Kaufman, A.E.: Volume Sculpting. In: *Symposium on Interactive 3D Graphics*, pp. 151–156. ACM, New York (1995)
5. Stewart, N., Leach, G., John, S.: An Improved Z-Buffer CSG Rendering Algorithm. In: *ACM SIGGRAPH Workshop on Graphics Hardware*, pp. 25–30. ACM, New York (1998)
6. Yagel, R., Cohen, D., Kaufman, A.: Discrete Ray Tracing. *IEEE Computer Graphics and Applications* 12(5), 19–28 (1992)
7. Yau, H.T., Tsou, L.S., Tong, Y.C.: Adaptive NC Simulation for Multi-Axis Solid Machining. *Computer Aided Design and Applications* 2(1-4), 95–104 (2005)
8. Lorensen, W.E., Cline, H.E.: Marching cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics* 21(4), 163–169 (1987)
9. Kim, Y.J., Varadhan, G., Lin, M.C., Manocha, D.: Fast Swept Volume Approximation of Complex Polyhedral Models. In: *ACM Symposium on Solid and Physical Modeling*, pp. 11–22. ACM, New York (2003)
10. Kobbelt, L.P., Botsch, M., Schwanecke, U., Seidel, H.P.: Feature Sensitive Surface Extraction from Volume Data. In: *28th International Conference on Computer Graphics and Interactive Techniques*, pp. 57–66. ACM, New York (2001)
11. Schaefer, S., Warren, J.: Dual Marching Cubes: Primal Contouring of Dual Grids. In: *12th Pacific Conference on Computer Graphics and Application*, pp. 70–76. IEEE Computer Society, Washington (2004)
12. Nielson, G., Hamann, B.: The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In: *2nd Conference on Visualization*, pp. 83–91. IEEE Computer Society Press, Los Alamitos (1991)
13. Inui, M., Kakio, R.: Fast Visualization of NC Milling Result Using Graphics Acceleration Hardware. In: *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3089–3094. IEEE Press, New York (2000)
14. Porter-Sobieraj, J., Marciniak, K.: Design and Exploitation of a Simulation Module for Injection Moulds Machining. In: Skolud, B., Krenczyk, D. (eds.) *Computer Integrated Manufacturing*, pp. 449–454. Wydawnictwa Naukowo-Techniczne, Warsaw (2003)

PATSI — Photo Annotation through Finding Similar Images with Multivariate Gaussian Models*

Michał Stanek, Bartosz Broda, and Halina Kwasnicka

Wrocław University of Technology, Institute of Informatics
{michal.stanek,bartosz.broda,halina.kwasnicka}@pwr.wroc.pl

Abstract. Automatic Image Annotation is important research topic in machine vision as it enables one to retrieve images from large databases by using textual queries. In recent years many machine learning techniques have been proposed to build detectors of concepts present on the images. In this paper we present a novel approach for image auto-annotation based on transfer of annotations from most similar images to the query image. We model image features by Multivariate Gaussian Distribution and measure distance between images by using Jensen-Shannon divergence. In spite of its simplicity, the proposed solution outperforms the state-of-the-art methods for image annotation and thus can be used as a baseline for developing other more elaborate methods.

1 Introduction

Traditional search engines were concern with only textual data for providing information that the user was looking for. Nowadays we can observe a trend, where also other modalities are becoming important in retrieval tasks. There is an enormous amount of visual data available on the Internet, as well as in off-line image databases. Automatic analysis of visual data is not a trivial task.

Traditional, so called Text-Based Image Retrieval (*TBIR*), dealt with this problem by retrieving images using textual information that is available in the same document as the target image. This approach suffers from small correlation between textual description and visual data[1]. On the other hand, not every image database has sufficiently rich meta-data describing images to perform efficient retrieval. In contrast to *TBIR* the *Content-Based Image Retrieval (CBIR)* deals only with content-based visual cues. For this purpose one could use manually created semantic labels (tags, annotations) for image retrieval. The process of labelling is a tedious, costly and error-prone. There is a clear need for an automatic method for labelling images. Thus the goal of automatic image annotation is to assign semantic labels for images. Assigned labels can be then used in several ways, most notably in search engines.

* This work is partially financed from the Ministry of Science and Higher Education Republic of Poland resources in 2008–2010 years as a Poland–Singapore joint research project 65/N-SINGAPORE/2007/0.

There are many reasons why automatic image annotation is a difficult task. We name just a few among them. The number of classes is usually very large (in other words: size of labels dictionary W is large). Available training data is often *weakly annotated*, i.e., annotations are often incomplete and may contain errors [2]. Last but not least, there is no direct correspondence between visual features and semantic labels.

There has been a plethora of studies on automatic image annotation utilizing machine learning techniques for learning statistical models from annotated images and apply them to generate annotations for unseen images. Most of the state of the art approaches can be classified into two categories, i.e., probabilistic modelling methods and classification methods.

Among the first category we name just a few especially interesting methods: Hierarchical Probabilistic Mixture Model (HPMM) [3], Translation Model (TM) [4], Supervised Multi-class Labelling (SML) [2], Continuous Relevance Model (CRM) [5], and Multiple Bernoulli Relevance Models (MBRM) [6].

CRM method is based on Bayes theorem and uses non-parametric approach. Parzen estimator is used combined with one dimensional Gaussian kernel for density estimation. MBRM is an extension on CRM based on Bernoulli Relevance Models, which outperforms other methods as reported by the authors [6]. Those methods were used as a reference baselines by many researchers working on the problem of image annotation [2, 14].

The methods of second category try to find correlation between words and visual features by training classifiers. Bayes Point Machine [7], Support Vector Machine [8] and Decision Trees [9], estimating the visual feature distributions associated with each word [10].

There are also methods that try to improve output of other image annotation methods. GRWCO [9], can be used to improve average recall and precision [11] of automatic annotator, by reducing the difference between expected and resulted word count vectors. Annotation refinement can be also achieved by using Word-Net which contains semantic relations between words [12]. The word co-occurrence models coupled with fast random walks are used in IRWR [13] for re-ranking the output annotations.

Recently, Makadia et. al. [14] proposed a family of baseline methods that are build on the hypothesis that visually similar images are likely to share the same annotations. They treat image annotation as a process of transferring labels from nearest neighbours. Makadia's method, does not solve the fundamental problem of determining the number of annotations that should be assigned to the target image. Thus they assume a constant number of annotations per image. The transfer is performed in two steps: all annotations from the most similar image are rewritten and the most frequent words are chosen from the whole neighbourhood until a given annotation length has been achieved. They also combine many similarity measures to obtain the subset of most similar images.

In this paper we propose a simple method for Photo Annotation through Finding Similar Images (**PATSI**) based on the hypothesis that similar images should share a large part of the annotations. High accuracy obtained by proposed

method on the standard benchmarking image datasets in conjunction with the simplicity of the method and its computational efficiency makes it a perfect candidate for being a baseline in the field of automatic images annotation. Proposed method solves also a difficult problem of choosing the appropriate number of annotations assigned for the target images.

This article is organized as follows. In the next section we describe proposed method, with particular emphasis on the transfer function, similarity criterion and used feature sets. Following section describes the experiments and achieved results. The paper is finished with conclusions and remarks on possible further improvements of the method.

2 Photo Annotation Through Finding Similar Images

In this section we describe how to create models of images, in order to compute similarities between them. Next, we describe a method for transferring annotations from similar images to query image and how to determine the annotation length for the query image.

In an automatic image annotation process, annotator \mathcal{A} describe previously unseen image \mathcal{I} by a set of concepts $W^{\mathcal{I}}$ from the semantic dictionary \mathcal{W} based on the training dataset D , containing image–words pairs ($D = \{(\mathcal{I}_1, W^{\mathcal{I}_1}), \dots, (\mathcal{I}_M, W^{\mathcal{I}_M})\}$).

In Photo Annotation through Finding Similar Images (*PATSI*) approach, for a query image \mathcal{I} , a vector of the most similar images from the training dataset D need to be found. The image \mathcal{I} in a training dataset D is represented by a n -dimensional vector of visual features $v^{\mathcal{I}} = [v_1^{\mathcal{I}}, \dots, v_n^{\mathcal{I}}]$. All visual features are a m -dimensional vector of low level attributes $v_i^{\mathcal{I}} = [x_1^{i,\mathcal{I}}, \dots, x_m^{i,\mathcal{I}}]$. The visual feature represent statistical information about color and texture in selected area of the image \mathcal{I} . We assume that the image is split into disjoint areas by a rectangular grid, but any other method can be employed for this problem.

Based on the visual feature vectors $v^{\mathcal{I}}$ we build a model $M^{\mathcal{I}}$ for the image \mathcal{I} . We assume that the $M^{\mathcal{I}}$ is multi-dimensional random variable described by multi-variate normal distribution and all vectors $v_i^{\mathcal{I}}$ are realizations of this model. Probability density function (*PDF*) for the model $M^{\mathcal{I}}$ is defined as:

$$M^{\mathcal{I}}(x, \mu, \Sigma) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \quad (1)$$

where x is an observation vector, μ is mean vector, and Σ is the covariance matrix. Both μ and Σ are parameters of the model calculated using Expectation-Maximization algorithm [15] on all visual features $v^{\mathcal{I}} = [v_1^{\mathcal{I}}, \dots, v_n^{\mathcal{I}}]$ of the image \mathcal{I} . In order to avoid problems of inverting covariance matrix (avoid matrix singularity) one may perform regularization of the covariance matrix. Models of images are build for all images in the training set, as well as for the query image.

To measure the distance between the models we propose to use Jensen–Shannon divergence, which is a symmetrized version of Kullback–Leibler divergence:

$$D_{JS}(M^{\mathcal{A}} \| M^{\mathcal{B}}) = \frac{1}{2} D_{KL}(M^{\mathcal{A}} \| M^{\mathcal{B}}) + \frac{1}{2} D_{KL}(M^{\mathcal{B}} \| M^{\mathcal{A}}), \quad (2)$$

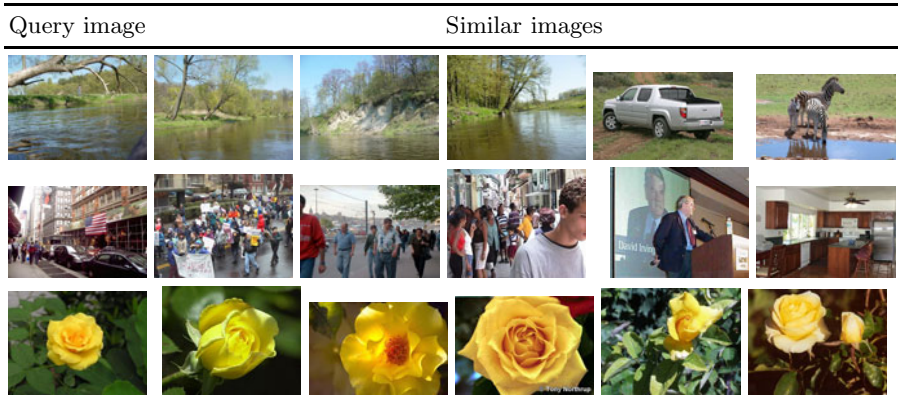
where M^A , M^B are models (PDF) for image A and B , D_{KL} is Kullback-Leibler distance which for multivariate-normal distribution takes the form:

$$D_{\text{KL}}(\mathcal{M}^A \parallel \mathcal{M}^B) = \frac{1}{2} \log_e \left(\frac{\det \Sigma_B}{\det \Sigma_A} \right) + \frac{1}{2} \text{tr} (\Sigma_B^{-1} \Sigma_A) + \frac{1}{2} (\mu_B - \mu_A)^\top \Sigma_B^{-1} (\mu_B - \mu_A) - \frac{N}{2}, \quad (3)$$

where Σ_A , Σ_B and μ_A , μ_B are covariance matrices and mean vectors from respectively image model A and B.

In order to annotate the query image \mathcal{Q} , one must find the model $M^{\mathcal{Q}}$ of that image, based on eq. 1. Then a list of K nearest models from training dataset D is created based on similarity distance measure (eq. 2). Table 1 presents an example subset of most similar images to arbitrary selected query image on MGCV 2006 dataset [16].

Table 1. Example of the most similar images to query image on MGCV 2006 dataset



Let $M^i, i = 1, \dots, K$, be the ranking of the nearest images models ordered by increase distance measure. Based on the hypothesis that images similar in appearance are likely to share the same annotation words we transfer keywords from nearest neighbours to the query image. All labels for the image on position r in the ranking are transferred with value $1/r$. This approach allows for labels from more similar images to have a larger impact on resulting annotation. The resulting query image annotations consists of all the words whose transfer value was greater than a specified threshold t . The threshold value t has an impact on annotation length and its optimal value for given dataset can be found with standard optimization methods.

The outline of the PATSI method can be then summarized by the following two phases:

1. Preparation phase:
 - (a) All images in training database are divided into disjoint regions.

- (b) For all regions statistical visual features are calculated.
 - (c) From visual features image models are created.
2. Query phase:
- (a) Divide query image into disjoint regions and calculate visual features vector for them
 - (b) Build query image model from its visual features.
 - (c) Calculate distance from query image model to all images models in training database.
 - (d) Take K images with smallest distances between models and create ranking of those images.
 - (e) Transfer all words from images in the ranking with the value $1/r$, where r is the position of the image in the ranking.
 - (f) As a final annotation take words which sum of the transfer values are greater or equal to provided threshold t value.

3 Experiments

In this section we present experimental evaluation of proposed method. We performed experiments on the three benchmarking data sets: ICPR 2004 [17], MGV 2006 [16] and IAPR TC-12 [18] whose characteristics are shown in Table 2.

Table 2. Properties of benchmark datasets

	MGV 2006	ICPR 2004	IAPR TC-12
Number of images	751	1 109	19 805
Dictionary size	74	407	291
Mean annotation length	5.0	5.79	5.72
Mediana of annotation length	5.0	5.0	5.0
Std. dev. of annotation length	1.28	3.48	2.56
Min. and max. annotation length	(2, 9)	(1,23)	(1,23)

For evaluation purposes we use three measures: *precision*, *recall* and *F-Score*. Precision of annotation determines how often the word w in the annotated images collection was used correctly. I.e., it is a ratio of correct occurrences of word w to all occurrences of word w . Precision is usually supplemented with recall — a measure that indicates how many images, which should be annotated with the word w has been annotated correctly by this word. The higher the precision and recall the better. Usually both measures are combining together using F-score, i.e., a harmonic mean of precision and recall.

For all the datasets images were split onto 400 rectangular regions (20 by 20 grid splitter was used). Visual feature vectors for all regions consist of the mean value of colours Red, Green and Blue, standard deviation of these values, number of edges in all RGB color channels, and the three eigenvalues of color Hessian computed in RGB color space.

Table 3. Overview of image annotation algorithms with their performance evaluated on a benchmark image databases

Method	Precision	Recall	F-Score
ICPR 2004 – all			
FastDIM	0.20	0.17	0.18
FastDIM + GRWCO	0.21	0.21	0.21
MCML	0.21	0.17	0.19
MCML + GRWCO	0.25	0.28	0.26
CRM	0.24	0.24	0.24
PATSI	0.27	0.34	0.30
ICPR 2004 – the best 60 words			
FastDIM	0.64	0.58	0.61
FastDIM + GRWCO	0.63	0.61	0.62
MCML	0.69	0.60	0.64
MCML + GRWCO	0.69	0.67	0.68
CRM	0.61	0.61	0.61
PATSI	0.82	0.94	0.88
MGV 2006 – all			
FastDIM	0.24	0.16	0.19
FastDIM + GRWCO	0.34	0.34	0.34
MCML	0.32	0.24	0.27
MCML + GRWCO	0.38	0.37	0.37
CRM	0.39	0.34	0.36
PATSI	0.38	0.46	0.42
MGV 2006 – the best 20 words			
FastDIM	0.58	0.53	0.51
FastDIM + GRWCO	0.59	0.61	0.60
MCML	0.61	0.59	0.60
MCML + GRWCO	0.64	0.62	0.63
CRM	0.58	0.57	0.57
PATSI	0.71	0.86	0.78
IAPR TC 12			
RGB	0.24	0.24	0.24
HSV	0.20	0.20	0.20
LAB	0.24	0.25	0.24
Haar	0.20	0.11	0.14
HaarQ	0.19	0.16	0.17
Gabor	0.15	0.15	0.15
GaborQ	0.08	0.09	0.08
MBRM	0.24	0.23	0.23
Lasso	0.28	0.29	0.28
JEC	0.28	0.29	0.28
PATSI	0.26	0.31	0.28

The performance of PATSI method in comparison with other state-of-the-art method are summarized in Table 3. In all experiments we take into account $K = 23$ most similar images and set threshold value of $t = 0.85$. To evaluate PATSI we use leave-one-out cross-validation method.

For MGV and ICPR dataset as a reference point we have taken the results presented in [9]. For these data sets the proposed method achieved significantly better results. Highest difference is visible for the best annotated words, where F-Score was improved by 20 percentage points in both sets (the relative improvement over CRM is about 37% and 44% for MGV and ICPR dataset respectively).

For IAPR TC 12 for a reference point we have taken the results presented in Makadia et. al. article [14]. We obtain comparable results to Lasso and JEC method on that benchmark set. Lasso and JEC, used the idea of transferring annotation from similar images, but both on those methods combine seven different similarity measures such as RGB, HSV, LAB, Haar, HaarQ, Gabor, GaborQ. Only combination of those measured gives comparable results to PATSI. Makadia et. al. method can not automatically determine the length of the annotation, assuming that this is one of the given parameters.

4 Conclusions and Further Works

In this paper, we proposed automatic image annotation method based on the hypothesis that images similar in appearance are likely to share the same annotations. Proposed method is efficient (in terms of both accuracy and small computational complexity) even for very basic grid image segmentation and feature sets. PATSI is similar to work of [14] in terms of main idea and basic hypothesis used, but differs in a few important ways. First, we have improved a method for transferring annotations to query image. We use simpler model for both computing similarities between images and for feature extraction from images. Last but not least, PATSI is able to determine automatically the number of annotations that should be transferred to the query image. We think that those properties makes PATSI a better baseline method then the work of [14].

In further work, we have to analyse method performance with different feature sets. We also would like to explore the possibilities of automatic method for optimization of t parameter and number of neighbours taking into account and its impact on quality of annotation process.

A very important advantage of using PATSI method is a high recall measure obtained on all datasets, which indicates that the final annotations can be further improved by the wrapper methods [9, 12, 13].

References

- [1] Goodrum, A.: Image information retrieval: An overview of current research. *Informing Science* 3 (2000)
- [2] Carneiro, G., Chan, A., Moreno, P., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(3), 394–410 (2007)

- [3] Hironobu, Y.M., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: Boltzmann machines, Neural Networks, vol. 4 (1999)
- [4] Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
- [5] Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures
- [6] Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1002–1009 (2004)
- [7] Chang, E., Goh, K., Sychay, G., Wu, G.: Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. IEEE Transactions on Circuits and Systems for Video Technology 13(1), 26–38 (2003)
- [8] Cusano, C., Ciocca, G., Schettini, R.: Image annotation using svm. In: Proceedings of SPIE, vol. 5304, pp. 330–338 (2004)
- [9] Kwasnicka, H., Paradowski, M.: Resulted word counts optimization-a new approach for better automatic image annotation. Pattern Recogn. 41(12) (2008)
- [10] Carneiro, G., Vasconcelos, N.: A database centric view of semantic image annotation and retrieval. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 559–566. ACM, New York (2005)
- [11] Kwasnicka, H., Paradowski, M.: Multiple class machine learning approach for an image auto-annotation problem. In: ISDA 2006: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Washington, DC, USA, pp. 347–352. IEEE Computer Society, Los Alamitos (2006)
- [12] Jin, Y., Khan, L., Wang, L., Awad, M.: Image annotations by combining multiple evidence & wordnet. In: MULTIMEDIA 2005: Proceedings of the 13th annual ACM international conference on Multimedia. ACM, New York (2005)
- [13] Llorente, A., Motta, E., Rüger, S.: Image annotation refinement using web-based keyword correlation. In: Chua, T.-S., Kompatsiaris, Y., Merialdo, B., Haas, W., Thallinger, G., Bailer, W. (eds.) SAMT 2009. LNCS, vol. 5887, pp. 188–191. Springer, Heidelberg (2009)
- [14] Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
- [15] McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, 2nd edn. Wiley Series in Probability and Statistics. Wiley-Interscience, Hoboken (March 2008)
- [16] Paradowski, M.: Metody automatycznej anotacji jako wydajne narzędzie opisujące kolekcje obrazów. PhD thesis, Wrocław University of Technology (2008)
- [17] Ipr 2004 image database (2004), <http://www.cs.washington.edu/research/>
- [18] Grubinger, M., Clough, P.D., Henning, M., Thomas, D.: The iapr benchmark: A new evaluation resource for visual information systems. In: International Conference on Language Resources and Evaluation, Genoa, Italy (May 2006)

Generation of Temporally Consistent Depth Maps Using Noise Removal from Video

Olgierd Stankiewicz and Krzysztof Wegner*

Chair of Multimedia Telecommunications and Microelectronics
Poznań University of Technology, Polanka 3, 60-965 Poznań, Poland
{ostank,kwegner}@multimedia.edu.pl
www.multimedia.edu.pl

Abstract. This paper presents a novel approach for providing depth maps that are temporally consistent. Temporal consistency is attained by noise removal from video. Presented approach was evaluated with use of a simple noise reduction technique and state-of-the-art depth estimation algorithm. Experiments on depth-based synthesis of standard multi-view test video sequences have been performed and yielded both subjective and objective results. These results provide evidence that the proposed approach increase temporal consistency of estimated depth maps.

Keywords: depth map estimation, temporal consistency, temporal noise removal.

1 Introduction

Depth map estimation is a technology that provides 3D representation of the scene [1]. Common approach to obtaining depth data is algorithmic estimation from video. Although many such algorithms are known in literature [2], depth estimation is still a challenge, even for the most advanced state-of-the-art techniques. One of the biggest of challenges in this research area, is how to provide depth maps that are consistent in time. Typically, depth data for video is estimated independently for each frame of the sequence. Unfortunately, estimation that is independent in time, causes depth of objects in the scene to fluctuate, due to noise. Such fluctuations are adverse, because they lead to occurrence of artificial movement in 3D representation. Desired depth map temporal consistency means that changes of the depth of objects in time are correlated with actual motion of the objects and do not vary from frame to frame in a random way.

Majority of state-of-the-art techniques that tackle temporal consistency, in various ways expand depth estimation algorithms into time domain. For example, in [3] authors propose to extend standard 4-neighborhood belief propagation depth map estimation scheme [4] to 6-neighborhood scheme by addition of

* This work was supported by the public funds.

temporal neighbors: from previous and from next frame. These neighbors are obtained by motion estimation. Therefore, depth value is optimized with respect to depth value in neighboring frames. In turn, authors of [5] propose segment-based approach. In order to provide temporally consistent depth value, apart from traditionally used spatial matching of segments, also temporal segment matching is performed. Such approach increase complexity of the whole depth estimation process, which already is computationally expensive.

We propose a novel approach to problem of temporal consistency. To tackle temporal inconsistency we propose to eliminate its cause. As mentioned before, depth map fluctuations are caused by noise, mainly temporal. We propose to employ noise reduction on video before depth estimation. As we show later, depth maps obtained in such way are more consistent in time.

Noise reduction is a well-known and widely recognized technical field. Wide variety of examples of noise reduction techniques can be found in [6,7,8]. Classical noise reduction techniques aim to provide a denoised image directly to the audience. In case of depth estimation, more artifacts are allowed, because denoised version of the image is only to be used for depth estimation. Thus, wider range of techniques can be considered.

Presented approach is a new general idea, because it can be applied to any depth estimation technique and any noise reduction technique without any modifications. Although that, for the sake of this paper we chose a fixed setup of these algorithms. We use depth estimation algorithm [9] that is being used as a reference for standardization of 3D television by ISO/IEC MPEG group. For noise reduction we use our simple denoising algorithm, presented in Section 3.

Our noise reduction technique employs temporal filtering and focuses on regions where it applies best to steady regions of the sequence. These are the regions, where the most of inconsistency in depth data occurs.

2 Idea of the Paper

This paper aims at production of better and more temporally consistent depth maps. The main idea consists in application of a temporal noise reduction technique before depth estimation algorithm. Each view of a multi-view video sequence, is independently denoised in time and then feed to a depth estimation algorithm (Fig. 1).

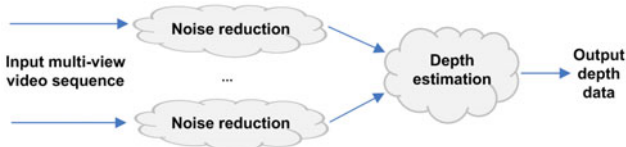


Fig. 1. Idea of the proposed approach. Any noise reduction and any depth estimation techniques can be used.

3 Noise Reduction Technique

For evaluation of the presented approach, we have employed a simple temporal noise reduction technique. Our method consists of three main phases (Fig. 2):

- Motion detection, where pixels are classified as moving or steady,
- Noise filtering, where steady pixels are filtered in time, and
- Artifact removal, where errors of motion detection stage are repaired.

Moving pixels are left unchanged during the entire processing. This is motivated by fact that there is uncertainty of whether motion cues (generated by Motion Detector) are caused by noise or by motion itself. Moreover, temporal filtering applies best to steady regions because in such case there is no need for computationally consumptive motion estimation and compensation, which we prefer to avoid.

All phases of the algorithm are performed in three pipelines of frames: original frames (input of the algorithm), binary motion maps (by-product of the algorithm) and denoised frames (result of the algorithm).

3.1 Motion Detection

The role of Motion Detector is to classify pixels from input frame as moving or as steady. Result of this classification is combined into a binary map called by us motion map.

Our motion detection algorithm is presented in Fig. 3. Each pixel of input frame is compared, by means of absolute differences, with corresponding pixels of N previous frames, stored in Motion Detection Buffer, by means of absolute differences. These absolute differences are then maximized between frames and over RGB color components. Resulting maximum absolute differences are feed to two parallel paths: top and bottom (Fig. 3). These paths, give cues about motion that occur in neighboring pixels, for each pixel. Top path give map of cues about maximal motion, and bottom path gives map of cues about distributed motion. Maximal motion cue map (top path) is obtained with use of dilation filter. Dilation is performed in rectangular window (of size 9×9 in experiments).

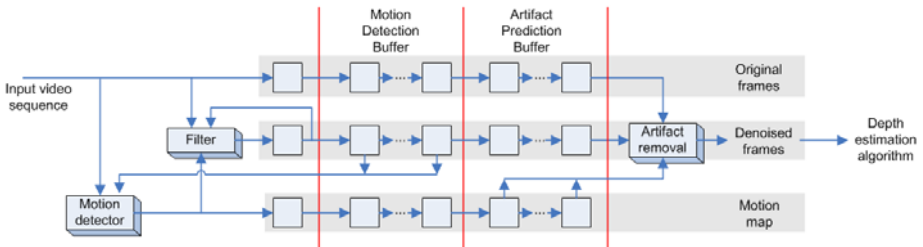


Fig. 2. Block scheme of the algorithm

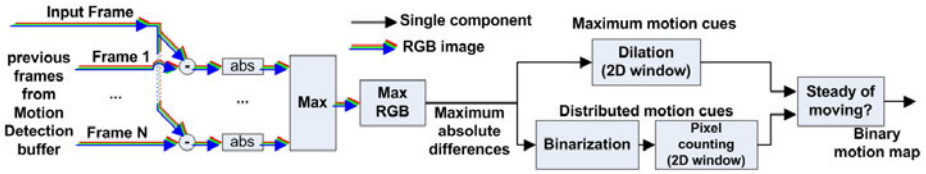


Fig. 3. Scheme of Motion Detector



Fig. 4. Motion map (right) obtained for exemplary frame (left) (white pixels moving, black pixels steady)

Distributed motion cue map (bottom path) is obtained by counting of pixels that exceed certain level (binarization) in window surrounding each pixel (window size is the same as in the top path).

Output of the motion detection, binary motion map (Fig. 4), is produced by combining of motion cue maps from top and bottom path. Pixel is marked as moving (white) if any of motion cues indicate movement (exceed certain level). Otherwise, pixel is marked as steady (black).

All arbitrary parameters, like window size and threshold levels depend on image resolution and noise intensity. These were optimized for experiments empirically.

3.2 Filtering

As mentioned before, pixels classified as moving are left unchanged and are not modified by the algorithm. Pixels classified as steady are independently filtered in time (Fig. 5a) with respect to previously filtered frame, stored in Denoised pipeline of the algorithm (Fig. 1). The idea behind this is to freeze the noise on steady pixels, so that the depth estimation is not confused with fast varying pixel values.

In our work we have exploited simple low-pass IIR (Infinite Impulse Response) filter of first-order (Fig. 5b). Low order filter was chosen to reduce computational complexity and to allow slight changes in the scene (e.g. day-time lighting).

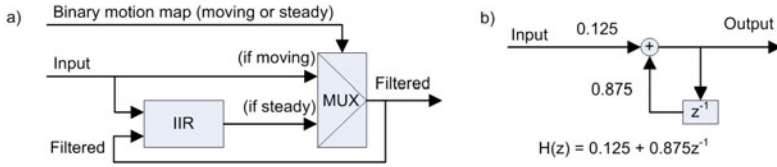


Fig. 5. Filtering scheme (a) and used low-pass IIR filter (b)

3.3 Artefact Rremoval

Noise removal scheme composed of presented Motion Detection and Filtering modules is simple and computationally efficient. Unfortunately, it may be a cause of artifacts resulting from hard-decisive classification of pixels as steady or moving.

Fig. 6 shows three trajectories of exemplary pixel: original value (a), filtered value (b) and value after artifact removal (c). At the beginning (column I), the pixel is classified as steady. It varies due to noise, which is filtered (filtered (b) is the same as (c)). Then (II), pixel value starts to change significantly and is classified as moving. As a result of that, filtering phase is omitted: (a), (b) and (c) are the same. Up to this moment, there are no artifacts. In column III, pixel is classified as steady, because its value changes very slowly. Filtered pixel trajectory changes even slower, resulting in discrepancy between trajectories (Fig. 6), which is lesser than threshold of motion detector. After a while, the discrepancy rise beyond threshold and pixel is instantaneously classified as moving in column IV. Filtering switches off, and thus trajectories are updated to original, which causes another steady column V. Rapid switching causes visual artifact in the output image.

To tackle that, we introduce an artificial removal step. Rapid changes of pixel classification (steady or moving) are predicted, with use of artifact prediction buffer (Fig. 2). If such rapid change is predicted, pixel value is linearly interpolated (Fig. 6c) between original (a) and filtered (b) trajectories before the change occurs.

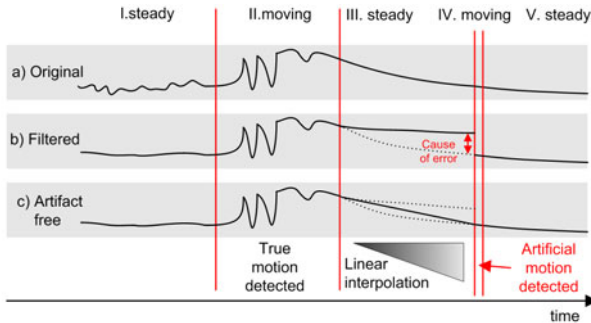


Fig. 6. Example of artifact removal on trajectories of exemplary pixel value

4 Results

We have evaluated proposed approach to estimation of temporally consistent depth maps by use of noise removal experimentally. To assess influence of presented noise reduction technique on quality of depth data, depth maps have been estimated from denoised and from original video. The tests have been performed on some of multiview video sequences that are currently used in 3D standardization [10,11]. A state-of-the-art depth estimation algorithm based on graph cuts [9] implemented in ISO/IEC MPEG Depth Estimation Reference Software have been used. For noise reduction, our presented simple technique was used.

Quality of depth maps was assessed indirectly, by assessment of quality of synthesized views (Fig. 7). These views have been synthesized with use of depth maps (generated from original and denoised video) and original sequences (even when depth maps were obtained from denoised version). Such methodology was motivated by fact, that direct subjective evaluation of depth maps has no sense.

Quality have been evaluated objectively (PSNR - Fig. 8 on vertical bars) and subjectively (15 subjects, MOS - Fig. 8) in comparison with original views. In case our study, MOS (Mean Opinion Score) is expressed by a 10-point continuous scale. Rating of the quality was in range from 1 (very bad with annoying impairments/artifacts) to 10 (excellent, artifacts are imperceptible).

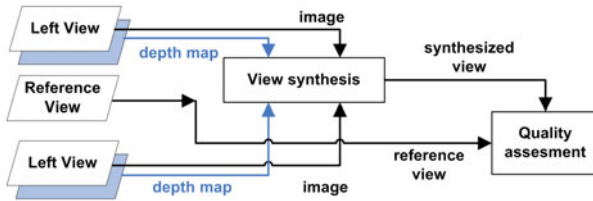


Fig. 7. Depth map quality assessment by assessment of quality of synthesized view

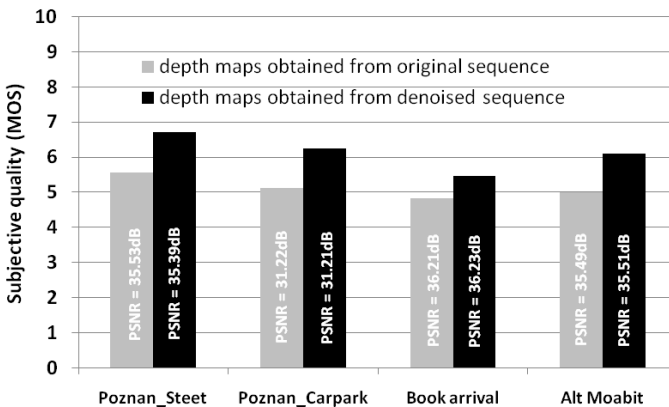


Fig. 8. Depth map quality assessment by assessment of quality of synthesized view

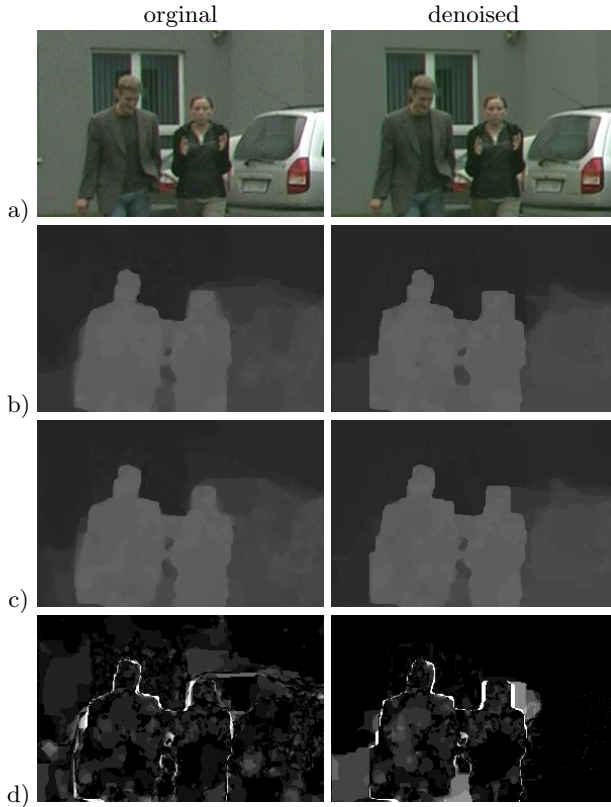


Fig. 9. Exemplary results of proposed technique: original (left) and denoised (right). a) image b,c) depth maps for two consecutive frames d) difference between depth maps.

Results presented in Fig. 9 show that use of proposed approach increased subjective quality of synthesized views from about 0.7 to 1.2 MOS points. It can also be noticed that PSNR levels have not changed. The latter is not surprising, because PSNR measure is not designed to assess quality of temporal consistency, and because only original sequences have been used for synthesis.

Figure 9 show exemplary results attained with and without use of proposed approach. As can be noticed on Fig. 9a, Moving objects (people) are left unchanged while background (wall and cars) is significantly denoised. It is worth to notice that denoised images are not blurred, because only temporal filtering is employed. Although quality of depth maps (Fig. 9b,c) has not changed, temporal consistency expressed as difference between frames (Fig. 8d) is vastly improved. As shown, background remains static (black means no changes) and thus is consistent in time. Of course, there is no improvement over moving objects, because they are not filtered.

5 Conclusions

A novel approach to providing temporally consistent depth maps has been presented. As has been shown, use of devised technique significantly increases subjective quality of depth maps. Although objective quality is not altered, proposed approach can be successfully used for estimation of depth maps, where temporal consistency is desired, like in 3D television systems. An interesting new direction of work is to test proposed approach with use of different algorithms for depth estimation and more advances noise removal techniques. In this work, only an exemplary setup with simple noise reduction technique was used. It may be worth to extend presented denoising technique in future, to support moving objects instead of leaving them unchanged.

References

1. Domański, M., Klimaszewski, K., Konieczny, J., Kurc, M., Luczak, A., Stankiewicz, O., Wegner, K.: An experimental Free-view Television System. In: IP&C, Bydgoszcz, Poland (September 2009)
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intern. Journal of Comp. Vision* 47, 742 (2002)
3. Scott Larsen, E., et al.: Temporally Consistent Reconstruction from Multiple Video Streams Using Enhanced Belief Propagation. In: ICCV 2007 (2007)
4. Pedro, F.F., Daniel, P.H.: Efficient Belief Propagation for Early Vision. *International Journal of Comp. Vision* 70(1) (October 2006)
5. Tao, H., Sawhney, H.S., Kumar, R.: Dynamic Depth Recovery from Multiple Synchronized Video Streams. In: ICIP 2003 (2003)
6. Vaseghi, S.V.: *Advanced Digital Signal Processing and Noise Reduction*, 3rd edn. John Wiley & Sons, Chichester (2006) ISBN: 978-0-470-09495-2
7. Van Etten, W.C.: *Introduction to Random Signals and Noise*. John Wiley & Sons, Chichester (2006) ISBN 0-470-02411-9
8. Dugad, R., Ahuja, N.: Video denoising by combining Kalman and Wiener estimates. In: *Proceedings of ICIP*, pp. 152–156 (1999)
9. Stankiewicz, O., Wegner, K., Wildeboer, M.: A soft-segmentation matching in Depth Estimation Reference Software (DERS) 5.0, ISO/IEC JTC1/SC29/WG11 MPEG/M17049, Xian, China (October 2009)
10. Domański, M., Grajek, T., Klimaszewski, K., Kurc, M., Stankiewicz, O., Stankowski, J., Wegner, K.: Poznan Multiview Video Test Sequences and Camera Parameters, ISO/IEC JTC1/SC29/WG11 MPEG/M17050, Xian, China (October 2009)
11. Feldmann, I., Mueller, M., Zilly, F., Tanger, R., Mueller, K., Smolic, A., Kauff, P., Wiegand, T.: HHI Test Material for 3D Video. MPEG/M15413, Archamps, France (April 2008)

Smooth Detail Features on Multiresolution Surface

Nanik Suciati¹ and Koichi Harada²

¹ Sepuluh Nopember Institute of Technology, Indonesia
nanik@if.its.ac.id

² Hiroshima University, Japan
harada@cedar.mis.hiroshima-u.ac.jp

Abstract. Addition of detail features on surface, such as raised and descend creases, which are encountered in many real objects, is identified as one important type of surface editing in 3D modeling. Its implementation which should allow addition of detail features on any part of surface, however, is not trivial due to the smoothness constraint of the underlying surface. This paper proposes a method for creating detail features on surface along a set of user-defined curves which preserves the smoothness of the final design. We implement our method in multiresolution framework so that editing the structural features of surface can be carried out easily as well as the detail features.

1 Introduction

In a very competitive world wide market, beside the quality and cost, the aesthetic aspect of industrial products is also becoming the important success key elements. Based on analyzing the working methods of the product designers [8], it appears that the shape of product is defined by two features: (a) structural features, providing the overall shape or global effect of the product, (b) detail features, originating the complete final shape of the product. In the designing activity by using Computer Aided Design (CAD) systems, in the first phase the designer creates surfaces that bound the overall shape, and then, in the second phase, add/edit the detail features characterizing the product functionally and aesthetically. The tools currently available in commercial CAD systems to support the designing of industrial products are still not sufficiently suited [5]. As the result, developing many types of computer-based product editing tools which consider the quality, cost and aesthetic aspects has become an emerging research direction in recent years.

Creating convincing 3D models is a major challenge in the broad area of industries, such as in automotive, entertainment, architecture, etc. However, the interfaces of commercial CAD systems are mostly based on constructing geometry, menus and selection, which are more natural to computer scientists and engineers than to artists [12], who usually perform the product designing by drawing or sketching. An alternative modeling system, which is based on sketch,

ideally provides a more natural interface and enables the less technically artists to leverage their sketching and drawing skills for model creation.

Polygons, non-uniform rational B-spline (NURBS) and subdivision surfaces are mostly used in the current systems to represent the underlying geometry of 3D models. In any surface representation, the availability of several types of surface editing which ease creating the geometric models is important. Multiresolution surface that accommodates editing a surface at different scales, allowing structural features (global shape) deformations as well as detail features (local shape) creation, has been used as the framework of many types of surface editing. Pasting the detail features from source surface into target surface [2], deformation of global shape with an intuitive preservation of the local shape [4], creation of sharp detail features on surfaces [3], deformation of shape with volume preservation [13] and creating wrinkles on surface [14] are examples of surface editing implemented in multiresolution framework.

Detail features such as raised and descend creases which are encountered in many real objects, for examples, the smooth lines on the body of car, the skin contraction on the human hand, etc, has evoked many attentions in the developing a surface editing type that allows addition of detail features on surface easily. However, due to smoothness constraint of the underlying surface, implementation of this style of surface editing, which should allow addition of details features on any part of surface, is not trivial, more over if the detail features are not aligned with iso-parameter lines or patch boundaries of the surface.

The goal of this research is to propose an algorithm that allows creation of detail features on surface along a set of the user-defined curves and preserves the smoothness of the final design. Our system supports sketch-based modeling system since the user could create detail features by drawing/sketching curves on the base surface, not by moving mesh of control points one-by-one. We implement our algorithm in the framework of multiresolution surface, which is an extension of multiresolution representation for endpoint-interpolating B-spline curves proposed by Finkelstein [7]. The framework ensures that editing the final design (with detail features have been added) at coarse resolution will modify the structural feature while preserving the detail features of the surface.

2 Related Works

There are many possible ways to add the detail features we desire to a surface. Free-form deformations [15] [16] modify a given surface shape through a suitable deformation, typically using a force field function. The advantage of these methods is that they are, in principle, applicable to any surface modeling setting. However, typically they do not present a unified representation that includes both the original surface and the edits. Thus, in many cases, the resulting representation is not the same as the original, but an extension of it. The main drawback of this approach is that algorithms that have been developed for the original representation are not directly applicable to the result and special cases may have to be considered.

Most algorithms developed on subdivision surface focused on defining special rules for creating sharp detail surfaces. Hoppe [10] introduces rules to create sharp features on subdivision surfaces. The work of DeRose [6] extends Hoppes approach to achieve creases with controllable sharpness by using subdivision rules parameterized by a sharpness factor. Biermann [1] builds upon subdivision schemes for piecewise-smooth surfaces where control mesh vertices and edges are tagged in order to generate singularities, such as creases, darts, and corners. In all of these techniques, there is a common requirement that features need to be aligned with the edges of the underlying control mesh. Therefore, the control mesh has to be designed with a particular feature in mind. However, a designer might want to first model an initial shape and apply detail features in later stages of the design at arbitrary locations on the surface. Another works by Biermann [3] and Khodakovsky [11] improve the works on subdivision surface by allowing creating detail features using a set of curves strokes at arbitrary locations on the surface as user input.

The closest work to the technique presented in this paper is that of Khodakovsky et al [11]. They describe a method for interactive creation of detail feature at arbitrary locations on the surface. To create a detail features along a curve, a perturbation according to a given profile is applied in the neighborhood of the curve. In their work, creating the detail features while maintaining the smooth condition of the surface is carried out by: (1) tracing the region of influence around the curve by computing the projection of rays onto surface, (2) moving all of the points in the region of influence according to a given profile. In our method: (1) we do not need to compute the projection of rays, since the region of influence is defined by user, which also allows creating much more variations of the detail features, (2) we increase the resolution of surface into an appropriate resolution and move only the mesh of control points in the region of influence, so that we perform a fewer number of computation. Since we implement our method in multiresolution surface, the final design which consisted of the base surface and the detail features will become a unified representation. Thus, editing the global shape of the final design will preserve its local shape.

3 Multiresolution Surface

The capability of multiresolution surface to accommodates editing a surface at different scales, allowing structural features (global shape) deformations as well as detail features (local shape) creation, evokes many attention of the using of multiresolution framework in many type of surface editing. Since there are some different methods to arrange multiresolution surface, the important point is how to represent one surface into some different resolutions without losing information. Thus, when moving to lower resolution, there should be a way to capture the difference information from its high resolution. Our multiresolution surface is an extension of multiresolution representation for endpoint-interpolating B-spline curves proposed by Finkelstein et al [7], which shows how multiresolution representation supports a variety of curve editing in a simple and efficient manner.

An endpoint-interpolating B-spline surface (referred as EI B-spline surface) is defined by blending the EI B-spline basis functions B running in two independent parameters u and v scaled by array of control points C (control mesh). A certain patch $Q_{i,j}$ of EI B-spline surface is computed by a double summation below:

$$Q_{i,j}(u, v) = \sum_{r=-3}^0 \sum_{s=-3}^0 C_{i+r,j+s} B_{i+r}(u) B_{j+s}(v) . \tag{1}$$

A multiresolution representation for EI B-spline surface is constructed by generating a hierarchy of control mesh, so that a surface is represented by some difference resolutions. In each resolution, a surface is approximated by a difference control mesh. Let C^n be a control mesh at resolution n consisting of $m \times m$ elements. The decomposition process is used to compute a low resolution version of control mesh C^{n-1} consisting of $m' \times m'$ elements, difference information in row Dx^{n-1} consisting of $m'' \times m$ elements and difference information in column Dy^{n-1} consisting of $m'' \times m'$ elements. If $m = 2^n + 3$, $m' = 2^{n-1} + 3$ and $m'' = 2^{n-1}$, the decomposition process can be expressed by a set of equation as follows:

$$\begin{aligned} Ct_{(m \times m')}^n &= \left(A_{(m' \times m)}^n C_{(m \times m)}^n \right)^T \\ Dx_{(m'' \times m)}^{n-1} &= B_{(m'' \times m)}^n C_{(m \times m)}^n \\ Dy_{(m'' \times m')}^{n-1} &= B_{(m'' \times m)}^n Ct_{(m \times m')}^n \\ Ct_{(m' \times m')}^{n-1} &= \left(A_{(m' \times m)}^n Ct_{(m \times m')}^n \right)^T . \end{aligned} \tag{2}$$

Process of decomposing C^n can be carried out recursively to $C^n - 1$ and its lower resolution versions, so that the original control mesh C^n can be expressed as a hierarchy of control mesh at lower resolution C^0, C^1, \dots, C^{n-1} , details in row $Dx^0, Dx^1, \dots, Dx^{n-1}$ and details in column $Dy^0, Dy^1, \dots, Dy^{n-1}$.

The reconstruction process to recover the original control mesh C^n from $C^n - 1, Dx^{n-1}$ and Dy^{n-1} is expressed by the following set of equation:

$$\begin{aligned} Ct_{(m' \times m)}^{n-1} &= \left(P_{m \times m'}^n C_{(m' \times m')}^{n-1} + Q_{m \times m'}^n Dy_{(m' \times m')}^{n-1} \right)^T \\ C_{(m \times m)}^n &= P_{m \times m'}^n Ct_{(m' \times m')}^{n-1} + Q_{m \times m'}^n Dx_{(m' \times m)}^{n-1} . \end{aligned} \tag{3}$$

For example, we have an original EI B-spline surface at resolution 2,2 consisting of 4×4 patches which is created by using 7×7 control mesh as shown in Fig. 1(a). A lower resolution version 1,1 of the surface consisting of 2×2 patches (see Fig. 1(b)) can be generated by the decomposition process, while a finer version at resolution 3,3 consisting of 8×8 patches (see Fig. 1(c)) can be generated by the reconstruction process.

4 Detail Features Creation

The input of our system is mesh of multiresolution surface, curves drawn at arbitrary location on the surface and parameters determining shape of detail

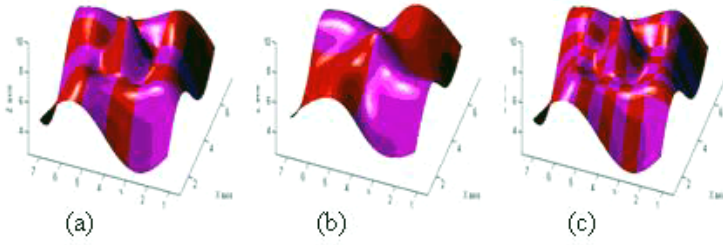


Fig. 1. EI B-spline surface at some different resolutions: (a) 2,2 (b) 1,1 (c) 3,3

features. The 4 parameters: *radius*, *height*, *D0*, and *n*, which are related to the butterworth function, are used to determine the shape of the detail features.

The three main steps of creating detail features are (1) increasing resolution of the surface mesh in order to maintain the smoothness of the surface and to maintain the shape of the added detail features, (2) identifying all control points in the distance within radius from the user-defined curves as a region of influence, and (3) displacing control points in the region of influence according to the butterworth function profile. An example of a user-defined curve on a surface, a result after creating detail features and an overview of our algorithm are shown in Fig. 2(a), 2(b) and 2(c), respectively.

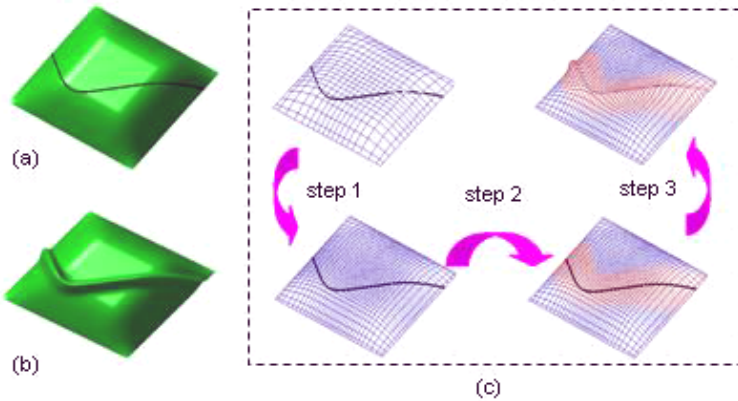


Fig. 2. An example input (a), an example of output (b) and an illustration of the 3 main steps in creating detail features (c)

4.1 Increasing Resolution of Mesh

In our multiresolution EI B-spline surface, 0 is the coarsest resolution of the surface, consisting of 4×4 mesh of control points in 3D-coordinates. Increasing

the resolution of surface is carried out by computing a set of Eqn. 3 with mesh of control points at current resolution as C^{n-1} , 0 as the value of detail in row Dx^{n-1} and 0 as the value of detail in column Dy^{n-1} . Since the higher the resolution of the surface mesh, the more number of control points used to represent the surface, representing a surface in high resolution will maintain the shape of detail features and the smoothness of the surface in the augmentation process.

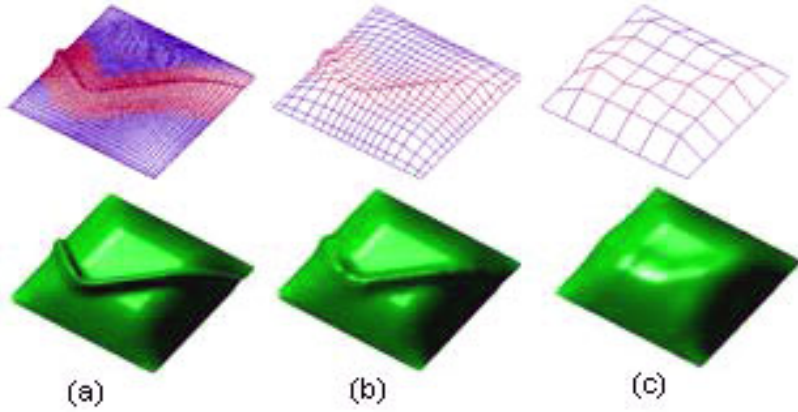


Fig. 3. Applying same detail features on different resolutions of the base surface: (a) resolution 4,4 (b) resolution 2,2 (c) resolution 1,1

Figure 3 shows the different results after adding detail features on different resolutions of the base surface. In low resolution, maintaining the smoothness of the final result and maintaining the shape of the added detail features are difficult to be done, as shown in Fig. 3(b) and 3(c). The proper resolution for creating detail features in this example is resolution 4,4, as shown in Fig. 3(a).

4.2 Identifying Region of Influence

Region of influence is a region on the base surface consisting of all control points within distance *radius* from the user-defined curves. We use Euclidian distance to compute the distance between control points and the user-defined curves. *Radius* is a parameter defined by the user to determine the width of detail features around the user-defined curves.

The following algorithm is used to identify the region of influence:

1. For each control point of mesh (CP)
 - (a) For each stroke point (SP) in the user-defined curves
 - i. Compute the Euclidian distance between CP and SP
 - (b) Choose the minimum Euclidian distance between CP and SP

4.3 Displacing Mesh

Since we wish to create many variations of detail features, such as bar-like, sharp-like and bubble-like, we choose butterworth, one of the most commonly used digital filters in image processing [9], as the profile function. In its original application domain, butterworth filters are transition filter between two extreme filters: ideal and gaussian, which its shape can be easily modified to the others filter. The following equation is used to define butterworth function:

$$f(r) = \frac{1}{(r/D_0)^{2n}} . \quad (4)$$

Figure 4 shows shapes of butterworth functions defined by $D_0=0.5$ and different values of parameters r and n . The value of r is assumed as Euclidian distance between control points and the user-defined curve. The values of $f(r)$ vary from 1 to 0, with cut-off frequency (the transition between $f(r)=1$ and $f(r)=0$) D_0 and order of the function n . The shape of butterworth function which can be easily modified through the using of different combinations of parameter values n and D_0 , can be used as the force function to create many variations of detail features in surface, such as bar-like, sharp-like and bubble-like.

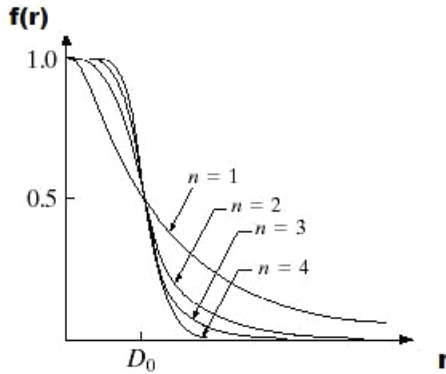


Fig. 4. The butterworth functions defined using different values of parameters n

All control points in the region of influence of the base surface are displaced in the surface normal direction according to the butterworth function profile. The height of detail features is specified by *height* parameter. Figure 5 (a)-(d) show some variations of detail features created using different values of *radius*, D_0 , *height* and n . In experiment, parameter $n=4$ and relatively big values of D_0 produce bar-like detail features, while parameter $n=4$ and small values of D_0 produce sharp-like detail features. Parameter $n=2$ can produce bubble-like detail features. Raised and descend creases is created using positive and negative values of *height* parameter.

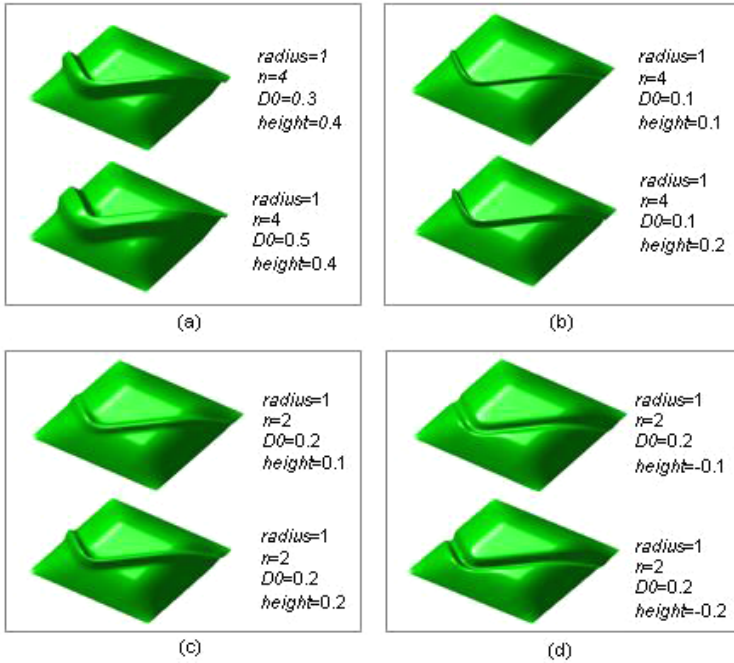


Fig. 5. (a) Bar-like detail features (b) Sharp-like detail features (c) Bubble-like detail features (d) Bubble-like descend creases



Fig. 6. An example of detail features created by multiple curves and the result after editing the structural features

Since we work in the multiresolution framework, editing the structural features should preserve the detail features which might have been added. As shown in Fig. 6, addition of detail features on surface, and then, is followed by modification of the structural features of the base surface, gives the final result which preserves the detail features. Figure 6(c) shows the final result after editing the structural features of surface in Fig. 6(b), which is simply carried out by decreasing the resolution of surface and capturing all information in high resolution, modifying the surface at low resolution, and then returning the surface to its original resolution.

5 Conclusion

We propose a method to create many variations of smooth detail features on surface along a set of the user-defined curves using butterworth as the force function. We implement our method in the framework of multiresolution endpoint-interpolating cubic B-spline surface, which allows editing the structural features of the surface without losing any detail features in its original resolution. Compared to the closest work in the features creation, our method performs a fewer number of computations and provides much more variations of the shape of detail features.

References

1. Biermann, H., Levin, A., Zorin, D.: Piecewise-smooth subdivision surfaces with normal control. Proc. of SIGGRAPH, 113–120 (2000)
2. Biermann, H., Martin, I., Bernardini, F., Zorin, D.: Cut-and-paste editing of multiresolution surface. ACM Trans. on Graphics, 21(3), 312–321 (2002)
3. Biermann, H., Martin, I., Zorin, D., Bernardini, F.: Sharp features on multiresolution subdivision surfaces. Proc. of Pacific Graphics, 140–149 (2001)
4. Botsch, M., Kobbelt, L.: Multiresolution surface representation based on displacement volumes. Computer Graphics Forum, 22(3), 483–491 (2003)
5. Cheutet, V., Catalano, C. E., Pernot, J. P., Falcidieno, B., Giannini, F., Leon, J. C.: 3D sketching with fully free form deformation features (δ -F4) for aesthetic design. Workshop on Sketch-Based Interfaces and Modeling, 9–18 (2004)
6. DeRose, T., Kass, M., Truong, T.: Subdivision surfaces in character animation. Proc. of SIGGRAPH, 85–94 (1998)
7. Finkelstein, A., Salesin, D. H.: Multiresolution curves. Proc. of SIGGRAPH, 261–268 (1994)
8. Fontana, M., Giannini, F., Meirana, M.: A free form feature taxonomy. Computer Graphics Forum, 18(3), 107–118 (1999)
9. Gonzales, R. C., Woods, R. E.: Digital image processing. Prentice Hall, 167–178 (2002)
10. Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W.: Piecewise smooth surface reconstruction. Proc. of SIGGRAPH, 295–302 (1994)
11. Khodakovsky, A., Schroder, P.: Fine level feature editing for subdivision surfaces. Proc. of ACM Symposium on Solid and Physical Modeling, 203–211 (1999)
12. Olsen, L., Samavati, F. F., Sousa, M. C., Jourge, J. A.: Sketch-based mesh augmentation. Workshop on Sketch-Based Interfaces and Modeling, (2005)
13. Sauvage, B., Hahmann, S., Bonneau, G.: Volume preservation of multiresolution meshes. Computer Graphics Forum, 26(3), 275–283 (2007)
14. Sauvage, B., Hahmann, S., Bonneau, G.: Length constrained multiresolution deformation for surface wrinkling. IEEE Int. Conf. on Shape Modeling and Applications, 21–31 (2006)
15. Sederberg, T., Parry, S.: Free-form deformation of solid geometric models. Computer Graphics, 20(4), 151–160 (1986)
16. Singh, K., Fiume, E.: Wires: A geometric deformation technique. Proc. of SIGGRAPH, 405–414 (1998)

Using Parallel Graph Transformations in Design Support System

Rafał Świdorski and Barbara Strug

Department of Physics, Astronomy and Applied Computer Science
Jagiellonian University, Reymonta 4, Kraków, Poland
{rafal.swiderski,barbara.strug}@uj.edu.pl

Abstract. In this paper an application of graph transformations using the parallel derivation approach in design system is presented. It is based on earlier research in formal language theory, especially graph grammars, and distributed models. A motivation and possible ways of application of the ideas presented in this paper are given. In this paper an implementation of the approach is also described. The implementation uses SQL and a cluster systems. The approach is illustrated by an example from the domain of flat layout design.

1 Introduction

This paper deals with a linguistic approach [2] to computer-aided design. The model of computer-aided design used here is based on graph structures for which a lot of research has been done in context of design [3]. Semantic knowledge about the design is encoded in attributes. The graphs used to represent objects being designed are generated by grammars which consist of graph rules.

In many design situations a complex problem is divided into a number of subproblems. So we propose a parallel approach to a design system based on applying graph rules to many subproblems at the same time. Searching for each subproblem solution is supported by the same grammar. Solving the whole problem requires the ability to apply productions to different subgraphs of the same graph representing a design solution being generated. [4,5,6,7,8,9].

Our previous research on solving the problems of parallel and distributed ways of using graph transformations is presented in [10,11,12,13,14]. In this paper the main problem is applying (different) productions in parallel to different subgraphs of the same graph. This approach uses an idea of control diagram in deciding on the appropriate production to be used. The presented graph grammars generate a set of graphs representing design task solutions compatible with the given design criteria. The proposed approach is illustrated on the example of designing house layout. Final remarks and future investigation directions are presented in conclusions.

2 Design Representation and Generation

One of the most useful representations in computer aided design systems is based on graphs. The design process can then be seen as the generation of graphs representing design solutions. Thus introducing a method for the generation of these graphs is required. Moreover such method must guarantee that graphs it generates are correct in respect to a design problem. Each generated graph must be convertible to a design. In many practical situations it must also satisfy predefined design criteria and constraints. Graph transformations are one of the methods fulfilling all above requirements.

In actual situations the object to be designed are complex and contain many parts which in turn can contain subparts. Thus the graphs representing such an object can become large structures difficult to maintain, update and analyze. Yet in many situations design decisions dealing with one part of the design are independent from the rest of it. Thus it seems reasonable to be able to work in parallel on different elements of the graph representing such a design.

A graph is defined as a pair (V, E) , where V and E are sets of graph nodes and edges, respectively. Formally the definition is as follows.

Definition 1. *A graph is a pair $G = (V, E)$ where E is a set of edges and each $e \in E$ has a form $e = \{(v_1, v_2, i)\}$, such that $v_1, v_2 \in V$ and i is the identifier of the edge e .*

For the needs of a design system a node in a graph may represent either an object or a group of objects. For example, in a flat layout design system a node can represent a floor of a house that in turn will be divided into rooms. Each node of a graph is labelled by a symbol from a predefined set (alphabet) that is later used to identify what the given node represents.

Edges in such a graph represent relations between objects. They are labelled by symbols being names of the relations. Labels are assigned to nodes and edges by means of node and edge labelling functions.

Both objects and relations between them may in turn have some features. To represent them attributing is used. Attributes represent properties (such as size, position, area, orientation and others) of an element represented by a given node. Attributes are assigned by node and edge attributing functions. Formally, an *attribute* is defined as a function $a : W \rightarrow D_a$, where W is a domain of the attribute and D_a a set of possible values of the attribute.

Let R_V and R_E be the sets of node and edge labels, respectively. Let A and B be sets of node and edge attributes and D_a and D_b sets of possible values of attributes of nodes and edges, respectively.

Definition 2. *A labelled attributed graph LAG over $R_V \cup R_E$ is defined as a 6-tuple $LAG = (V, E, lab_V, lab_E, att_V, att_E)$, where*

1. (V, E) is a graph,
2. $lab_V : V \rightarrow R_V$ is a node labelling function,
3. $lab_E : E \rightarrow R_E$ is an edge labelling function,

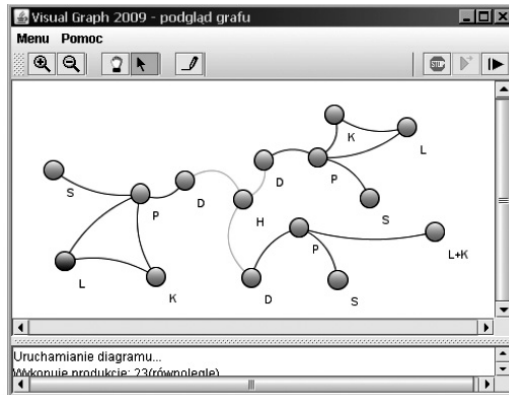


Fig. 1. Example of a graph representing a flat layout

4. $att_V : V \rightarrow P(A)$ is a function assigning a set of attributes to each node,
5. $att_E : E \rightarrow P(B)$ is an edge attributing function, i.e. it assigns a set of attributes to each edge.

In Fig. 1 a graph representing parts of a flat generated by a Visual Graph application is shown. Nodes represent parts of a flat (rooms, halls, sleeping and living areas) and edges represent communication possibility between them. Such a graph may be extended to add more details. Attributes defining such parameters like size or distance are not also shown on the picture.

A labelled attributed graph defined above can represent an infinite number of designs having the same structure. To represent an actual design we must define a graph interpretation.

An interpretation of a graph G is defined as a pair of functions (I_V, I_E) , where I_V assigns geometrical objects to nodes, while I_E establishes a correspondence between edges and relations between these objects. The objects assigned by I_V are usually called primitives. They may be either simple geometrical objects or more complex objects. The geometry of these objects may be internally represented by means of any known representation that allows for easy application of similarity transformations. Geometrical objects used depend on the domain of application.

The interpretation allows us to define attribute functions for a graph and then determine its instances. An instance of a graph is a labelled attributed graph in which to each nodes and edges values of their attributes are assigned. In a house design system for all rooms we have to specify its size attribute for example.

When having a graph interpretation we can create a visualization of a designed object. By changing the set of available primitives the interpretation (and a resulting design) is changed without any changes made to its graph structure.

2.1 Graph Generation

The graphs representing designs may be dynamically generated by means of so called graph grammars. A graph grammar can be designed to allow for generating only graphs representing the particular class of designs.

Intuitively, a graph grammar is described as a system of graph transformations consisting of formal rules called productions. Each production is composed of left-hand side and right-hand side graphs and the first graph is replaced by the second one only if it appears in the graph to be transformed. A production is applied to a current graph (starting from the initial one). The sequence of such replacements, which starts from the initial graph, is called a derivation process.

However in many design situations there is no need to apply the transformations in a sequence. Many productions can be applied at the same time. To do it we have to search for many subgraphs in a current graph. Each found subgraph should be isomorphic with the left-hand graph of a production. To be able to apply productions in parallel we have to limit the set of subgraphs to disconnected ones.

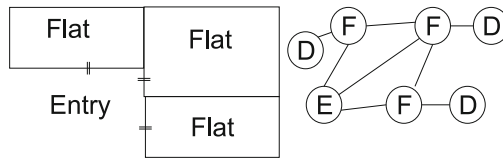


Fig. 2. Example of a house layout outline and its graph representation

For example in a house design different parts of a house can be designed simultaneously. In Fig. 2 an outline of a house layout is presented and its graph representation. Using an appropriate graph grammar this design can be developed to generate a final detailed design of all flats. It is obvious that the generation of each flat detailed structure is independent from the others, but as they all share the same entrance some care must be taken.

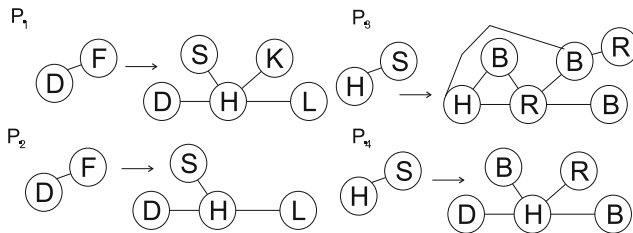


Fig. 3. Some of the productions from the house layout grammar

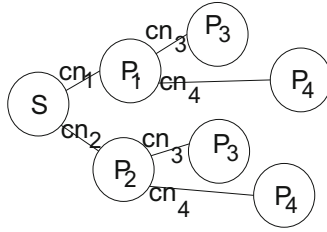


Fig. 4. Part of the control diagram

As it can be seen in Fig 3 there may be more than one production with the same left side graph. Thus a decision must be taken on which of these productions is to be applied. In our approach this decision is based on design knowledge which is coded in node attributes. Thus a control diagram, which is responsible for triggering appropriate productions is defined. Part of this diagram is shown in Fig. 4. This diagram shows the order in which productions can be applied, if more than one is possible conditions (or predicates) are checked. In the example conditions cn_1 and cn_2 are based on the value of the attribute *area* of node labelled by F , and conditions cn_1 and cn_2 are based on the value of the attribute *area* of node labelled by S . This conditions represent the fact that a flat or a sleeping area must have some minimal size to be divided into more rooms, and if the size is too small the lower number of rooms can be added.

3 The Implementation

In order to achieve this goal, a prototype language for subgraph structure description was created. It can describe relations between vertices in a subgraph and generate a set of edges in a way that allows easy search of subgraph structures in a graph. This approach allows implementation in SQL language as well as simple scaling based on cluster systems. To be able to apply productions a subgraph searching method must be established.

3.1 Subgraph Searching

Definition of subgraph is needed to determine what is the left side of production, of which the right side we want to embed. For the needs of the implementation it is defined as a set of edges $E_g \dots$, where: E_g is set of the form $\{(v_{g1}, v_{g2}, e_{gi})\}$.

To find a subgraph we need to describe the structure of the graph, represented by the set $H \forall e_i \in E$ and $\forall v_i \in E[v_1, v_2]$ that $\{(v_i, \overline{E : v_1 \in v(e_i) \vee v_2 \in v(e_i)}, e_i)\}$ Where: $v(e_i)$ is function returning vertices for i -th edge as parameter.

Subgraph structure. For the above graph structure a subgraph structure is described in the following way:

1. Set of form $X \{(e_{g_i}, e_{g_j}, x_{ij})\}$ such that:

$$\forall e_{g_i} \in E_g, \forall e_{g_j} \in E_g \exists i > j : x_{ij} = \left\{ \begin{array}{l} \overline{\overline{e_{g_i}[v_{g_1}, v_{g_2}] \in e_{g_j}[v_{g_1}, v_{g_2}]} = 1; 2} \\ \overline{\overline{e_{g_i}[v_{g_1}, v_{g_2}] \in e_{g_j}[v_{g_1}, v_{g_2}]} = 2; 3} \\ e_{g_i}[v_{g_1}, v_{g_2}] \notin e_{g_j}[v_{g_1}, v_{g_2}]; \quad 1 \end{array} \right\}$$

2. $\forall v_i \in (E_g[v_{g_1}] \cup E_g[v_{g_2}])$ create a set G such that:

$$\left\{ \overline{\overline{(v_i, E_g : v_{g_1} = v_i \vee v_{g_2} = v_i)}} \right\}.$$

Finding subgraphs in a graph. For finding sub-graph structures in a graph, let's create set $I \{(v_i, e_i)\}$:

$$\forall v_i \in (E_g[v_{g_1}] \cup E_g[v_{g_2}]) \exists G v_i \overline{\overline{E_g : v_{g_1} = v_i \vee v_{g_2} = v_i}} \cong \cong H[E : v_1 \in v(e_i) \vee v_2 \in v(e_i)]$$

and $\forall e_i \in I, \forall e_j \in I \exists (e_i, e_j) \subseteq X[e_i, e_j]$ and implement the conditions 4.3.(1). As a result of this operation we obtain a subset of possible solutions, assuming $\overline{\overline{E_g}}$ that e_i is unique in the subset of solutions and the size of the subset is equal $\overline{\overline{E_g}}$.

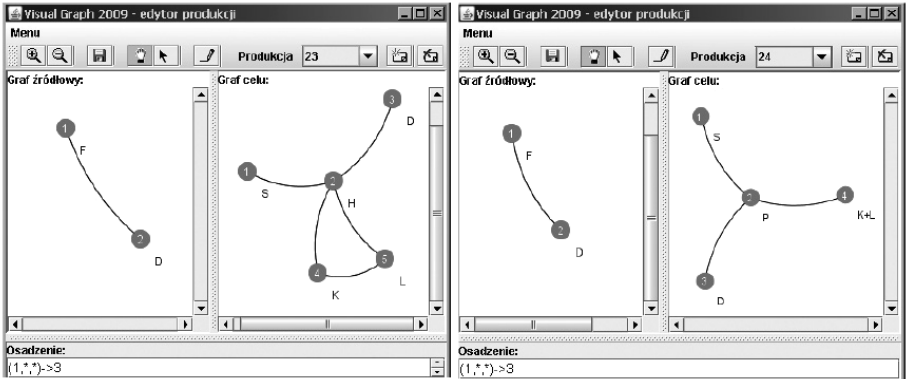


Fig. 5. Production and its embedding transformation

Description of production. Replacement of the left side of production with the right side of productions requires a subgraph isomorphic with the left side of the production to be removed from the current graph. It involves also the removal of all edges connecting the removed subgraph to the rest of the graph, which are known as the embedding. To properly connect a right side graph with the rest graph we have to define an embedding function.

Embedding function has form: $(V_{i_S}, \Gamma_{i_S}, E_{S_{direct}}) \rightarrow V_{i_T} \vee * \Sigma_{i_T}$

Where:

- * - any value,
- V_i - vertex of graph (S-ource or T-arget),
- Γ_i - edge label,

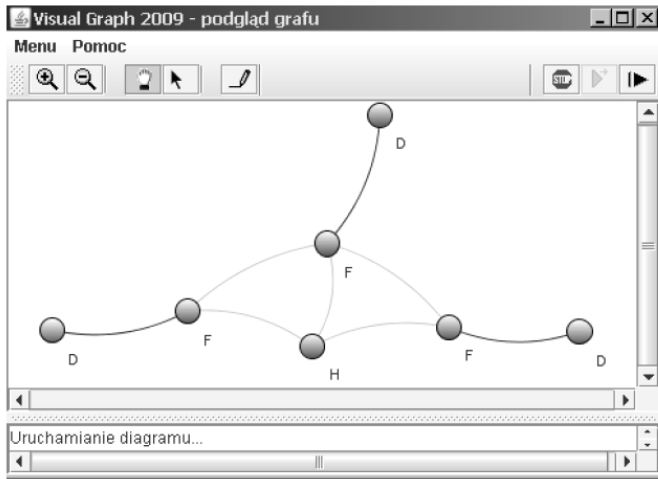


Fig. 6. Graph before the application of the productions

- E - directed edge of graph {in,out,any},
- Σ_i - vertex label.

Embedding function for one of the productions shown in Fig 6 is as follows:
 $(1,*,*) \rightarrow 2$, $(2,*,*) \rightarrow 3$, $(3,*,*) \rightarrow 4$ and $(4,*,*) \rightarrow *A$.

3.2 Complexity of the Algorithm

The complexity of the algorithm can be estimated as $2^{\bar{e}-1} * 2^{\bar{E}} + \bar{e}!$. However in practical situations the node labelling makes the actual complexity much lower. Further optimization is possible.

3.3 Visual Graph Software

The approach described in this paper was implemented as a Visual Graph application. This software, which implements the described algorithm, has been written in JAVA and uses SQL engine. It can efficiently find subgraphs and provides several embedding transformations.

4 Conclusions and Future Work

In this paper a framework for parallelization of design tasks was presented. It is worth noticing that this approach does not make any assumptions about the type of graph transformations that are used. The use of this method in design context both simplifies it and adds complexity. On one side parallel application of productions accelerates the process of generating a new design. On the other hand it requires finding multiple subgraphs in a current graph. The approach

presented here was implemented and tested on the example of flat layout design system, but the system can be used for any domain for which a grammar can be defined.

Currently a research is being carried out on using this approach for generating other objects. We also plan further research into the decision process of selecting productions, which now is based on attribute values only. It is also planned in the near future to use computer cluster to accelerate the task.

References

1. Rozenberg, G.: Handbook of Graph Grammars and Computing By Graph Transformation. Foundations, vol. I. World Scientific Publishing Co., NJ (1997)
2. Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G.: Handbook of Graph Grammars and Computing By Graph Transformation. Application, Languages and Tools, vol. II. World Scientific Publishing Co., NJ (1999)
3. Grabska, E., Palacz, W.: Hierarchical graphs in creative design. *MG & V* 9(1/2), 115–123 (2000)
4. Csuhaj-Varju, E., Vaszil, G.: On context-free parallel communicating grammar systems: Synchronization, communication, and normal forms. *Theoretical Computer Science* 255(1-2), 511–538 (2001)
5. Csuhaj-Varju, E., Dassow, J., Kelemen, J., Paun, G.: Grammar systems. A grammatical approach to distribution and cooperation. *Topics in Computer Mathematics*, vol. 8. Gordon and Breach Science Publishers, Yverdon (1994)
6. Csuhaj-Varju, E.: Grammar systems: A short survey. In: *Proceedings of Grammar Systems Week 2004*, Budapest, Hungary, July 5-9, pp. 141–157 (2004)
7. Kelemen, J.: Syntactical models of cooperating/distributed problem solving. *Journal of Experimental and Theoretical AI* 3(1), 1–10 (1991)
8. Martin-Vide, C., Mitrana, V.: Cooperation in contextual grammars. In: Kelemenov, A. (ed.) *Proceedings of the MFCS 1998 Satellite Workshop on Grammar Systems*, pp. 289–302. Silesian University, Opava (1998)
9. Simeoni, M., Staniszkis, M.: Cooperating graph grammar systems. In: Paun, G., Salomaa, A. (eds.) *Grammatical models of multi-agent systems*, pp. 193–217. Gordon and Breach, Amsterdam (1999)
10. Grabska, E., Strug, B.: Applying Cooperating Distributed Graph Grammars in Computer Aided Design. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Waśniewski, J. (eds.) *PPAM 2005*. LNCS, vol. 3911, pp. 567–574. Springer, Heidelberg (2006)
11. Grabska, E., Strug, B., Slusarczyk, G.: A Graph Grammar Based Model for Distributed Design. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) *ICAISC 2006*. LNCS (LNAI), vol. 4029. Springer, Heidelberg (2006)
12. Kotulski, L., Strug, B.: Distributed Adaptive Design with Hierarchical Autonomous Graph Transformation Systems. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloat, P.M.A. (eds.) *ICCS 2007*. LNCS, vol. 4488, pp. 880–887. Springer, Heidelberg (2007)
13. Kotulski, L., Strug, B.: Using Graph Transformations in Distributed Adaptive Design System. In: *ICCVG 2008*. LNCS, vol. 5337, pp. 477–486. Springer, Heidelberg (2008)
14. Kotulski, L., Strug, B.: Parallel Graph Transformations in Distributed Adaptive Design. In: *Workshop on Graph Computation Methods, International Conference on Graph Transformations (GCM-ICGT 2008)*, Leicester (2008)

Fast, Parallel Watershed Algorithm Based on Path Tracing

Michał Świercz and Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland
{swierczm,iwanowski}@ee.pw.edu.pl

Abstract. In this paper a fast, parallel watershed algorithm for segmentation of digital grey-scale images is presented. We show an original parallelisation technique based on the "shared nothing" principle and its application to a modified path-tracing watershed algorithm, which allows a vast majority of computations to be broken up into several independent tasks that can be run in parallel on different processing nodes. This approach eliminates the need of any complex synchronization between the threads and translates to a very high efficiency and speed of the algorithm. Sample results are discussed, with emphasis on their correctness and execution times.

1 Introduction

The watershed transformation [9] is an established, powerful tool for morphological image segmentation. It has found many applications in different science areas. The concept of watersheds comes from the field of topography, referring to the division of an continental landscape into water catchment basins, which collect rain water flowing down from areas of land associated with them. This concept can be utilised in image segmentation to generate partitions of the image corresponding to objects of interest. A grey-scale picture is considered a topographic relief, where the height of a point is proportional to the grey level of the corresponding pixel.

During the recent years, two conceptually distinct techniques have been developed to calculate the watershed transform: The first technique (immersion-based) proposes the transformation by immersing the topographic surface in a lake, with holes pierced in all local minima of the surface. It was implemented in e.g. [16,8]. The second technique (by path-tracing, called also rainfall-based) simulates the behaviour of rain water over the topographic surface associated with the image, so is in a sense the reverse of the immersion approach. This method has been described, among others, in [23,7,10].

There have been several successful attempts of parallelisation of the watershed transform [45,6]. There are, however, inherent difficulties associated with this task, the most significant of which is the flow of processing. Due to the nature of the watershed transform, the order, in which pixels are analysed depends only on the characteristics of the image itself, and makes it very difficult

to arbitrarily divide the image into smaller sub-images for independent processing. Such methods usually require complex synchronisation and communication mechanisms between the processes.

Our approach aims at eliminating this problem altogether by performing a preliminary labeling of a small number of selected pixels first, which in turn allows to divide the main image into completely independent sets of pixels in regard to the labeling task and process them completely separately from each other. We refer to the principle behind this architecture as "shared nothing" rule. As will be demonstrated, our parallel watershed algorithm vastly outperforms both a reference immersion algorithm [1] and path-tracing algorithm [11], while retaining a clean and very scalable construction.

This paper is divided into 4 sections. Section 1 is introductory, in section 2 we present the principles of the "rainfall" segmentation method and describe our efficient parallel method based on this technique. Experimental results are presented and discussed in section 3, while section 4 includes conclusion and final remarks.

2 Algorithm

In this section we provide a detailed description of our algorithm. First, some fundamental notations are defined, then a description of a modified path-tracing sequential algorithm is outlined. Finally, we present an efficient parallel technique based on the sequential watershed transform.

2.1 Basic Definitions and Notations

Before describing in detail our approach, we introduce some basic definitions used throughout this paper. We define the following terms:

1. Pixel q is considered a **neighbour** of pixel p if and only if pixel q is adjacent to pixel p on the underlying connectivity grid. If the height of q is the lowest of all the neighbours of p , and also lower than the height of p , q is the **lowest neighbour** of p . If there are multiple pixels meeting this criterion, they are all considered lowest neighbours of p .
2. **Path** is a series of pixels, in which every two consecutive pixels in the series are neighbours in the image, and every pixel in the series is unique. As **length** of path we will understand the number of pixels in the path.
3. **Flat-zone** of height h is a set of pixels such that all the pixels are of the same height h and every two pixels can be connected by a path with all the pixels in the path also having a height of h .
4. **Regional minimum** is a pixel that does not have a neighbour with a lower height, or a flat-zone consisting of pixels without lower neighbours¹. This is one of the cardinal terms in watershed segmentation and broadly represents a catchment sink for flowing rain water.

¹ In a similar way the regional maximum is defined, but it is not used in the current study.

5. **Plateau** is a flat-zone which is neither a regional minimum nor a regional maximum, i.e. such that contains at least one pixel that have lower neighbour and at least one having higher neighbour. A pixel on a plateau is considered an **outlet** if it has at least one lower neighbour.
6. **Steepest descent path** is a shortest possible path from pixel p_A towards pixel p_B , where each successive pixel is the lowest neighbour of the previous pixel in the path, and the starting pixel p_A is not lower than pixel p_B .

An additional explanation is required for the case of steepest descent paths on plateaus. This is a special scenario, as inner pixels of a plateau do not have a lowest neighbour. In this case, we assume that the part of the steepest descent path that lies on the plateau is the shortest path between the plateau entry pixel and the nearest outlet, where geodesic distance is assumed as distance criterion.

As we mentioned before, there can exist several equally valid steepest descent paths for a pixel, leading to different regional minima. Therefore, such a pixel can belong to each of these catchment basins and a criterion must be chosen to determine which basin to use. In this work, we use the „first-found” criterion, by which to continue steepest descent path we pick the first viable lowest neighbour found during the neighbourhood scan. This is a simple and fast method that, even though is biased in specific scenarios, generates satisfying overall results.

2.2 Sequential Watershed Algorithm

Our sequential watershed algorithm implements the rainfall approach to find a downstream flow along the steepest descent path from each pixel, and distribute region labels marking the association to a regional minimum. A mechanism called directional code is used to code descent paths, where each visited pixel receives a temporary marking in the output label array. The value of this temporary marking represents the position of pixel in the image. A path can be therefore viewed as a series of pointers to pixels stored in the output array.

First, the image is surrounded by a 1-pixel wide frame of *guardian pixels* which are of higher altitude than any of the actual pixels, and therefore provide an uncrossable boundary for path tracing. This is done to improve performance and eliminate unnecessary checks of boundary conditions in each neighbourhood scan. The path-tracing process is performed for each pixel in the image and can be summarised as follows (Fig. 1 illustrates consecutive steps of the algorithm):

1. Neighbourhood scan is performed to find the lowest neighbour of the starting pixel. The lowest neighbour of the starting pixel becomes the next step in the descent path and is marked with directional code. Each new pixel in the path is in turn scanned for its lowest neighbour to add to the path, and the process continues until either a flat-zone or a single-pixel minimum is encountered. This case is demonstrated on the example of pixels e1 and e2 in Fig. 1 (a), flat-zone starts at e3 and consists of pixels of height 15. Further processing of the flat-zone depends on its type (regional minimum or plateau).

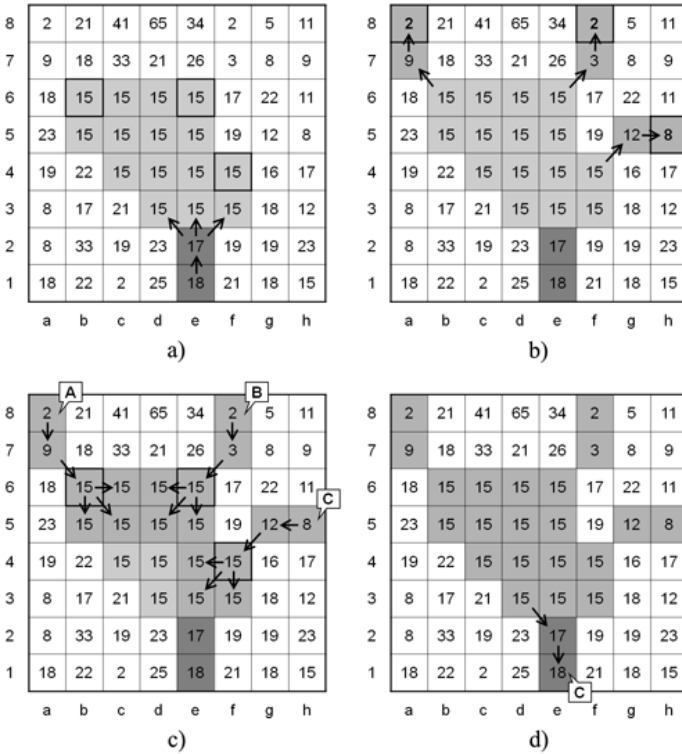


Fig. 1. Stages of sequential watershed

2. If a regional minimum (no matter if it is a single pixel or a flat-zone) is reached, it is labeled with a new identifier (unless it was already labeled during previous processing). Then, the descent path leading to the regional minimum is traversed and this minimum's label is propagated along the path, thus concluding the tracing procedure.
3. If a plateau is encountered, it is scanned to find all the pixels belonging to it, as well as to locate the plateau's outlets (pixels e6, b6, f4 in Fig. 1 (a)). All the descent paths originating in the plateau's outlets are now independently traced to find the labels of the outlets (Fig. 1 (b) and (c)). The plateau itself is then labeled using the region growing (*prairie fire*) method (Fig. 1 (c)), where the labeled outlets act as seeds for propagation and distribute their labels to their neighbours on the plateau. The labels spread with each iteration of the region growing to gradually fill the whole plateau. After labeling the plateau, the processing returns to the pixel from which the plateau was entered, and label propagation up its path is performed (Fig. 1 (d)).

The algorithm presented is able to perform the watershed transform by scanning the image just twice (plus once for initialization), which contributes greatly to

its swiftness. Also, there are cases where descent paths originating from different pixels merge at some point along the way. To avoid unnecessary tracing, we use a method called early path termination, introduced in [10]. Whenever an already labeled pixel is encountered during descent path tracing, the path is not traced further (as it was obviously already scanned) and we proceed with label propagation, just as if a minimum was reached.

2.3 Parallel Watershed Transform

We will demonstrate our concept of parallel watershed transform on an example of two-thread simultaneous processing. This setup was chosen for clarity of presentation, but the main principles of the method are also valid if the number of threads is higher. General steps of the parallel watershed transform are as follows:

1. Split image into two non-overlapping sub-images with a 1-pixel wide *separation line* (as shown in Fig. 2 (a)).
2. Execute descent path tracing and labeling for the pixels belonging to the separation line (Fig. 2 (b)).
3. Split the image into two sub-images, defined by the original image edges and the separation line (Fig. 2 (c) and (d)).
4. Process each sub-image independently in a separate CPU thread using the sequential watershed algorithm.
5. Merge partial results into final label matrix of the whole image by means of fast bulk memory copy.

In step 1, we choose a 1-pixel-wide line to break up the original image into two sub-images as presented in Fig. 2 (a). As in the sequential approach, we use surrounding frame of *guardian pixels* (G) which are higher than the highest allowed pixel altitude and ensure that no path tracing will move out of the image boundaries. Best results are achieved when the sub-images are of equal size, so that on average it takes a similar amount of time to process each of them. It is also desirable to perform this division in such a way that the two resulting sub-images occupy two continuous blocks of memory. This is dependent upon implementation and the way array data is stored in memory (row-major or

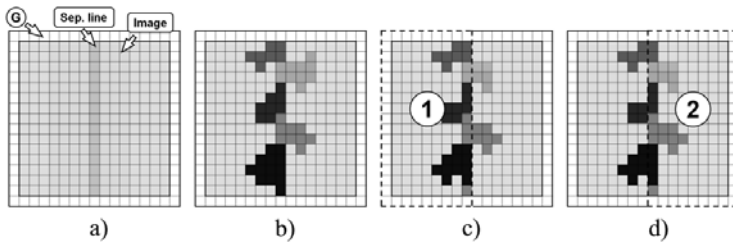


Fig. 2. Parallel watershed – stages

column-major) in a given programming environment. There is no need to perform memory copy at this stage, as only read access is required to the data structure containing the image and all threads can access it freely without interference with one another.

In step 2 all the pixels belonging to the separation line (we will refer to them as *separation pixels*) are processed and labeled. Separation pixels are arranged in a queue, and steepest descent path is found for each of them. Steepest descent path from each separation pixel is traced and after a corresponding regional minimum is reached, this minimum's label is propagated up the descent path. The result of this step is a labeled separation line along with all the neighbouring pixels and local minima belonging to the steepest descent paths of the separation pixels (Fig. 2 (b)). Step 3 involves labeling all the remaining pixels in the two sub-images in Fig. 2 (c) and (d). Sub-images 1 and 2 are bordered by guardians and the already-labeled separation pixels, which guarantees, that there cannot exist a path that would need to be traced across the two sub-images. Each sub-image is an independent labeling task which then is executed as a separate thread, in parallel. The final step involves merging the label arrays of each individual labeling thread into final, complete label array thus concluding the segmentation of the image. This is done by copying the memory locations storing partial label arrays into a continuous block so that they can be accessed by a single addressing and treated as a single data structure. As we pointed out in discussing step 1, the separation line direction is chosen to facilitate this procedure, so that each merge action involves only one simple memory block copy.

Table 1. Execution times of immersion-based, rainfall-type and proposed algorithms

Image	Size	Immersion-based	Path-tracing	Proposed
Baboon	512x512	36 ms	27 ms	10 ms
Lena	512x512	35 ms	27 ms	13 ms
Earth	350x400	10.5 ms	10.5 ms	4.5 ms

The presented parallel method combines both sequential and parallel processing. Step 2 is performed as a single-thread workload and the benefits of our approach do not present at this stage. However, the number of separation pixels is relatively small compared to the total number of pixels in the image so the workload needed to be computed sequentially is not significant and does not impact the overall performance of the algorithm, especially for larger images.

The computational complexity of the algorithm is an approximation of the values reported by [2], however, the real time consumption is much smaller due to parallel construction. The memory requirements are also similar, two array structures are needed - one for storing the image data, and the other for storing labels, temporary direction code and other mid-process data. Additionally, two queue structures are needed for plateau processing. In the worst-case scenarios, one of the queue will need to store all of the pixels in the image, so to improve execution speed we opted for static allocation. As the main goal in developing

this method was time performance, we felt this choice was justified, taking into account the size of RAM memory available to modern computers.

3 Evaluation and Experimental Results

The performance and correctness of the parallel algorithm was evaluated on a set of test images, including natural images such as "Baboon", "Lena" or "Earth" (Fig. 4) for performance benchmark, and artificial images such as the classic Acid2 test image for detailed, low-level correctness evaluation (as seen in Fig. 3, with segments shaded for clarity of presentation).

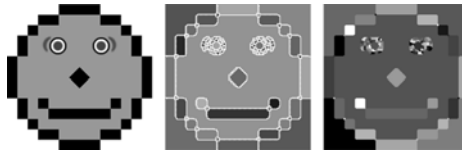


Fig. 3. Low-level test: Original image (left), Reference watershed [1] (middle), Parallel watershed (right)

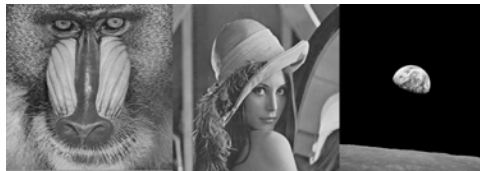


Fig. 4. Test images: Baboon (left), Lena (middle), Earth (right)

All measurements were taken using Compal PC computer with T7500 Core 2 Duo 2.2GHz CPU, 2GB of 800MHz DDR2 RAM running Windows XP operating system. The execution times presented in Table 1 are averaged over 10 tests, after discarding the shortest and the longest result. We used a stand-alone Windows console application for testing against immersion-based algorithm [1] and rainfall-type algorithm [11]. An application released by [11] was used for reference.

The low-level tests have shown that our parallel algorithm extracts proper features of the image and yields results directly comparable to reference implementations. The running times are approximately three times shorter than immersion-based reference method, and about two times shorter than rainfall-based method. It is worth pointing out that in case of images with large, uniform areas such as Earth, the immersion and rainfall techniques produce comparable results, yet the parallel approach is still over two times faster.

4 Conclusions

The presented parallel watershed algorithm proved to be a high-performing, efficient method of digital image segmentation. The "shared-nothing" parallel architecture of the method proved to be vastly superior in terms of execution time while yielding results of quality comparable to existing methods, both in low-level tests and in general task evaluation. The original approach presented in this paper allows for easy scaling of the algorithm to more advanced multi-core processors and can be further utilised to construct larger-scale implementations.

References

1. Vincent, L., Soille, P.: Watersheds in digital spaces - an efficient solution based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6), 583–598 (1991)
2. Osmá-Ruiz, V., Godino-Llorente, J.I., Sáenz-Lechón, N., Gomez-Vilda, P.: An improved watershed algorithm based on efficient computation of shortest paths. *Pattern Recognition* 40(3), 1078–1090 (2007)
3. Bieniek, A., Moga, A.: An efficient watershed algorithm based on connected components. *Pattern Recognition* 33(6), 907–916 (2000)
4. Moga, A., Gabbouj, M.: Parallel Image Component Labeling With Watershed Transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(5), 441–450 (1997)
5. Moga, A., Cramariuc, B., Gabbouj, M.: Parallel watershed transformation algorithms for image segmentation. *Parallel Computing* 24(14), 1981–2001 (1998)
6. Roerdink, M., Meijster, A.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae* 41, 187–228 (2000)
7. Sun, H., Yang, J., Ren, M.: A fast watershed algorithm based on chain code and application in image segmentation. *Pattern Recognition Letters* 26(9), 1266–1274 (2005)
8. Rambabu, C., Chakrabarti, I.: An efficient immersion-based watershed transform method and its prototype architecture. *Journal of Systems Architecture* 53(4), 210–226 (2007)
9. Beucher, S.: The watershed transformation applied to image segmentation. *Scanning Microscopy International* 6, 299–314 (1991)
10. Stoev, S.L.: RaFSi - A Fast Watershed Algorithm Based on Rainfalling Simulation. In: 8th International Conference on Computer Graphics, Visualization, and Interactive Digital Media (WSCG 2000), pp. 100–107 (2000)
11. Lin, Y.-C., Tsai, Y.-P., Hung, Y.-P., Shih, Z.-C.: Comparison Between Immersion-Based and Toboggan-Based Watershed Image Segmentation. *IEEE Transactions on Image Processing* 15(3), 632–640 (2006)

Detection of Tumor Tissue Based on the Multispectral Imaging

Adam Świtoński^{1,2}, Marcin Michalak^{2,3}, Henryk Josiński^{1,2},
and Konrad Wojciechowski^{1,2}

¹ Polish-Japanese Institute of Information Technology
Aleja Legionów 2, 41-902 Bytom, Poland

{aswitonski,hjosinski,kwojciechowski}@pjwstk.edu.pl

² Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
firstname.name@polsl.pl

³ Central Mining Institute, Plac Gwarków 1, 40-166 Katowice, Poland
Marcin.Michalak@gig.eu

Abstract. We have prepared multispectral image database of skin tumor diagnosis. All images have been labeled with two classes - tumor and healthy tissues. We have extracted pixel signatures with their spectral data and class assigning, thus obtained train dataset. Next we have used and evaluated the supervised learning techniques for the purpose of automatic tumor detection. We have tested Naive Bayes, KNN, Multilayer Perceptron, LibSVM, LibLinear, RBFNetwork, ConjectiveRule, DecisionTable and PART classifiers. We have obtained results on the level of 99% classifier efficiency. We have visualized classification for example images by coloring class regions and verified if they overlap with labeled regions.

1 Introduction

Multispectral image contains spectral information of its every pixel, represented usually by several different values. It is much more than only data of three RGB components of color image; thus processing and recognition could give much better results. We can distinguish objects by their color, but such a simple technique is usually inaccurate, because of a lack of object boundaries in the color space - the different objects look very often almost the same. In such a case multispectral imaging could improve results. Even objects of the same color contain different spectrum.

If we know exact spectrum shape of the classified object types we can try to match spectrums based on the some kind of the distance metric. However such a technique is very often useless. Multispectral image gives inaccurate or even damaged spectrum data and requires some correction techniques, which usually are not able to restore the proper spectrum. Secondly, the same object type has no single spectrum and it differs in a set of its instances. For example in medical applications, spectrum of a given disease could differ in early, medium and advanced stage.

Segmentation and labeling of multispectral images is often performed by pixel based classification. It means that regions and their labels are determined by pixel description without spatial information. Such methods are useful because of the above mentioned object distinction in a spectral space. Examples of supervised and unsupervised learning techniques are presented in [14], [4], [2] and [10].

One of non-invasive methods of detection tumor tissues is photodynamic diagnosis. It utilizes the phenomenon of selective cumulation by tissue special colorant which absorbs the light of specified wavelength. It can be used for tumor detection of skin, oesophagus, endobronchial tract, stomach and others. On the basis of the assumptions and results of photodynamic diagnosis, we conclude that multispectral imaging could be able to separate tumor and healthy tissues.

2 Image Database

In multispectral acquisition we have used system (see Fig. 1) with a VariSpec liquid crystal tunable filter [8]. Such a filter is able to control electronically spectral transmission by keeping the band of selected wavelength and removing the rest, which is done by the proper polarization of crystal plates. Tunable filter is mounted in front of the Andor Luca high sensitivity monochrome camera. The camera takes photos of images with given single wavelengths, which form following multispectral channels. The whole process is controlled by special software with the direct access to the filter and the camera. Electromagnetic radiance passes through the endoscope, connected to our device. What is more it could also be used in gastroscopy and colanoscropy.

We have collected image dataset with 40 different images of a skin tumor diagnosis acquired at the Medical University of Silesia. They are described by medical experts, who pointed regions with tumors. The first 20 images have been captured with white light and next 20 ones, similarly as in photodynamic diagnosis, with blue light. White light and blue light images represent exactly the same

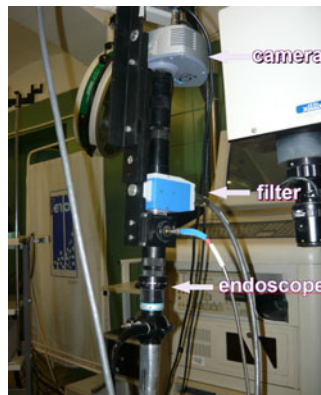


Fig. 1. Acquiring device

cases, in the acquisition process we simply switched external light connected to endoscope. The images contain 21 channels with wavelengths ranging from 410 to 710 nm. On the basis of the images we have prepared dataset of pixels signatures with their values from all spectral channels. We have randomly reduced dataset to 5000 instances, because of computational complexity. To divide our dataset into the train and test part we have used a cross validation with groups created by the pixels belonging to the same image. It is a real case in which we would like to make proper classification of the image based on the train dataset derived from other images. Pixels belonging to the same image have some kind of dependency, because they present the same kind of tumor or healthy skin.

To prove this thesis we have calculated standard deviation in the whole dataset and the average of standard deviations of all images, separately for both classes. The first one is several times larger, see Fig. 2.

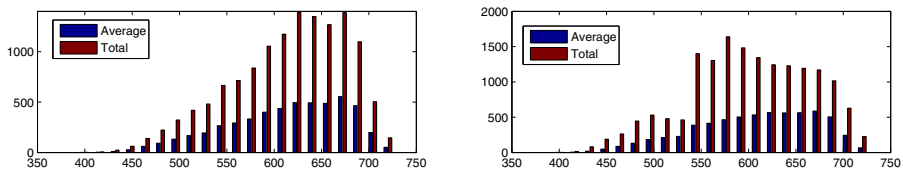


Fig. 2. Total and average of all images standard deviation of attributes in class healthy (left) and tumor (right) tissues

3 Algorithms

We have decided to test the following classifiers: NaiveBayes, KNN, Multilayer Perceptron, LibSVM, LibLinear, RBFNetwork, ConjunctiveRule, DecisionTable and PART.

3.1 NaiveBayes

Naive Bayes classifier determines class with highest probability of the given object [13]. To calculate that, it uses Bayes's rule of conditional probability, which needs densities of classes. They are estimated by the statistical methods based on the train dataset. Naive means that we naively assume independence of attribute in classes which reduce computational complexity, because it requires only separate estimation of densities for all attributes. There are two major approaches to density estimation in Naive Bayes. Normal estimator in which we assume normal distributions of attributes and calculate only means and variances in classes and kernel based estimator which uses kernel function.

3.2 KNN

KNN classifier stores all training examples and uses distance function to locate k nearest neighbors of a classified object. In the set of k nearest neighbors the class

with maximum number of instances is determined [13]. The crucial problem is the proper defining distance function and choosing k - the number of analyzed nearest neighbors. We have used Eucalidean distance function calculated on the normalized and raw features space. In fact KNN is Bayes classifier with kNN density estimator.

3.3 MultiLayer Perceptron

The basic component of the multilayer perceptron [13] is the perceptron often called neuron which transforms the inputs into single reply. It calculates linear combination of their inputs based on its weights and next activation function is applied. Multilayer perceptron is the network built from the layers of perceptrons in such a way that all outputs of a given layer are connected the all inputs of the neurons of the next layer. Inputs of the first layer take attributes of the classified object and the last layer outputs give a reply of the classifier. The training phase has to determine the weights of all neurons. If we know the reply of the neuron, which is true for the last layer, we can calculate error and update weights using gradient methods. If we only know errors of the front layer we can propagate it back proportionally to the weights. Such a process is repeated until stability or a given number of iterations is reached. The ability of the network to estimate decision areas depends on the complexity of the network structure and parameters of learning algorithms.

3.4 LibSVM

Support Vector Machines (*SVM*) were defined as the classification [1] and later also as the regression [5] tool. For the classification problem it assumes that there exists a hyperplane, separating objects into two class. If l is the number of x_i train vectors from the \mathbb{R}^n space, decision classes are described by values from the set $\{-1, 1\}$ and $[w, b]$ is the vector of hyperplane parameters, the standard model of classifier training leads to the following optimization formula: $\min_{w, b, \xi} \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^l \xi_i$ with constraints $y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0, i = 1, \dots, l$. ξ_i are slack variables which are the measure of the deviation from the linear separability and C is the upper limitation of the nonseparability error ($\langle \cdot, \cdot \rangle$ means product of two vectors). As the result of the dual optimization problem, for every train object x_i its Lagrange multiplier α_i is obtained and the final decision function for the classification problem takes the form $\text{sgn}(\sum_{i=1}^l y_i \alpha_i \langle x_i, x \rangle + b)$. As it is clearly seen, the classification result depends only on train objects with nonzero Lagrange multiplier values. These objects are called "support vectors".

The extension for linear nonseparability needs the kernel function K which for the pair of vectors u and v gives the value of product of vectors projections ($x_i \rightarrow \phi(x_i)$): $K(u, v) = \langle \phi(v), \phi(u) \rangle$. That leads to the decision function: $\text{sgn}(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b)$. The compromise between the number of support vectors and the nonseparability error can be kept in the modified *SVM* model. The μ parameter ($\mu \in (0, 1]$) is the upper bound for training error and the

lower bound of the support vector fraction. In this model the primal optimization problem has the form: $\min_{w,b,\xi,\rho} \frac{1}{2}\langle w, w \rangle - \nu\rho + \frac{1}{l} \sum_{i=1}^l \xi_i$ with constraints $y_i (\langle w\phi(x_i) \rangle + b) \geq \rho - \xi_i$ and $\xi_i \geq 0, l, \rho \geq 0$.

3.5 LibLINEAR

LibLINEAR [6] is a special implementation of SVM. It assumes that the OSH is linear – the kernel function is not used). Training of the classifier may be performed in several ways, which differ by the applied loss function of the classification error. The loss function can be linear (L1-loss SVM) or quadratic (L2-loss SVC). LibLINEAR is dedicated for large sets of high-dimensional data.

3.6 RBFNetwork

RBFNetwork is the classifier which is based on radial basis functions as the neural network activation functions. The radial basis function (RBF) meets the requirement that its value depends only on the distance between the argument and the selected point c that is called "base". If we assume that r is the distance from the base ($r = ||x - c||$) the most popular RBF can be defined as: $\rho(r) = \exp(-\beta r^2)$ (gaussian RBF), $\rho(r) = \sqrt{r^2 + \beta^2}$ (multiquadratic RBF), $\rho(r) = (r^2 + \beta^2)^{-1/2}$ (inverse multiquadratic RBF). The RBF neural network is a linear combination of several radial functions. It contains three layers: input – with the number of nodes equal to the number of objects attributes, hidden – with the number of nodes equal to the number of bases and output – with the number of nodes equal to the number of classes in the train set. The Weka RBFNetwork algorithm uses the k-means clustering algorithm to provide the basis functions and learns a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. If the class is nominal it uses the given number of clusters per class. It also standardizes all numeric attributes to zero mean and unit variance [13]. The following parameters can be specified as: the number of bases, the maximum number of logistic regression iterations and the minimum standard deviation for the cluster.

3.7 ConjunctiveRule

In this algorithm a single conjunctive rule which can predict the class label is built. The rule consists of the logical conjunction of the defined number of conditional descriptors. Each descriptor is the form `attribute <op> value`, where <op> can be the one `>`, `>=`, `<`, `<=`. If the test instance is not covered by this rule, then it is predicted using the default class distributions/value of the data not covered by the rule in the training data. This learner selects conditional descriptors by computing the Information Gain of each descriptor and prunes the generated rule using Reduced Error Pruning (REP) or simple pre-pruning based on the number of descriptors. The information of single descriptor is the weighted average of the entropies of both: the data covered and not covered

by the rule. The most important parameter of the algorithm is the number of conditional descriptor in the rule. Other parameters are the number of folds for pruning (the one is used for pruning and the other for growing the rule) [13].

3.8 DecisionTable

The problem of learning decision tables consists of selecting the right attributes to include. The common way is measuring the table's cross-validation performance for different subsets of attributes and choosing the best-performing subset. There are numerous ways of subset selection such as ExhaustiveSearch, BestFirst (climbing method) and a lot of heuristics. The Weka DecisionTable algorithm implements the Decision Table Majority model [9]. It summarizes the dataset with a 'decision table' which contains the same number of attributes as the original dataset. Then, a new data item is assigned to a category by finding the line in the decision table that matches the non-class values of the data item. DecisionTable employs the wrapper method to find a good subset of attributes for inclusion in the table. By eliminating attributes which contribute little or nothing to a model of the dataset, the algorithm reduces the likelihood of over-fitting and creates a smaller and condensed decision table [13].

3.9 PART

The PART algorithm [7] combines the advantages of C4.5 [11] and RIPPER [3] algorithms, and is also known as Divide-And-Conquer method. This method constructs a rule by dividing overly general rules into a set of rules, which correspond to conjunction subsets of the examples. It then continues recursively with those rules for which the corresponding subsets contain both positive and negative examples. The final rule set consists of all specialized rules for which the corresponding sets contain positive examples only.

4 Results and Conclusions

Table 1 presents obtained classification results for the best parameters set of every classifier. We have focused on the classifiers efficiencies in the meaning of the percentage of correctly classified pixels.

Except for Naive Bayes and ConjunctiveRule classifiers, which have obtained very poor efficiencies, white lighting gives better results. The reason for this is

Table 1. Classification results

algorithm	White	Blue
Naive Bayes	61,34%	63,36%
KNN	95,4%	64,4%
Multilayer perceptron	98,82%	70,34%
LibSVM	74,26%	72,75%
LibLINEAR	82,05%	70,00%

algorithm	White	Blue
RBFNetwork	94,48%	69,58%
ConjunctiveRule	60,05%	63,59%
DecisionTable	72,23%	65,12%
PART	88,5%	65,83%

that in general, it contains more spectral information. Because of much worse and in practice unacceptable results, we have omitted blue lighting in the further analysis.

The MultiLayer Perceptron best results on the level of 99% has achieved. KNN and RBFNetwork classifiers have also very high (almost 95%) efficiencies. Probably the nature of the problem requires decision areas of very irregular shapes, which are best modeled by these classifiers.

In Fig. 3 we have presented the influence of the number of hidden layer neurons and the number of learning cycles on the multilayer perceptron efficiency. The classification problem does not require complexed structure of the network, only 20 hidden neurons are sufficient. However, very important is the way of learning the network, which should be slow - small learning rate and lots of learning cycles.

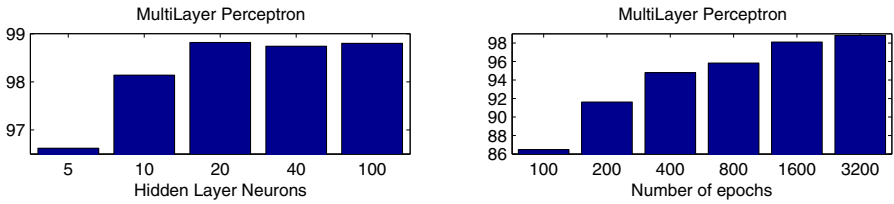


Fig. 3. Multilayer Perceptron parameters

Fig. 4 presents ROC curve [7] for multilayer perceptron. It is almost an ideal case of area under roc over 99.5%, which is the result of very high classifier efficiency.

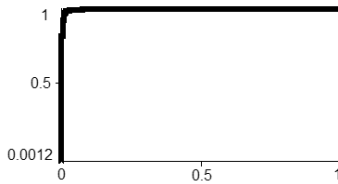


Fig. 4. ROC

In Fig. 5 we have visualized multilayer perceptron classification results for three multispectral images. The left one represents only healthy tissues, the medium one only tumor and the right one consists of some regions of healthy and other regions of tumor tissues. We have labeled images with two colors, based on the classification results - the red color represents the class of tumor and green one healthy tissues.

We can examine classification results for the whole tumor and healthy images which should be visualized by single proper color, respectively red and green. This is satisfied, except for some minor, very small regions. This mistakes are

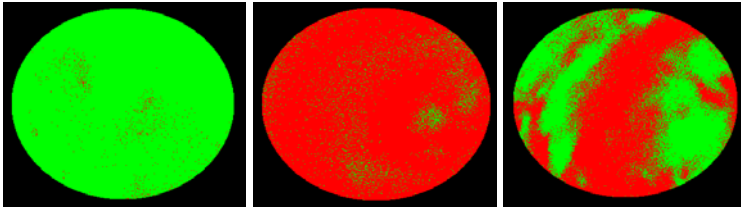


Fig. 5. Classification results visualization. Red - tumor tissues, green - healthy tissues.

probably caused by a noise of acquisition process and can be almost completely removed by postprocessing filtering, for example we may apply morphological closing and opening [12]. The image with tumor and healthy tissues is properly visualized with red and green regions. Similarly to previous examples we can notice some noise, which can also be removed by morphological filtering.

Our final conclusion is that supervised machine learning techniques are able to recognize tumor tissues based on the multispectral imaging.

Acknowledgment

This paper has been supported by the project „System with a library of modules for advanced analysis and an interactive synthesis of human motion” co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme - Priority Axis 1. Research and development of modern technologies, measure 1.3.1 Development projects.

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152 (1992)
2. Camps-Valls, G., Bruzzone, L.: Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 43(6) (2005)
3. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the 12th International Conference on Machine Learning, pp. 115–123
4. Derrode, S., Mercier, G., Pieczynski, W.: Unsupervised multicomponent image segmentation combining a vectorial HMC model and ICA. In: Proceedings of the IEEE International Conference on Image Processing (2003)
5. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.N.: Support vector regression machines. *Adv. in Neural Inf. Process. Syst.* IX, 155–161 (1997)
6. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871–1874 (2008)
7. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, pp. 144–151 (1998)

8. Gat, N.: Imaging spectroscopy using tunable filters: a review. In: Proc SPIE-Int. Soc. Opt. Eng. (2000)
9. Kohavi, R.: The Power of Decision Tables. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 174–189. Springer, Heidelberg (1995)
10. Mercier, G., Derrode, S., Lennon, M.: Hyperspectral Image Segmentation with Markov Chain Model. In: IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2003 (2003)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
12. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer, Heidelberg (2003)
13. Witten, I.H., Frank, E.: Data Mining. In: Practical Machine Learning Tools and Techniques, 2nd edn. Elsevier, Amsterdam (2005)
14. Woolfe, F., Maggioni, M., Davis, G., Warner, F., Coifman, R., Zucker, S.: Hyperspectral microscopic discrimination between normal and cancerous colon biopsies. IEEE Transactions on Medical Imaging 99(99) (1999)

A New Method for ECG Signal Feature Extraction

Adam Szczepański¹, Khalid Saeed¹, and Alois Ferscha²

¹ AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
Al. Mickiewicza 30, PL-30059 Cracow, Poland

{aszczepa,saeed}@agh.edu.pl

² Johannes Kepler University Linz, Institute for Pervasive Computing
Altenberger Strasse 69, 4040 Linz, Austria
ferscha@pervasive.jku.at

Abstract. The paper concentrates on electrocardiographic (ECG) signal analysis for the sake of acquiring proper heartbeat datasets for further detection and recognition of anomalies in an ECG signal. The analysis is based on the distribution of the voltage extreme values of the signal and the time distribution of proper extreme values. The main assumption of the work is creation of as efficient and fast algorithm as possible. The main part of the analysis is carried out on the smoothened signal with the original signal used as the reference for the purpose of keeping desired precision of proper heartbeat identification in the situations when during smoothening necessary data is lost. The proposed method of feature extraction will be used to create the distinctive normal heartbeat samples of patients and the average pattern of the patients' heartbeat for reference purposes in analysis of the real time signal obtained from 1-lead holter carried by a patient constantly.

Keywords: ECG signal, signal analysis, 1-lead holter, ECG feature extraction.

1 Introduction

First clear electrocardiographic (ECG) signal was obtained over a century ago in 1895 by Willem Einthoven [1]. Since then ECG analysis became a standard and a trusted technique of detecting cardiac defects. ECG measurement is now used both for monitoring and diagnosis as a detection method of vast range of anomalies and the examination lasts 24 hours and 12-lead holter is used as a measuring device [2].

In contrast to hospital diagnostics, studies described in this paper are intended to lead to a development of 1-lead holter which will be carried by people with high risk of myocardial infarction or other rapid life threatening cardiac injury to indicate alert immediately when the life threatening situation occurs. Therefore, during the first stage of the project, algorithms for isolating single heartbeats

and obtaining reference sequences of heartbeats to create the average pattern of ECG of the patient and the heartbeat profile of the patient from signal are proposed.

2 ECG Signal Representation

The schematic of a single heartbeat in ECG signal and the actual recording are presented in Fig. 1. The actual signal has a large number of local extreme values.

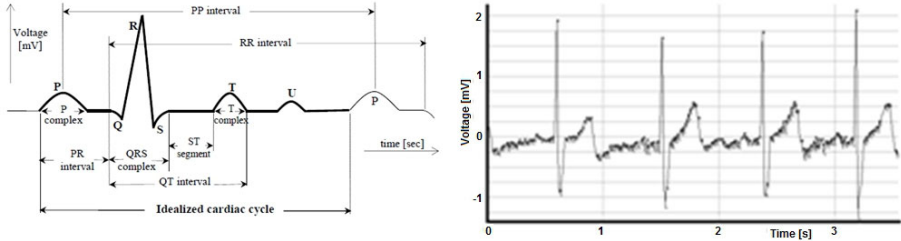


Fig. 1. Schematic (left) [3] and actual (right) ECG signals. On the actual signals extreme values are highlighted.

Points presented on the signal as P, Q, R, S and T are the main reference points in interpretation of ECG signal [4,5]. The U Wave which may appear in some people's heartbeats after T Wave may be skipped in further analysis [6].

What Could be Measured? The most important values to be measured are regularity of the signal, the distances between characteristic points and interferences in the shape of single heartbeat [4,7].

Signal may also contain artifacts which are generated by external factors such as the movement of electrodes and the chest muscles' activity. This should be detected and excluded before the analysis in order to identify the exact proper ECG signal position [2].

In the case of regularity of the signal R-R intervals, P-P intervals and bits per minute (bpm) are usually measured [4,8] and in the case of signal shape PR, QRS and QT interval could be measured and the extreme values of single heartbeat are checked for any changes in the average heartbeat. In many cases of cardiac diseases some parts of the signal, especially P Wave, is by-passed and is covered by T Wave from previous heartbeat. Some parts of the heart could also inversely polarize, leading to inverted parts of the signal [9,10]. Therefore the main goal of the real time heartbeat analysis shall be the detection of the anomalies in the normal rhythm by the detection of the characteristic signal points and calculation of their correlations. Parameters of ECG signal should be treated individually for every patient. Normal ECG values are described in [4,11]. In the system that will detect rapid cardiac malfunctions the main goal is to detect any rhythm of the heart and the shape of the signal that departs from average and to report the need of immediate medical assistance of the person.

3 Analysis of the Signal

This paragraph describes the algorithm based on particular sample analysis. The algorithm was tested on 15 recordings from [5] and 54 recordings (2 signals each) taken from [12]. Both the concept and the implementation of the algorithm are new and presented for the first time. The ECG signal feature extraction algorithm consists of 4 steps (Fig. 2).

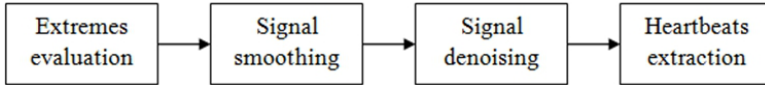


Fig. 2. The flowchart of the algorithm

If any constant time value is used in the algorithm as a constraint, it is calculated experimentally on the basis of the typical average ECG signal values [4,10]. All the millivolt values depend on the mV span of the input signal.

The signal used for demonstration purposes in this paper lasts 43 minutes and has the frequency of 200 samples per second. The signal has both regular and irregular heartbeats and some artifacts [5].

3.1 Smoothing of the Sample

Table 1 shows the distribution of the digital minima and maxima in the exemplary signal. The split of the signal into not more than 15 ranges was deduced experimentally. R and S points of heartbeats are clearly distinguishable but high density of extremes, previously shown in Fig. 1, leads to difficulties in detecting P and T. It is also noticeable that the position of Q point is not always obvious. That is why the smoothing of the signal is necessary.

Elimination of Strongest Artifacts. The range with most minima is chosen as the starting range of the true signal. Then the most probable span containing proper ECG signal is calculated using the method presented in Fig. 3.

The counter of the regions that does not meet the criteria was necessary because if the signal is clear, often the R and S points are located after about 2 range gap of regions without or with insignificant number of extremes so the algorithm should not stop the search when it approaches the first range that does not meet the criteria of number of extremes inside it.

The next step is the elimination of all the minima that are above minima range and all of the maxima that are below the maxima range. Then the ranges of minima and maxima are merged into one range and it is widened by 1/3 of the single range span.

The signal is analyzed again and all the samples that are not extremes and are not in the range are marked as deleted and the extremes are checked whether their values are in -3 mV to 3 mV range and, if so, another check is made to

Table 1. Distribution of minima and maxima of the exemplary ECG signal in 15 ranges. Bottom and Top describe the value range to which extremes belong, Mean is the mean value of the extremes in the range and Count is the number of extremes. Highlighted ranges are the most probable location of the proper signal. The 1 mV span has the range of 4927 digital units [13].

Minima total: 125555				Maxima total: 125545				
No	Bottom	Top	Mean	Count	Bottom	Top	Mean	Count
1	-25471	-22077	-25270	43	-25471	-22077	-25234	43
2	-22076	-18682	-20224	14	-22076	-18682	-20249	10
3	-18681	-15287	-16792	19	-18681	-15287	-16733	17
4	-15286	-11892	-13380	19	-15286	-11892	-13416	10
5	-11891	-8497	-9960	87	-11891	-8497	-10019	57
6	-8496	-5102	-5721	1909	-8496	-5102	-6142	174
7	-5101	-1707	-2592	8335	-5101	-1707	-2448	4740
8	-1706	1688	-343	105802	-1706	1688	-191	105662
9	1689	5083	2306	9076	1689	5083	2454	12032
10	5084	8478	6321	188	5084	8478	7341	664
11	8479	11873	9335	44	8479	11873	9515	2086
12	11874	15268	12946	9	11874	15268	12924	33
13	15269	18663	16811	7	15269	18663	16941	8
14	18664	22058	0	0	18664	22058	20660	3
15	22059	25455	23996	3	22059	25455	24283	6

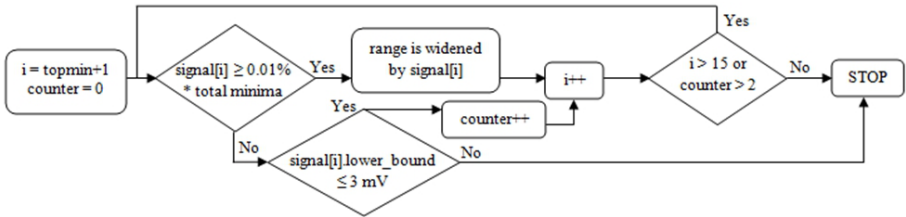


Fig. 3. Flowchart showing the appending of the signal range for the regions upper than the region with most minima. The same method, but without counter, is used for the regions lower than the region with most minima. For the maxima ranges the methods are the same. Topmin is the index of the region with most minima (8 in Table 1).

find out if the nearest neighbors of them are in the calculated range. If any of the nearest neighbors is not in the range, extreme is marked as deleted. Also whenever a point is marked as deleted, all the points in $(signal\ frequency)/5$ range on both sides of this point are marked as deleted. The reason of this is that high voltage artifacts have irregular shape or are often surrounded by lower voltage anomalies which are left in the signal if only points outside the range are eliminated, as shown in Fig. 4.

The Selected Method of Smoothing. When the signal true range is calculated and high voltage artifacts are removed, the signal is smoothed to lower the

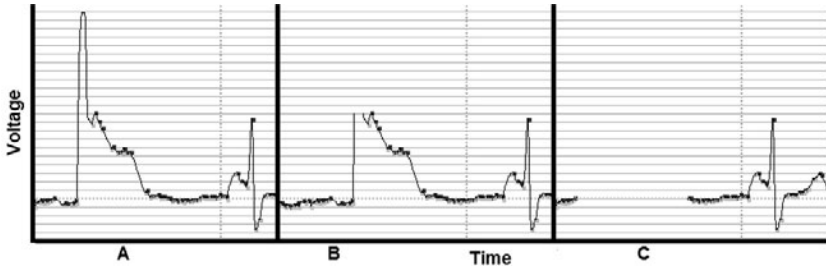


Fig. 4. The original signal (A), the signal after filtering of points that are not in range (B) and the signal after filtering also the neighbors (C)

number of extremes especially in low voltage part of the signal. For that purpose the variation of the method of bidirectional moving average [14] is chosen. The new sample value is calculated from (1):

$$x(i) = \frac{\sum_{k=-t}^t y(k)}{2 * t + 1}. \tag{1}$$

In (1) y is the original signal, x is the estimated signal and t is 1% of the signal frequency rounded up. The proposed method of estimating how many smoothing cycles should be applied is to smooth the sample as long as the change in the number of extremes before and after smoothing is less than 1% of the original number of extremes plus two more cycles. Also in the exemplary sample if the minimum value is lower than -0.5 mV or the maximum value is greater than 0.7 mV the extreme is not smoothed. These voltage constraints depend on the voltage span of the signal.

Table 2 shows the ranges left in the exemplary sample after the described procedure.

Table 2. Distribution of minima and maxima of the smoothed ECG signal

Minima total: 13262					Maxima total: 13407			
No	Bottom	Top	Mean	Count	Bottom	Top	Mean	Count
6	-8496	-5102	-5691	1743	-8496	-5102	-5972	39
7	-5101	-1707	-3264	2342	-5101	-1707	-2939	567
8	-1706	1688	-606	8988	-1706	1688	-218	7875
9	1689	5083	3309	138	1689	5083	2503	2382
10	5084	8478	6094	39	5084	8478	7705	497
11	8479	11873	0	0	8479	11873	9516	2033
12	11874	15268	0	0	11874	15268	12415	14

3.2 Searching for Heartbeats

After smoothing in the possible location of S and R points there may be more than one minimum and maximum (Fig. 5).

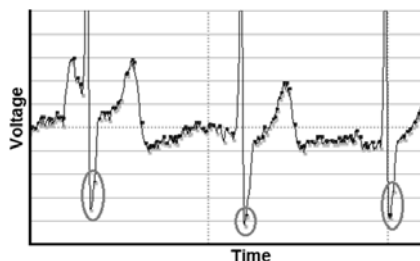


Fig. 5. Distribution of maxima and minima in the lower part of the signal. Ovals on the picture highlight the problem.

This problem is solved during the acquisition of R and S points by searching the top minimum value for S and maximum for R in the short span around the obtained point.

At first the algorithm first finds R-S pairs of the heartbeats. This is acquired by searching the signal for the pairs maximum-minimum which are not more than $1/8$ seconds away and the mV span of the pair is greater than 50% of the current signal range. After that the right side of the S point is checked for T point. T point is considered to be the biggest larger than 0 maximum which is not further than $1/3$ of the second away from R point. The left side of the R point is searched for the existence of Q point. Q point is considered as the lowest minimum in the range of $1/6$ second from R point but the mV span of Q-R pair should be not less than 60% of the span of the positive part of ECG signal. The P point is the largest maximum in the $1/4$ second span on the left side of the R point as the highest maximum before Q point.

3.3 Obtaining the Reference Signal

The signal is searched for the sequences of regular heartbeats. The average length of the heartbeat is calculated at first and then for all heartbeats the nearest neighbors from left and right are found. If there are any deleted earlier points between two heartbeats, this site of heartbeat is marked as not having the neighbor. Also all the distances greater than twice the signal frequency are eliminated. The shorter distances are checked whether between the end of the one signal and P point of the other signal there are any values less than the end point value or greater than P point value. If yes, the neighboring heartbeats are marked as not having neighbors on that side. Next the values of the heartbeat points are retrieved from the original signal. This is done by searching for the nearest maximum for P, R and T points and nearest minimum for Q and S points.

4 Results

Resulting number of properly acquired heartbeats is around 99.43% of total heartbeats in the signals. Results for exemplary samples as long as the results for all samples are shown in Table 3.

Table 3. The results

Name	Bits total	Detected beats	Rate of detection (%)
Worst 200 Hz sample	1322	1254	94.86%
Best 200 Hz sample	1976	1976	100%
Worst 1000 Hz sample	227	247	91.90%
Best 1000 Hz sample	3148	3148	100%
15 200 Hz samples	23454	23710	98.92%
108 1000 Hz samples	90013	89613	99.56%
Total (123 signals)	113072	90013	99.43%

The obtained results are comparable to the results of other popular algorithms like in [15] where 97.65% rate of detection was acquired, [16] with 99.85% rate of detection and [17] with 99.76% rate of detection.

On the contrary to those algorithms, the proposed algorithm does not use any filtering, wavelet transforms or slope detection techniques which leads to the acquisition of fast calculations' speed.

5 Conclusions and Future Work

The obtained efficiency of the algorithm is comparable to other commonly used algorithms. The main advantage of the proposed algorithm is that it does not need any other prefiltering than what may be provided by the signal measurement hardware. Also the lack of any software filtering means that when the system is working in real time it needs only a short (around 1 second length) buffer before it can start the analysis.

The next step of the studies shall lead to determining how long the ECG signal should be to extract the proper average of the heartbeats of the patient. Also fragments of the ECG signal should be connected with patient activities to determine correct patterns for all the possible stress and effort conditions.

This will eventually lead to the development of a 1-lead holter that would monitor the patient state constantly.

Acknowledgement

The research is supported within the FP7 project SOCIONICAL, No. 231288.

References

1. ECG Library, <http://www.ecglibrary.com/ecghist.html>
2. Adamec, J., Adamec, R.: ECG Holter. Springer Science+Business Media, LLC, New York (2008)
3. Gawroski, P., Saeed, K., Kuakowski, K.: Early warning of cardiac problems in crowd. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS (LNAI), vol. 6071, pp. 220–229. Springer, Heidelberg (2010)
4. Jones, S.A.: ECG Notes: Interpretation and Management Guide. F. A. Davies, Philadelphia (2005)
5. Riener, A., Ferscha, A., Aly, M.: Heart on the road: HRV analysis for monitoring a driver's affective state. In: Proceedings of the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009), pp. 99–106 (2009)
6. Stouffer, G.A.: Practical ECG Interpretation: Clues to Heart Disease in Young Adults. Wiley-Blackwell, UK (2009)
7. Van Mieghem, C., Sabbe, M., Knockaert, D.: The Clinical Value of the ECG in Noncardiac Conditions. *Chest* 125, 1561–1576 (2004)
8. Trahanias, P., Skordalakis, E.: Syntactic pattern recognition of the ECG. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 648–657 (1990)
9. Hampton, J.R.: The ECG Made Easy, 6th edn. Churchill Livingstone, New York (2003)
10. Khan, M.G.: Rapid ECG Interpretation, 3rd edn. Humana Press, New Jersey (2008)
11. Hung-Chi, L.: ECG in the Child and Adolescent. Blackwell Publishing, Massachusetts (2006)
12. Non-Invasive Fetal Electrocardiogram Database, mothers' part of the signal was used for analysis, <http://www.physionet.org>
13. Kemp, B., Vrri, A., Rosa, A.C., Nielsen, K.D., Gade, J.: A simple format for exchange of digitized polygraphic recordings. *Electroencephalography and Clinical Neurophysiology* 82, 391–393 (1992)
14. Chou, Y.: Statistical Analysis. Holt Reinhart Winston, New York (1975)
15. Haque, M.A., Rahman, M.E., Al Sayeed, C.A., Uddin, B.M.Z.: A Fast Algorithm in Detecting ECG Characteristic Points. In: Proceedings of Second International Conference on Electrical and Computer Engineering (ICECE 2002), Dhaka, Bangladesh, December 26–28, pp. 160–163 (2002)
16. Cuiwei, L., Chongxun, Z., Changfeng, T.: Detection of ECG Characteristic Points Using Wavelet Transforms. *IEEE Transactions on Biomedical Engineering* 42, 21–28 (1995)
17. Chiarugi, F., Sakkalis, V., Emmanouilidou, D., Krontiris, T., Varanini, M., Tollis, I.: Adaptive Threshold QRS Detector with Best Channel Selection Based on a Noise Rating System. In: Computers in Cardiology 2007, pp. 157–160 (2007)

Error Concealment Method Selection in Texture Images Using Advanced Local Binary Patterns Classifier

Želmira Tóthová, Jaroslav Polec, Tatiana Orgoniková, and Lenka Krulikovská

Department of Telecommunications, Slovak University of Technology
Ilkovicova 3, 812 19 Bratislava, Slovakia
polec@ktl.elf.stuba.sk, florekova.zelmira@gmail.com

Abstract. There are many error concealment techniques for image processing. In the paper, the focus is on restoration of image with missing blocks or macroblocks. In recent years, great attention was dedicated to textures, and specific methods were developed for their processing. Many of them use classification of textures as an integral part. It is also of an advantage to know the texture classification to select the best restoration technique. In the paper, selection based on texture classification with advanced local binary patterns and spatial distribution of dominant patterns is proposed. It is shown, that for classified textures, optimal error concealment method can be selected from predefined ones, resulting then in better restoration.

Keywords: Error concealment, re-synthesis, inpainting, texture, extrapolation, classification.

1 Introduction

The transmission of images coded by block or segment based techniques via a noisy channel [5] may lead to block or segment loss. Therefore error detection and concealment at the decoder side has to be applied. Commonly, spatial error concealment is used. It utilizes the surrounding correctly received image information to restore the damaged or missing pixels. A standard approach [4] assumes that the image content is changing smoothly. Hence the algorithm tries to restore the transition across the block boundary as smooth as possible. The extrapolation-based method of [9] tries to reconstruct the missing pixels as a weighted linear combination of correctly received pixels. Hence, the method is computationally very complex. The transmission of block-coded image data via wireless channel is described in [6]. Very interesting and novel method for spatial error concealment based on successive extrapolation of missing blocks is described in [7].

The texture of an image might be defined broadly as the interrelationships between pixels in that image. The ability to analyze and manipulate image texture has a number of interesting applications. The simplest application is to create

a new image with the same texture but of different size and shape to a sample image. Seamless editing of images is also a possibility. These applications all rely on the ability to re-synthesize a sample texture to fit a variety of constraints. In automated systems, it is difficult to decide which of the methods offered would be the most successful one. As selection method, one of the texture classification methods is proposed here. The Brodatz texture database [10], VisTex [11] and Simoncelli personal database [12] were used as a source.

2 Synthesis

2.1 Non-hierarchical Procedure for Re-synthesis of Complex Textures

Procedure for synthesis of an image with the same or similar texture as a given input image was proposed by Harrison in [3]. To achieve this, the output image is built up by successively adding pixels selected from the input image. Pixels are chosen by searching the input image for patches that closely match pixels already present in the output image. The accurate reproduction of features in the input texture depends on the order in which pixels are added to the output image. A procedure for selecting an ordering which transfers large complex features of the input to the output image is described. This procedure is capable of reproducing large features even if only the interactions of nearby pixels are considered.

2.2 Discrete Orthogonal Transforms for Gappy Image Extrapolation

Papers [12][8] present different methods for approximation of non-square areas. The texture of an area is successively approximated and then cut to the shape of the segment. The same principle is used in this paper for error concealment by estimating the missing image content using the surrounding area belonging to the same object, thus (supposedly) having the same texture pattern. The image content of the known segments is successively approximated and the missing segment is obtained by extrapolation [9] (also [7]).

2.3 Object Removal by Exemplar-Based Inpainting

Algorithm is proposed for removing large objects from digital images. The challenge is to fill in the hole that is left behind in a visually plausible way. In the past, this problem has been addressed by two classes of algorithms: "texture synthesis" algorithms for generating large image regions from sample textures, and "inpainting" techniques for filling in small image gaps. The former work well for "textures" - repeating two dimensional patterns with some stochastic; the latter focus on linear "structures" which can be thought of as one-dimensional patterns, such as lines and object contours. [6] presents algorithm that combines the advantages of these two approaches. Different approaches with similar goals could be also included to inpainting [24].

3 Classification

In this part, the requirements for classifier selection will be defined and the classifier will be selected. Invariance towards external effects as rotation, scaling, illumination, etc. are usually not required for similar applications. Naturally, a model with some of these characteristics will be preferred, if it does not result in substantial increase of time or memory demands of the algorithm. Computational complexity will be one of the important factors. The most important attribute for decision will be the success rate of classification. Since for the classification algorithm here, usually no requirements are strictly defined, we will select the model that solves the problem as good as possible, which is the representation that is computationally effective as well as it guarantees high success rate of the classification.

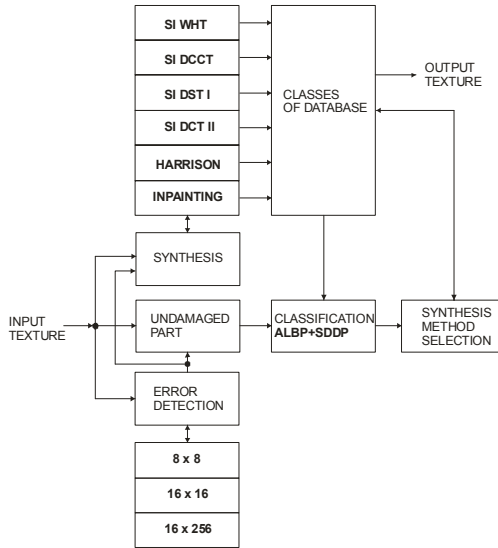


Fig. 1. Masking method selection procedure. (SI - shape independent transform).

If we consider conclusions from [13,14,15,16,17,19,20], the best results from the success rate point of view were acquired using Markov random fields (in particular, VZ-classifier), local binary patterns and advanced local binary patterns with spatial distribution of dominant patterns. Success rate of the classification approached 100%. From other methods, model for DCT coded images could be mentioned. This representation should be used in applications, where DCT coded images should be classified. For photorealistic images, PCA and neural networks [18] are successfully used for classification. Local binary patterns (LBP) have one great advantage compared to VZ-classifier. It is their invariance feature regarding rotation and histogram equalization. Rotation invariance results from

Table 1. Classifier Selection (dropped part 16x256 pxs),best results are shaded out

	checker board			jdice			D3		
	SI DOT	Harri son	Inpain ting	SI DOT	Harri son	Inpain ting	SI DOT	Harri son	Inpain ting
PSNR	48,46	9,39	34,29	15,87	14,74	10,7	11,73	11,26	11,2
ALBP	X					X	X		
DCT D	11,58	10,99	11,53	2,16	1,38	0,67	0,62	0,62	0,59
DCT ED	32,17	29,56	32,21	5,6	4,65	2,06	2,14	2,02	1,87
SB16 D	3,81	6,83	4,07	2,5	1,58	0,75	0,48	0,5	0,54
SB16 ED	9,54	9,32	9,43	5,92	4,98	2,24	2,06	1,93	1,82
SB7 D	1,26	1,9	1,32	1,1	0,82	0,29	0,19	0,19	0,18
SB7 ED	9,28	8,99	9,17	5,92	4,96	2,18	1,97	1,83	1,68
	grains			bark			bullseye		
	SI DOT	Harri son	Inpain ting	SI DOT	Harri son	Inpain ting	SI DOT	Harri son	Inpain ting
PSNR	8,53	8,56	8,85	12,73	12,77	12,27	37,41	11,19	12,93
ALBP			X	X			X		
DCT D	1,18	1,1	1,14	1,37	1,43	1,33	3,81	3,85	4,11
DCT ED	3,74	3,53	3,52	2,82	3,28	2,71	10,11	10,09	10,42
SB16 D	1,15	0,99	1,09	0,86	1,12	0,87	3,97	5,18	5,59
SB16 ED	3,2	3,14	2,69	2,44	3,2	2,45	9,46	9,44	9,8
SB7 D	0,37	0,35	0,33	0,32	0,5	0,34	1,82	2	2,1
SB7 ED	3,03	2,98	2,45	2,25	3,09	2,27	9,23	9,17	9,53

used operator, which, in turn, generates computational requirements increase, because for each computed local binary pattern, the one with the lowest value must be found, which means to do bit shifting. Invariance towards histogram equalization comes directly from the definition of LBP operator.

If for two neighboring pixels i and j holds $x_i \neq x_j$, where x_i is luminance for pixel i , this inequality holds also after histogram equalization. And, as LBP operator is based on these inequalities, histogram equalization has no influence. On the contrary, it is mentioned in [23], that for using VZ-classifier in preprocessing phase, it is necessary to do image luminance normalization to get invariance towards linear variations of the luminance within all images.

Therefore, invariance towards rotation or histogram equalization for LBP would not bring increase of algorithm time demand. Another advantage of LBP compared to VZ-classifier is, that components of feature vector for LBP are known before, on the other hand, for VZ-classifier are acquired using k-means clusterisation in training phase, which slows the process down. If we decide to use LBP for representation, either local binary patterns as used by Ojala [20], or ALPB with spatial distribution of dominant patterns from Liao and Chung [14] might be employed. The difference between them is using the spatial distribution of dominant patterns. To create representing grey level aura matrix is time and memory consuming operation, but it can bring significant increase of probability of correct classification. The decision is very subjective. For this research, ALBP classifier was selected. Computational complexity can be decreased using only one neighborhood. In [20], LBP was tested with more types of LBP operator. In some cases, combination of more neighborhood types was used, leading to success rate increase. In [14], more than 99% of textures were correctly classified. Unfortunately, the paper does not specify what type of the neighborhood

was used. LBP model using 3x3 neighborhood reached success rate slightly over 80%. The certain improvement can be brought by feature of advanced local binary patterns, that is, to create a feature vector, where not only uniform patterns are used, but also 20 of the most frequent from all patterns.

4 Solution

To select a method for error masking we have two possibilities: 1. To classify a damaged texture correctly - define a class, for which one of the masking methods is substantially more effective compared to others, and proceed as [22] or 2. To use all available error masking methods and use corrected images as classifier training. Corrected image with one classification method is a class. For testing, undamaged part of the image is used. Corrected image is classified based on the classification of undamaged part.

Training on Brodatz texture database [10], procedure 1. results were very unconvincing, therefore we decided to use procedure 2. (Fig. 1). Software classifier [21] was used, that is implementation of [14]. Verification the accuracy of the classifier selection by method 1. was done for images with masking of dropped image part of size 16x256 pixels. Results are presented in Tab.1 and are evaluated by objective criterion PSNR used for reconstructed images and by objective criterion commonly used for the classifier. D is Euclidean distance weighted by standard deviation in the particular band and ED is Euclidean distance. SB (sub - band) is modification of DCT procedure, however it is realized in time, not spectral domain. SB16 stands for frequency equal distribution into 16 sub-bands, SB7 is standard distribution to 7 sub-bands as it is common in wavelet spectral decomposition.

The classifier is trained by corrected textures and appropriate classes are created. Then we test undamaged image parts of different sizes to determine to which class they belong. In future it will be interesting to analyze the influence of size, location and number of image parts on the success of classification as it is common for procedures, where texture classification is used. As we can see from Tab. 1, ALBP classifier does not correspond to the best results according to PSNR in two of six cases, however the results for texture "bark" are very similar for all three error masking so we can declare that ALBP classifier does not correlate with the best results only in one of six cases. In contrast, result from spectral classifiers show higher rate of an uncertainty.

5 Results

Depending on the way how the texture images were acquired, they can be divided into natural (photographic) and artificial images. One of the most common divisions of texture images is based on geometric adjustment of inside objects. By periodicity, textures are divided to [12]:

- periodic - artificial texture images generated according to the specified rules;
- pseudo-periodic - artificial textures generated with no specific rules or natural texture images;
- non-periodic - artificial textures with random added patterns or natural patterns with random or structured placed patterns.

When the images are sent through the channel using block-oriented compression techniques, they are error prone. Images received can contain errors. In the experiment we examined both types of errors - missing isolated blocks and group of missing successive blocks. For the block loss, reconstruction techniques for missing 8x8 or 16x16 blocks were examined. For group of successive blocks, we can consider loss of blocks across the image, therefore dropped image part of size 16x256 pixels. Tab. 2 shows the results on selected textures from each of the mentioned classes when using three listed masking approaches and others error types. ALBP was used for classification. The some images are shown on Fig. 2, Fig. 3 and Fig. 4.

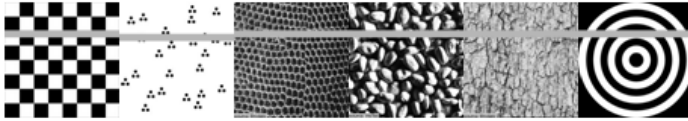


Fig. 2. Damaged images: (a) checkerboard, (b) jdice, (c) D3, (d) grains, (e) bark, (f) bullseye

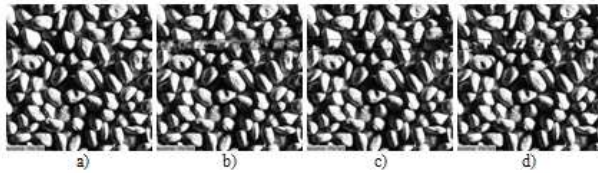


Fig. 3. Missing strip of blocks reconstruction: (a) original picture grains and results: (b) transformational extrapolation, (c) re-synthesis and (d) inpainting

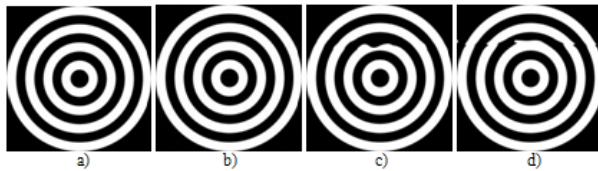


Fig. 4. Missing strip of blocks reconstruction: (a) original picture bullseye, (b) transformational extrapolation, (c) re-synthesis and (d) inpainting

Table 2. Results on selected textures (X denotes classif. result)

		DCT	Harrison	Inpainting	DCT	Harrison	Inpainting
		8x8	8x8	8x8	16x16	16x16	16x16
checkerboard	PSNR	42.21	infinity	infinity	47.63	38.581	infinity
	ALBP		X		X		
jdice	PSNR	13.65	24.24	29.95	6.20	6.78	28.13
	ALBP			X			X
D3	PSNR	17.00	19.16	13.57	13.78	15.49	15.55
	ALBP	X					X
grains	PSNR	15.36	13.79	13.47	10.10	13.10	12.36
	ALBP	X				X	
bark	PSNR	13.90	13.06	13.63	11.19	13.05	11.26
	ALBP	X				X	
bullseye	PSNR	36.33	38.94	36.02	38.44	13.56	30.14
	ALBP	X			X		

6 Conclusion

The experiment shows, that reverted procedure using the classifier for masking purposes is a suitable method for evaluation of results from different kinds of texture synthesis algorithms. Therefore, the distance of feature vectors can be considered as a suitable objective criterion to substitute MSE or PSNR especially in cases, where these criteria cannot be used because the parts of image are missing. As a result we propose this method for selection of masking procedure as suitable one.

Acknowledgments. Research described in the paper was financially supported by the Slovak Research Grant Agency (VEGA) under grant No. 1/0883/08.

References

1. Gilge, M., Engelhart, T., Mehlan, R.: Coding of Arbitrary Shaped Image Segments Based on a Gen. Orthogonal Transform. *Image Communication* 1, 153–180 (1989)
2. Kaup, A., Aach, T.: Coding of Segmented Images Using Shape-Independent Basis Functions. *IEEE Trans. on Image Processing* 7(7), 937–947 (1998)
3. Harrison, P.: A non-hierarchical procedure for re-synthesis of complex textures. In: WSCG 2001, Plzen, Ceska republika, pp. 190–197 (2001)
4. Wang, Y., Zhu, Q.F., Shaw, L.: Maximally smooth image recovery in transform coding. *IEEE Transactions on Communications* 41(10), 1544–1551 (1993)
5. Kotuliakova, K.: Hybrid ARQ Methods in Wireless Communication Channels, PhD. Thesis, SUT, Bratislava (2005)
6. Criminisi, A., Perez, P., Toyama, K.: Object Removal by Exemplar-Based Inpainting. In: *Proceedings of CVPR 2003*, pp. 721–728 (2003)
7. Meisinger, K., Kaup, A.: Spatial error concealment of corrupted image data using frequency selective extrapolation. In: *Conf. Rec. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, May 17–21, pp. III-209–III-212 (2004)
8. Polec, J., et al.: New Scheme for Region Approximation and Coding with Shape independent Transform. In: *Proceedings IAPRS*, vol. XXXIV, part 3A/B, pp. B214–B217 (2002)

9. Polec, J., Karlubikova, T.: Discrete Orthogonal Transform for Gappy Image Extrapolation. In: Proceedings of ICCVG 2004, Warsaw, Poland. Computational Imaging and Vision, vol. 32, pp. 222–227. Springer, Heidelberg (September 2004)
10. Brodatz, P.: Textures, a photographic album for artists and designers. Dover Publications Inc., New York (1966), <http://www.ux.his.no/~tranden/brodatz.html>
11. Vision Texture, <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>
12. Portilla, J., Simoncelli, E.: Representation and Synthesis of Visual Texture, <http://www.cns.nyu.edu/~eero/texture/>
13. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 691–698 (June 2003)
14. Liao, S., Chung, A.C.S.: Texture Classification by using Advanced Local Binary Patterns and Spatial Distribution of Dominant Patterns. In: ICASSP 2007, pp. I-1221– I-1224 (2007)
15. Bautista, P.A., Lambino, M.A.: Co-occurrence matrices for wood texture classification. Electronics and Communication Department, Iligan Institute of Technology (2001)
16. Partio, M., Cramariuc, B., Gabbouj, M., Visa, A.: Rock Texture Retrieval using Gray Level Co-occurrence Matrix. In: NORSIG 2002, Trollfjord, Norway, October 4-7 (2002)
17. Huang, Y.-L., Chang, R.-F.: Texture features for DCT-coded image retrieval and classification. In: Proceedings of Acoustics, Speech, and Signal Processing, ICASSP 1999, March 15-19, vol. 6, pp. 3013–3016 (1999)
18. Oravec, M., Pavlovicova, J.: Face Recognition Methods Based on PCA and Feedforward Neural Networks. In: Proc. of the IJCNN 2004, Budapest, vol. 1, pp. 437–442 (2004)
19. Vyas, V.S.: Priti Rege: Automated Texture Analysis with Gabor filter. GVIP Journal 6(1), 35–41 (2006)
20. Ojala, T., Pietikainen, M.: Texture classification, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm
21. Apalovic, L.: Texture Classification. Thesis, UK Bratislava (2009)
22. Polec, J., Pohancenik, M., Ondrusova, S., Kotuliakova, K., Karlubikova, T.: Error Concealment for Classified Texture Images. In: EUROCON 2009, Saint Petersburg, Russia, pp. 1348–1353 (2009)
23. Blunsden, S.: Texture Classification using Non-Parametric Markov Random Fields. University of Edinburgh, School of Informatics (2004)
24. Arias, P., Caselles, V., Sapiro, G.: A Variational Framework for Non-local Image Inpainting. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMCCVPR 2009. LNCS, vol. 5681, pp. 345–358. Springer, Heidelberg (2009)

Early Warning System for Air Traffic Control Using Kinetic Delaunay Triangulation

Tomáš Vomáčka and Ivana Kolingerová

University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic
{tvomacka,kolinger}@kiv.zcu.cz

Abstract. We describe the current methodology used in the air traffic control and show how the kinetic Delaunay triangulation may be used to predict potentially dangerous situations of two aircrafts on a crash course. We use a real data obtained from the radar readings of the air space of the Czech Republic and process this data in order to create and manage the triangulation. We also show how the alternation of certain application-specific parameters may change the overall performance of the triangulation.

1 Introduction

In the field of air traffic control it is vitally important to be able to detect potentially dangerous situations in such an advance that the upcoming danger can be avoided. Currently the air traffic is monitored and controlled mostly by human operators. Each of these operators obtains a certain (rather small) subset of the air traffic data provided by various tracking devices such as radars, etc. Our approach provides a global perspective of the problem. By analyzing the whole set of aircrafts detected by the radar at once, we are able to detect potential threats before any other postprocessing is applied to this set and to detect such situations when two aircrafts are on a crash course.

In order to detect these cases, we exploit a data structure called kinetic Delaunay triangulation (KDT). This data structure connects the advantages of an ordinary Delaunay triangulation or DT [1], with the ability to contain moving points. Among the various features of DT, we use namely the fact that all the points in DT are connected with their nearest neighbors. Combined with the movement of the points, it provides us with an efficient collision detection system [2]. The KDT was already used by Gold and others for similar purpose in a marine environment [3,4]. However, the use in the aeronautical applications differs in some key features. Furthermore, during the implementation of our application we tested several theoretical expectations of the KDT behavior.

Even though our method represents a global view on the problem and thus is not suitable for use in the current air traffic control, it shows a possible way of development in this field. Should the currently used approach be replaced by some kind of a global control system, our application would represent one of

the candidates for the new system. The paper is organized as follows: Section 2 gives the basic overview of some of the currently used practises in the air traffic cont. Section 3 provides details about the geometrical aspects of the problem. Section 4 shows results obtained by measuring the output of our test application. Finally, a conclusion of our work is given in Section 5.

2 Air Traffic Control Systems

2.1 Current Methods and Conventions

According to [5], the operators who control the movement of the aircrafts in certain portion of the air space are each assigned only a small portion of the nearby traffic. Thus each human operator is responsible only for approximately ten aircrafts. When an aircraft lands or leaves the space assigned to a certain operator, that operator's control over this aircraft is lost or transferred to some other operator respectively.

Furthermore, the aircrafts move only through strictly defined corridors called routes (except some special cases), which are defined differently for different kinds of aircrafts (e.g., helicopter routes may differ from international flight routes). The routes also differ in accessibility – certain routes are only available for some heights, visibility conditions, etc.

The predefined heights on which the aircrafts travel along the routes are called flight levels. The flight levels, even though they are defined precisely, vary with changing atmospheric pressure. This effect is caused by the definition of the flight levels. The flight level is a level of constant atmospheric pressure which causes the aircrafts to flight in different heights for different atmospheric conditions even though the aircrafts are located on the same flight level. Further information on the routes, flight levels and other air traffic rules may be obtained in [6].

2.2 Air Traffic Data Sets

The data set we used in our application was obtained from the Air Navigation Services of the Czech Republic. It was generated from a record obtained by logging real radar entries during standard traffic. Upon detection, the radar data was processed by a tracker device which supplied them (among others) with aircraft-unique identifiers thus ensuring that when a single aircraft appears more than once in the record, it will be uniquely identifiable each time it reappears.

The data in the set are only sorted timewise – the records are logged as the radar detects them. Each record contains the location, height, heading and velocity information about one plane. Note that the data set only contains information about the *current* whereabouts of a given aircraft. No information about the overall aircraft trajectory is provided. This fact is a consequence of the means of obtaining the data and although it may seem restricting, it correlates very well with the data structure we use, because our implementation of KDT only allows linear trajectories of the moving points (see further).

3 Geometric Features of the Problem

3.1 Problem Overview

Let us briefly summarize how we handle the given problem with our application. As we may see in Fig. 1, the application may be divided into four steps. In the first step the radar provides us with aircraft data; these data are then projected into the Euclidean plane (step 2), a kinetic Delaunay triangulation is constructed over the projection (step 3) and used for detecting possible collisions (step 4). The following text describes steps 2 and 3.

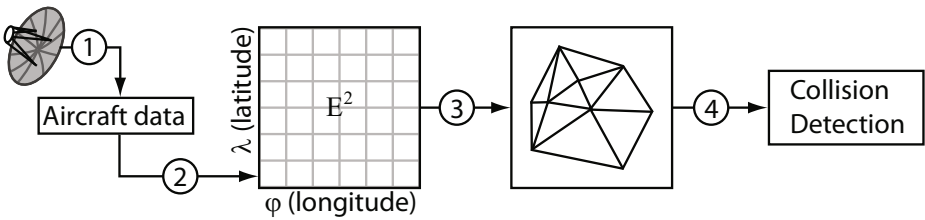


Fig. 1. Block scheme of our application

3.2 Aircraft Movement Mapping to 2D

Due to the restrictions forced upon the aircraft movement (the flight levels and routes), the problem of aircraft navigation is in fact (simply said) only a 2.5D problem. As such, it may be managed by exploiting a planar triangulation with height values assigned to each of the planes. The triangulation will only use the latitude and longitude coordinates of the planes to check for collisions. If a collision is then detected in the 2D projection, the relevant planes will be checked in full 3D and the collision warning will be proposed further only if the collision will really occur.

For the purposes of our application, we used a simple planar projection using *WGS84* model of the Earth for the coordinate transformation [7] which is the most commonly used model in this field of expertise. Another problem related to the aircraft mapping to 2D is that the aircrafts do not fly along linear trajectories. Using the newest available data from the radar readings for each aircraft and one of the mentioned Earth models, we predict the next position of the aircrafts by using a simple linear extrapolation. Fig. 2 shows the difference between the actual path of an aircraft (solid black line, denoted p_{act}) and the path predicted using the linear extrapolation (dashed gray line, denoted p_{pred}).

3.3 Kinetic Delaunay Triangulation

Given a set of n planar points $S = \{P_1, \dots, P_n\}$, Delaunay triangulation $DT(S)$ constructed over S is a maximum set of edges E such that (see [1]):

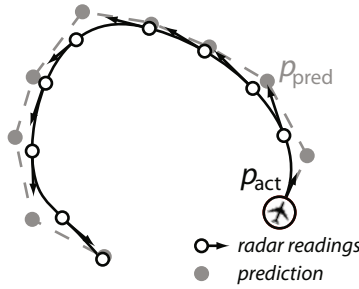


Fig. 2. Aircraft path prediction

1. No two edges in E intersect in a point not in S .
2. The edges in E divide the convex hull of S into triangles.
3. No point in S lies inside a circumcircle of any of the triangles defined by E (*Delaunay criterion*).

Let us define the points P_1, \dots, P_n such that:

$$P_i(t) = [x_i(t), y_i(t)] \tag{1}$$

$$x_i(t) = x_i(0) + v_i^x \cdot t, y_i(t) = y_i(0) + v_i^y \cdot t \tag{2}$$

where $P_i(0) = [x_0, y_0]$ is the initial position of point P_i and $\mathbf{v}_i = [v_i^x, v_i^y]$ is the velocity vector of P_i – i.e., the points move along linear trajectories without acceleration. If there is at least one point $P_i \in S : \|\mathbf{v}_i\| \neq 0$ then, according to [8], the topology of the triangulation will have to change because the Delaunay criterion will become violated by the movement of the points in S . These changes occur at specific time instants which are called topological events. A topology change always occurs over a pair of adjacent triangles and is handled by swapping the edge that is common to those triangles. Example of this behavior is given in Fig. 3 where P_4 moves inside the circumcircle of $P_1P_2P_3$. Note that the algorithm for KDT management is not discussed in detail here due to its extensiveness. It may be found in literature, e.g., in [2,8,9].

When $KDT(S)$ is created and when its topology changes during the process of handling a topological event, some new topological event times are computed.

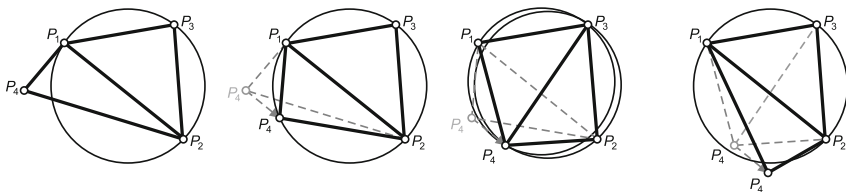


Fig. 3. Topological event

These newly computed events are then stored in a priority queue where they are ordered according to the times when they will occur. If the points move as defined in Eq. 2, the computation of the times of topological events will result in solving a polynomial of degree up to four [9]. It is also important to note that not all of the events need to be computed – for instance if we know that a topological event will occur between the triangles $P_1P_2P_3$ and $P_2P_4P_3$ at $t = t_e$ (see Fig. 4) it is not necessary to compute events between $P_2P_4P_3$ and $P_3P_4P_5$ if we know that none of them will occur for $t < t_e$. The process of computation of the topological events is the most time-consuming part of the KDT management and it is vital to reduce the number of computed topological events as much as possible. One possible way to reduce the amount of computed events is by using the Sturm sequences of polynomials [10]. Then we are able to quickly determine how many topological events will occur for a given triangle pair during any given time interval [9].

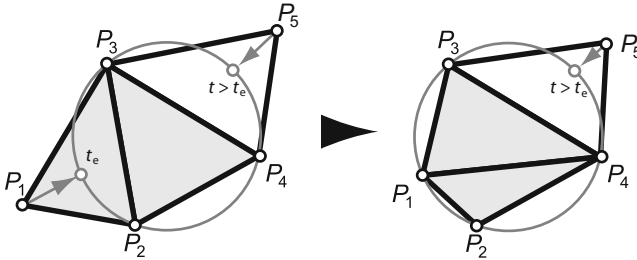


Fig. 4. Computation of redundant topological events

Apart from the topologic events, the topology of the triangulation will also need to be changed when a new information about an aircraft is obtained from the tracker. This will happen every time when the newly obtained position $P_q(t + \Delta t)$ differs from the position precomputed by extrapolation $P_q(t) + \mathbf{v}_q \cdot \Delta t$, where t is the time variable and Δt is the time step between two trackers readings of the same aircraft (see Fig. 2) by more than a given tolerance $\tau \geq 0$ as shown in Eq. 3.

$$\frac{\|P_q(t + \Delta t) - (P_q(t) + \mathbf{v}_q \cdot \Delta t)\|}{\|P_q(t + \Delta t)\|} \leq \tau \tag{3}$$

If the condition in Eq. 3 is satisfied, we modify the point $P_q(t + \Delta t)$ so that its velocity vector remains unchanged (i.e., we use the newest information obtained from the radar at t_{i+1}) and its position is set to $P_q(t) + \mathbf{v}_q \cdot \Delta t$ (i.e., the position predicted at time t_i). The error generated by this approach may increase with each iteration but it will be corrected as soon as it reaches the value given by τ . To do so, we remove the point from the triangulation (by using a removal algorithm similar to e.g., [11]) and reinsert it back on the correct position.

4 Results

As said before, the triangulation behavior will be different if we allow some small inaccuracy τ of the position of the aircrafts. Our experiments show that τ should be several percent at most – this will allow us to only update the velocity vectors of the aircrafts as new radar readings are obtained by the application thus increasing the performance of the application as shown in Fig. 5.

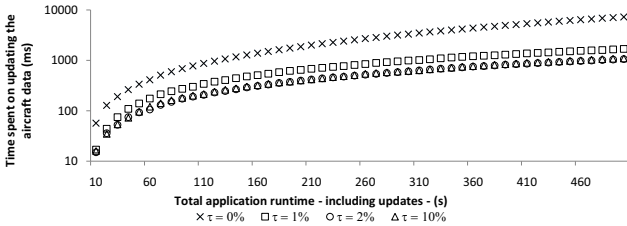


Fig. 5. Comparison of the performance for various values of τ

Introducing $\tau > 0$ reduces the time consumed by updating the aircraft positions to about 15% of the time for $\tau = 0$ since the positions are not updated for most of the time. This will become important for larger datasets (e.g., thousands of planes).

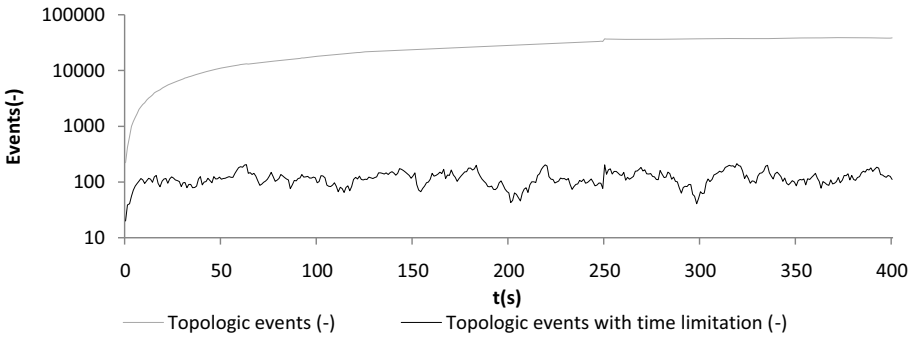


Fig. 6. Length of the priority queue with and without the time threshold

Due to the fact that the radar antenna is rotating, we may omit those events which will happen after one rotation because by then, we will have obtained new aircraft data and use them to compute new events. The graph in Fig. 6 shows us the difference in queue lengths with the thresholding on the event computation on and off – we may clearly see that the event time thresholding reduces the length of the priority queue to approximately 1% of its original size.

5 Conclusion and Future Work

In the given scale of the air traffic over the Czech Republic, our algorithm is stable and very fast and if used for larger datasets (more intense air traffic) we anticipate it to be usable even for situations where the number of the currently monitored aircrafts increases dramatically. Our method shows a possible way of future enhancement in the field of air traffic control as it provides a global point of view and is able to detect potentially dangerous situation automatically. In the future, we plan to artificially alter the real datasets in such a way that they will include aircraft collisions and perform various types of advanced testing.

Acknowledgements

We would like to thank to Mr. Karel Hanton and Mr. Petr Mach from the *CS Soft group* who provided us with the air traffic data, expert consultations on the topic and valuable references to various information sources.

References

1. Hjelle, O., Dæhlen, M.: *Triangulations and Applications*. Springer, Heidelberg (2006)
2. Gavrilova, M., Rokne, J., Gavrilo, D.: Dynamic collision detection in computational geometry. In: *12th European Workshop on Computational Geometry*, Munster, Germany, pp. 103–106 (1996)
3. Gold, C.M., Condal, A.R.: A spatial data structure integrating GIS and simulation in a marine environment. *Marine Geodesy* 18, 213–228 (1995)
4. Goralski, I.R., Gold, C.M.: Maintaining the spatial relationships of marine vessels using the kinetic Voronoi diagram. In: *ISVD 2007: Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering*, Washington, DC, USA, pp. 84–90. IEEE Computer Society, Los Alamitos (2007)
5. Hanton, K., Mach, P.: CS SOFT group. Personal Communication (2009)
6. Air Navigation Services of the Czech Republic, Aeronautical Information Service: Various documents, <http://lis.rlp.cz>
7. National Imagery and Mapping Agency, DoD: World geodetic system 1984, its definition and relationship with local geodetic systems, technical report 8350.2. Technical report (1997), <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>
8. Albers, G., Guibas, L.J., Mitchell, J.S.B., Roos, T.: Voronoi diagrams of moving points. *International Journal of Computational Geometry and Applications* 8(3), 365–380 (1998)
9. Vomáčka, T.: Delaunay triangulation of moving points in a plane, diploma thesis. Master's thesis, University of West Bohemia, Univerzitní 22, Pilsen, Czech Republic (2008)
10. Ralston, A.: *A First Course in Numerical Analysis*. McGraw-Hill, Inc., New York (1965)
11. Devillers, O.: On deletion in delaunay triangulations. In: *Symposium on Computational Geometry*, pp. 181–188 (1999)

GPU Calculated Camera Collisions Detection within a Dynamic Environment

Adam Wojciechowski and Grzegorz Wróblewski

Institute of Computer Science, Technical University of Łódź
ul. Wólczańska 215, Łódź, Poland

Abstract. Existing collision detection methods usually need long pre-calculation stage or difficult, time-consuming real-time computation. Moreover, their effectiveness considerably decreases with the growth of the complexity of the scene. Especially dynamic scenes with moving objects require the necessity of each frame collisions recalculation due to changeable objects' position and orientation. So far seemingly promising solutions supported by potential fields do not introduce satisfactory functionality as they are mainly devoted to static scenes with one predefined goal. This paper introduces a method which offers a new dynamic GPU supported potential field construction which lets the camera collide with both dynamic and static objects. Additionally, the proposed method does not need pre-calculation stage and provides an almost scene-complexity independent solution. The presented method is based on an `arbf1` shader solution¹, which means that most contemporary graphics cards can operate it without constraints

Keywords: collision detection, GPGPU, potential field, navigation, dynamic scene.

1 Introduction

Collision detection is a fundamental issue underlying objects' simulation process. It is important especially in 3D games or VR applications devoted to various phenomena simulation when movement of the camera should be bounded or behaviour of mutually moving objects is considered. The process of collision detection is theoretically very easy, however the cost of calculations increases steeply with the complexity of the scene. The phenomena can be complicated not only by complexity of an individual object but also by the amount of independently moving objects to consider as well. Nevertheless, the most common scenario in both games and VR applications is one or a few moving objects in a static environment.

Collisions are usually resolved in two steps [1]. The first step is based on a broad phase approach when algorithms cull away pairs of objects that can not possibly collide. This level usually makes use of various pre-calculated bounding volume hierarchies. The second step is called a narrow phase approach and

¹ A shader is a small program executed on GPU.

applies accurate collision detection by means of real-time face-face arithmetic intersection calculations.

As complexity of the objects may vary (a regular box vs. a very irregular tree), arithmetic calculations costs make programmers simplify considered objects. Many types of bounding boxes are used instead of original objects in calculations. Considerable reduction in the number of triangles is rather unavoidable in contemporary real-time applications. Moreover, users have already got accustomed to certain simplifications, i.e. nobody expects that a thin branch of a tree can be an obstacle on his way, whereas the tree trunk is always expected to stop the user.

Methods based on potential fields belong to the group of approximated collision detection algorithms and may considerably relieve the calculation process. They usually require a pre-calculation stage. However, a GPU calculations can be performed in real-time to an actual camera context. Potential fields assigned to the scene's objects become a source of potential field forces. These forces, sampled at an actual camera position, influence its consequent behaviour. Nevertheless, these methods may vary in accuracy, memory requirements and hardware dependencies. The method presented here is supposed to be faster, more precise and adjustable than others presented in literature.

2 Related Work

Construction of a potential field already met in literature was considered mainly in robotics [3,6,9]. In such environments there is usually one goal and a few static obstacles which should be passed by on the way to the predefined goal. Potential distribution is in conformity with a mathematical potential field theory. The target object is placed in the minimum of a local potential field and the obstacles are situated in local fields' maximums. Finally, algebraic superposition of the local fields is carried out. As a result, global potential field distribution is obtained and a robot can follow the way from its initial position to the global potential field minimum (goal). Potential field construction methods are used similarly in 3D environments where robot is replaced with a virtual camera. Nevertheless, potential field characteristic, prevailing currently in literature, has many serious drawbacks [1,2,7,8,15]. Local potential fields originating from obstacles form unpredictable extremums which are situated in unexpected places and they are blocking the camera on the way to the goal [4,5]. Forces coming out of potential field distribution do not reflect objects' shape, do not let the camera approach obstacles and moreover do not protect the camera from getting into the objects. Such potential field traps must be solved by additional correction algorithms [7,8,15].

Contrary to robotics' assumptions, most of the applications require unconstrained camera movement without one predefined goal. The camera should neither get into the objects nor be blocked in unexpected places. Collision detection should be precise enough to let the user reach each place in the scene not occupied by the objects. Such conditions can be fulfilled by forces coming out of newly

constructed potential field distribution both in static and dynamic environments. The most relevant to this paper is Beckhaus' CubicalPath method [12], however in the case of dynamic objects, the method suffers from time-consuming potential field recalculation process.

The presented paper is based on previous works on collisions with static and dynamic objects [13,14] performed with the CPU calculations. As a continuation, a GPU approach is presented with hardware generation and superposition of potential field distributions. Owing to hardware computations, no pre-calculation stage is needed. Additionally, efficient calculations accompanied with changeable view volume settings used for rendering with the technical camera can produce adequate potential field supporting horizontal and vertical movement of the user/camera.

3 Camera Navigation in a Dynamic Environment

By default camera is steered by input devices, i.e. mouse or keyboard. Input data from the keyboard can be translated to a vector \mathbf{K} parallel with the camera viewing direction (\mathbf{dir}). As a result, in each frame the position of the camera (\mathbf{pos}) is modified by vector \mathbf{K} (fig. 1a) according to equation 1

$$\mathbf{pos}' = \mathbf{pos} + \mathbf{K} \quad (1)$$

Simultaneously, a prevailing camera view direction can be modified by input data, for example coming from the movement of the mouse. Its movement generates vector \mathbf{L} (fig. 1b), which results in new camera view direction (\mathbf{dir}') according to equation 2

$$\mathbf{dir}' = \mathbf{dir} + \mathbf{L} \quad (2)$$

Camera guided by input data can freely navigate throughout the scene until it reaches potential field gradient in the static or dynamic objects' neighbourhood. In this area potential field forces would oppose camera tendencies to approach

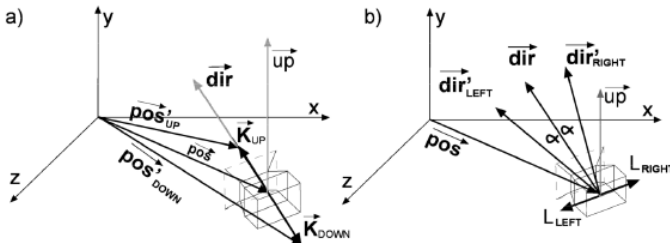


Fig. 1. Modification of the camera position and orientation by input devices. (a) position of the camera (\mathbf{pos}) modified by moving FRONT or BACK – i.e. by pressing keyboard UP or DOWN arrows; (b) direction of the camera (\mathbf{dir}) modified by turning LEFT or RIGHT – i.e. by moving mouse LEFT or RIGHT.

objects' contour. Final modification of the camera position (pos') influenced by all static and dynamic objects' forces can be calculated in accordance with equation [3](#)

$$\mathit{pos}' = \mathit{pos} + \alpha_1 \mathbf{K} + \alpha_2 \mathbf{F}_{\text{pot}} \quad (3)$$

where \mathbf{F}_{pot} is a superposed potential force generated for an actual camera position, according to the method presented in the next section, and α_i are scalar coefficients which balance influences of input devices and collision forces. With an adjustment of α_i values, strength of collision forces can be easily modified. Studies on α_1 and α_2 correlation are tested and presented in [13](#).

Movement of the virtual camera, which represents the user's point of view (the user's eyes), within a three dimensional environment can be constrained to predefined limits that reflect the user's casual behaviour. The height of the camera (the shortest distance between camera position and the ground – Y-axis coordinate) during the movement is usually fixed or constrained to certain levels: an ordinary height (h_c), a maximum height (h_{max}) and a minimum height (h_{min}), while X and Z coordinates can be freely altered with input devices and potential field forces. Assuming that the virtual camera is installed in user's head, these heights might represent, respectively: level achieved while walking mode (h_c), level achieved while jumping (h_{max}) and level achieved while squatting (h_{min}). Such assumption means that only objects whose parts are placed lower then actual camera height (h_c , h_{max} or h_{min}) can collide with the camera (virtual user). Limitations generated for h_c , h_{max} and h_{min} can differ and depend on objects' structure (fig. [2](#)).

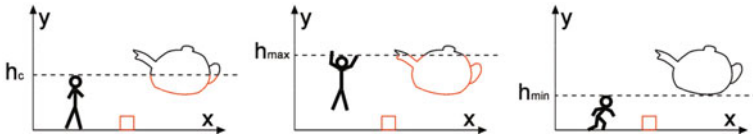


Fig. 2. Red elements mark parts of the objects that can collide with the camera/user at its selected heights: h_c – walking, h_{max} – jumping, h_{min} – squatting

Additionally from the camera/user point of view not all of the clipped objects but just a close camera neighbourhood should be considered in each frame for collisions detection. GPU approach based on a scene top-down technical orthogonal projection, presented in the paper, supports capturing desired viewer context. A dedicated orthogonal view frustum for the technical camera is placed above the user and looking down the floor (negative Y direction), with the near plane at the viewer height and the far plane at the floor level, can provide a texture reflecting viewer vicinity (fig. [3](#)).

Moreover, adequate clipping planes modification easily adjusts the method for different camera altitudes. Placing the near clipping plane at h_c , h_{min} or h_{max} can satisfy, respectively, the walking, squatting and jumping modes from the figure [2](#).

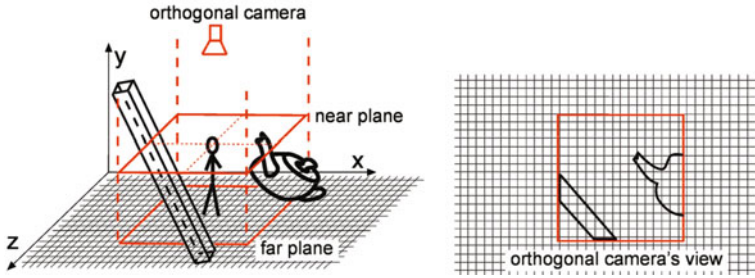


Fig. 3. Technical orthogonal view frustum determining vicinity of the viewer with parts of the objects considered for possible collisions. (a) isometric view of a GPU used view frustum; (b) orthogonal projection of possibly colliding objects within the viewer vicinity.

4 Potential Field Construction

For further calculations, we consider one temporary camera height (h_{temp}) representing viewer vertical position. For that camera height, an orthogonal view frustum is defined with near clipping plane at h_{temp} (fig. 3). 3D surrounding objects which are to be taken into account are drawn to create a monochrome map of camera vicinity (fig. 4). Technically lighting and texturing should be disabled to ensure that a monochrome image is obtained and culling must also be disabled to draw all objects, even if they are partly clipped (which can happen for objects that descend into the explorer's height range). With texture set as a rendering target two images are generated, one with red channel set to 1 for background (fig. 4b) and second with green channel set to 1 for objects (fig. 4c). Blue and alpha channels are ignored. Resolution of a texture should depend on the context of the camera under consideration, and performance of a graphics card. In this paper, the texture resolution used was 128x128 pixels and the view frustum length was 2 units (in world coordinates), resulting in a grid having 64 nodes for every unit in the world coordinates.

On the basis of the textures, reflecting viewer's potentially colliding objects (fig. 4), consecutive potential field distribution must be generated. Dedicated

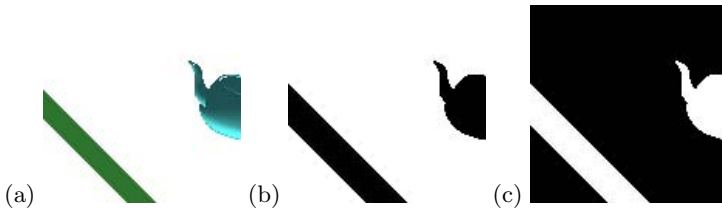


Fig. 4. Initial rendering of 3D objects in the camera vicinity: (a) part of the scene under consideration, and a monochrome maps of the (b) red (where background = 1) and (c) green (where objects = 1) channels

potential field is defined according to equation 4. CPU implementation conditions potential value $U(\mathbf{q})$ on the Euclidean distance $d(\mathbf{q})$ from the considered point \mathbf{q} to the closest obstacles' contour (object),

$$U(\mathbf{q}) = \begin{cases} \max(-1, \frac{-d(\mathbf{q})}{R}) & \text{inside the contour} \\ \min(1, \frac{d(\mathbf{q})}{R}) & \text{outside the contour} \end{cases} \quad (4)$$

where R is a radius of the potential field under consideration.

Since GPU operations are based on textures and the programming language `if` statement is not present [10], a dedicated raster based method evaluating distance from the contour is proposed. For each pixel in the projection image (separately red and green channel), the shader evaluates its square fixed-size environment to find the distance to the nearest pixel with the value 0. The square has sides of the size $2L + 1$ pixels, where L defines the *radius* of the square environment under consideration (in the paper $L = 4$). First, the result variable (`res`) is initialized with the environment radius L . For each pixel in the environment, the sum of its distance (`length(delta)` in eq. 5) from the center and pixel value (1 or 0) multiplied by a high penalty factor (not smaller than the radius) is calculated and compared to the result variable (eq. 6). If the newly-calculated value is smaller, then it is stored in the result variable. The loop iterates over `dx` and `dy` from $-L$ to L , with its body implemented as follows:

```
delta = float2(dx, dy); // (5)
```

```
res = min(res, length(delta) + (16 * texRECT(arr, coords + delta))); // (6)
```

where `dx` and `dy` are the variables used to iterate over the square environment, `res` is the result variable, `arr` and `coords` are the inputs of the shader (texture sampler and texture coordinates, respectively), and the constant value 16 represents the above-mentioned penalty factor. If the value of the pixel is zero, the sum is equal to the distance, and it can pass the test. If the value is one, the result is larger than the penalty constant and therefore is discarded. In this way, no conditional statements are needed to test if the value of the pixel is zero or one. After processing the whole environment, the result candidate is multiplied by the value of the processed pixel, so that only pixels with a value of 1 are taken into account; pixels with a value of zero get zero as the result. As a result, in both red (outer) and green (inner) channels the appropriate gradients of distances are obtained (fig. 5).

Because of the way the initial monochrome images are produced, no pixel can have the value 1 in both red and green channel, and the scaled potential field gradient superposition may easily be performed (eq. 7,8). The texture green channel value (inner of the object) is multiplied by -1 and summed with red channel (outer of the object) yielding the desired output (fig. 6).

```
float4 here = texRECT(arr, coords); // (7)
```

```
res.r = res.r * here.r - res.g * here.g; // (8)
```

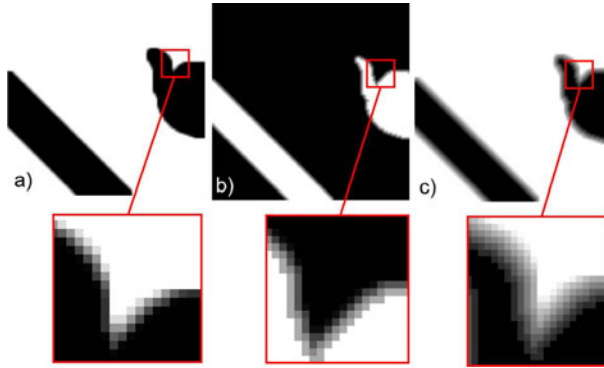



Fig. 5. Distribution of a potential field within 128x128 pixels neighborhood (a) for the red channel (white = 1, black = 0) (b) for the green channel (white = 1, black = 0) (c) superpositioned potential field gradient (white = 1, black = -1)

where **here** is the value of the texture sampled at the coordinates passed as an argument to the shader, and **res** is the result variable. Finally, the potential value is scaled to the appropriate range $[-1; 1]$ and stored in the red channel (fig. 5c). Other channels are ignored.

During the method tests it was noticed that in certain situations (fig. 6a) an aliasing effect may result in a jugged movement. In such situations an optional blurring step can be used (fig. 6b). A map can be blurred n times with n being one of the parameters of the algorithm. At each step blur filter (fig. 6c) is applied to the pixel and its environment.

The third step of the method is to calculate forces on the basis of potential gradient according to equation 9

$$\begin{aligned}
 \mathbf{F}(\mathbf{q}) &= [F_x(\mathbf{q}), F_z(\mathbf{q})] & (9) \\
 F_x(\mathbf{q}) &= U(\mathbf{q}_{-1,1}) - U(\mathbf{q}_{-1,-1}) + U(\mathbf{q}_{0,1}) - U(\mathbf{q}_{0,-1}) + U(\mathbf{q}_{1,1}) - U(\mathbf{q}_{1,-1}) \\
 F_z(\mathbf{q}) &= U(\mathbf{q}_{1,-1}) - U(\mathbf{q}_{-1,-1}) + U(\mathbf{q}_{1,0}) - U(\mathbf{q}_{-1,0}) + U(\mathbf{q}_{1,1}) - U(\mathbf{q}_{-1,1})
 \end{aligned}$$

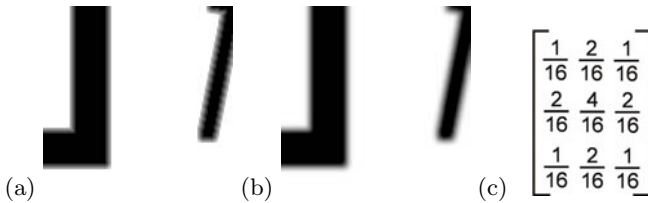


Fig. 6. Blurring the texture to solve an aliasing effect. (a) texture with an aliased edge, moving along which causes jugged movement; (b) after blurring the texture, the unevenness is eliminated and the movement is smoother; (c) blurring filter.

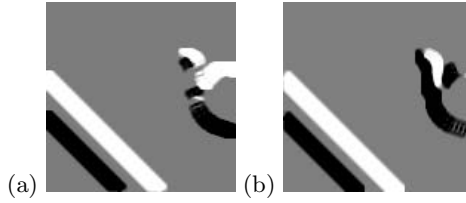


Fig. 7. Distribution of potential field forces in camera neighbourhood. (a) component X saved in a red texture channel (white color represents force that acts along the positive X axis, black color force acting along the negative X axis, and gray represents neutral situation where no force acts); (b) component Z saved in the green texture channel.

Forces are represented as two dimensional vectors (X and Z axis in OpenGL coordinating system), because only horizontal movement is applied for the camera. Forces calculated according to equation 9 are stored in a new texture. As each force has two components, the data is written into two channels of a texture – red and green. Resulting red and green channels are presented in fig. 7.

The resulting texture of forces’ components is then transferred from the texture into the main memory, and further used by the CPU to update the camera’s position. First of all, the potential force vector (F_{pot} from eq. 3) acting on the camera/viewer in a temporary position (fig. 8) is calculated according to equation 10.

$$F_{pot}(p) = \frac{\sum_{i,j=-1}^2 F(q_{i,j}) \cdot e^{d(p,q_{i,j})}}{\sum_{i,j=-1}^2 e^{d(p,q_{i,j})}} \tag{10}$$

Then the position of the camera is updated with the input data (user direct activity) and an actual potential filed force F_{pot} from equation 3. This procedure is repeated several times, until no force acts on the camera, or the limit of iterations is reached.

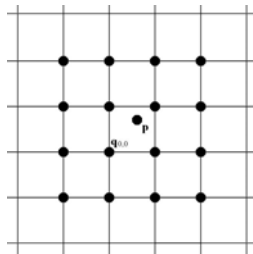


Fig. 8. Position of the camera p is influenced by mean force calculated by means of equation 10 on the basis of the forces in the set of 16 neighbouring nodes ($q_{i,j}$) from eq. 9.

5 Tests

In order to prove the effectiveness of the method, a virtual maze with various objects was created (fig. 9). In the maze, there was an object moving along a path (a box), object that was rotating (a turnstile), object flying up and down in the air (a flying teapot) and an object that was changing shape in a random manner (called the “bubble”). Also, some objects were placed partially above the viewer’s height (h_{temp}). The objects were constructed out of primitive bodies (spheres, cubes, etc.) and the “bubble” was constructed out of six rotated, translated and scaled spheres.

The task of the test was to find a way through the maze. The way the objects were placed in the maze made it impossible to find a way through without colliding with them. All the collisions were correctly solved by means of introduced potential field method. Finally, all the people that took part in the test managed to find a way through the maze, correctly interacting with all obstacles (all collisions were solved properly).

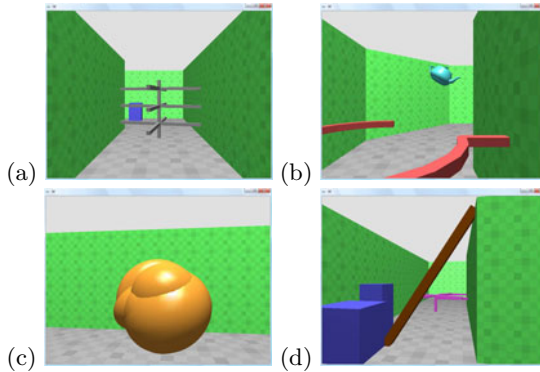


Fig. 9. Sample screenshots with the scene used during the tests; (a) a rotating turnstile, (b) a flying teapot, (c) the “bubble”, (d) a pillar and another turnstile

6 Conclusions

So far, the potential field theory was mainly utilized in robotics. There were also a few attempts to make use of its advantages in a three dimensional scene navigation. Unfortunately, pure mathematical potential field calculation approach has led to many situations in which the user was trapped or experienced other problems during movement [12][78]. However, in opposition to existing methods, the worked out solution has turned out to be resistant to delays in the case of dynamic objects. The presented method does not need long pre-calculation stage, as most of the time consuming calculations are performed on a graphics card. GPU multiple objects rendering is much more effective than previous CPU potential field computations [13][14]. This article has shown that adequate potential field distributions and potential field superposition algorithms can be a

very useful tool. In this paper, potential fields were used for collision detection process, which was perceived as a time consuming and calculation costly task. The proposed method has turned out to be not only easy to use but also easily adjustable to the scene complexity as well. The effectiveness of the method was proved in tests.

References

1. Beckhaus, S., Ritter, F., Strothotte, T.: CubicalPath - Dynamic Potential Fields for Guided Exploration in Virtual Environments. In: PG 2000. IEEE, Los Alamitos (2000)
2. Beckhaus, S.: Dynamic Potential Fields for Guided Exploration in Virtual Environments. PhD thesis, Otto-von-Gueicke-Universitat Magdeburg (2002)
3. Dułęba, I.: Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych. Akademicka Oficyna Wydawnicza Exit (2004)
4. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Mobile Research* 5(1), 90–99 (1986)
5. Khatib, M., Chatila, R.: An Extended Potential Field Approach for Mobile Robot Sensor-based Motions. In: *Intl. Conf. on Intelligent Autonomous Systems, IAS'4* (1995)
6. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Dordrecht (1991)
7. Li, T.Y., Chou, H.C.: Improving Navigation Efficiency with Artificial Force Fields. In: *Proceedings of 14th IPPR Conference on CVGIP, Taiwan* (2001)
8. Li, T.Y., Hsu, S.W.: An Intelligent 3D User Interface Adapting to User Control Behaviours. In: *Proc. of the 9th Int. Conf. on Int. UI, Madeira, Portugal*, pp. 184–190 (2004)
9. Murphy, R.R.: Introduction to AI Robotics. MIT Press, Cambridge (2000)
10. Szajerman, D., Pietruszka, M.: Real-time ice visualization on the GPU. *Journal of Applied Computer Science* 16(2), 89–106 (2008)
11. Watt, A., Policarpo, F.: 3D Games. Real-time Rendering and Software Technology. Addison-Wesley, Reading (2002)
12. Wojciechowski, A.: Wspomaganie dynamiki procesu nawigacji w eksploracyjnym środowisku wirtualnym. PhD Thesis, Technical University of Lodz (2005)
13. Wojciechowski, A.: Potential field based camera collisions detection in a static 3D environment. *Machine Graphics and Vision* 15(3/4), 665–672 (2006)
14. Wojciechowski, A.: Potential field based camera collisions detection within dynamically moving 3D objects. LNCS. Springer, Heidelberg (2009)
15. Xiao, D., Hubbard, R.: Navigation Guided by Artificial Force Fields. In: *Proceedings of CHI 1998, Los Angeles, USA* (1998)

Utilization of Multi-spectral Images in Photodynamic Diagnosis*

Andrzej Zacher

The Silesian University of Technology, Institute of Informatics
Akademicka 2, 44-100 Gliwice, Poland
Andrzej.Zacher@polsl.pl

Abstract. This paper introduces multi-spectral images for healthy and cancerous parts of human skin. It compares light spectrum calculated from those images with spectrum obtained from simulation. First the mathematical model of tissue and Monte Carlo algorithm of light propagation in turbid media is presented. This theory was then extended to imitate the fluorescence phenomenon, necessary for cancer recognition. Then the processing method of non-normalized multi-spectral images was described. Finally both results were compared to confirm that the assumed model is correct. Having all those information it will be possible to simulate such environment, which applied into reality, would make the cancer diagnosis much faster.

1 Introduction

Basal Cell Cancer (BCC) is one of the most popular skin cancer. The sick rate seems to show the relationship with the level of pigmentation. This is why Ultra Violet Radiation (UVR) is the most common reason for BCC in all people races. Also scars, ulcers, immunosuppression and genetic disorders are possible risk factors for BCC occurrence. Most patients are elderly with similar clinical symptoms. Single, translucent nodules with ulceration in the center are in many cases difficult to diagnose, especially when BCC occurs in color of the skin. Investigations showed that 89% of all cases were aligned on the head and neck regions. Mean survival rate for metastatic BCC is between 8 months and 3.6 years [1].

Photodynamic diagnosis is mainly a method of treatment of superficial tumors, by illuminating cancerous part of human tissue in combination with an application of photosensitizing drug. Most of skin diseases are characterized by significant photosensitivity, which is related to haem synthesis, enzyme disorders and drug accumulation. One of the most active agent is protoporphyrin IX, which were applied to all patients during investigations [2]. Different reflectance of healthy and BCC tissue gives the information about tumor stage.

* Supported by Ministry of Science and Higher Education grant R13 046 02.

2 Previous Work

The application of diffusion theory to describe steady-state light transport for point, line and plane sources and compared with Monte Carlo results was presented in [3]. They were able to accurately describe the shape of the fluence rate distributions due to light sources, but the absolute values were difficult to calculate. The adaptation of the method to fluorescence measurements and photodynamic therapy were presented as well.

In another work an eight-layer model of human skin was used. Optical properties of tissue were considered as a function of wavelength (absorption and scattering). Additionally microsphere fluorescent materials were implanted under the tissue surface. Their excitation and emission spectrum were represented by six discrete data points. The simulations were performed to predict the distribution of excitation and fluorescent photons. Photons escaping the tissue were scored in a positional matrix together with their respective wavelengths and weights. It turned out that about 7% of input light is diffusely reflected. Moreover, about 20% of the input light was converted to fluorescence that escaped the tissue surface. Simulations proved that the quantitative analysis of emitted photons is possible [4].

The study about skin reconstruction was presented in [5]. They froze skin tissue samples and cut them into pieces of about $10\mu\text{m}$ thick. Using microscope to excite the emission of autofluorescence and Monte Carlo simulation, they were able to reconstruct in vivo skin autofluorescence spectrum. Seven-layer model with different optical parameters - like refractive index, scattering, absorption and regeneration of fluorescence photons - for each layer and each wavelength was proposed. The highest fluorescence detection efficiency in the skin was observed for dermis. Besides the blood absorption bands, comparison of the measured spectra with the reconstructed in vivo spectrum showed good agreement.

Morphological and biochemical changes discovered in cancerous tissue, disturb its absorption, scattering and fluorescence properties. In vivo optical spectroscopy can examine changes associated with pre-cancer phase. A recent study described in [6] concentrates on two key areas: characterizing nuclear morphology using light scattering spectroscopy and investigating changes between epithelial cells and stroma at early stages of carcinogenesis. Both approaches are needed as long as fluorescence and reflectance spectra contain complementary information.

Time-resolved fluorescence was investigated in experimental and theoretical ways in [7]. They used a normal muscle tissue and a sarcoma tumor with porphyrin photosensitizer in the middle. The fluorescence emission was studied by comparing fluorescence decay of photosensitizer in different stages of cancer growth. It was noticed in the in vivo experiment that the fluorescence from the normal tissue decayed more quickly than from the tumor structure. Decay is also dependent on the tumor thickness - the thicker cancer, the more delayed decay. Those information can be useful for tumor diagnosis.

Problem of energy transport in scattering medium was also solved in [8] using Feynman approach. It utilizes path integrals method to obtain the most probable way the light propagate in medium and provides results without making

diffusion approximation. Scattering events determined mainly by phase function make it possible to calculate probability distribution function for a photon to follow a given path between two point within given time. Data generated by Monte Carlo simulation and compared to this analytical solution gives perfect agreement. Path integral approach could be probably used to solve the inverse problem, where having optical measurements of a tissue internal structure could be recognized.

3 Monte Carlo Model

Photon propagation in turbid medium like tissue, can be easily simulated by Monte Carlo algorithm. It utilizes the idea of tracing photons in a medium, where absorption and scattering are dominant events. The movement is governed by a set of rules, which relies on random sampling of propagation variables, which effects in different paths and distribution of particles.

This method can be briefly reviewed in the following way:

1. Initialize photon with a weight of unity (W).
2. Shoot photon from the light source
3. If the photon was reflected by tissue then store its weight, exit angle and position in the special data structure called photon map. Otherwise the simulation continues inside tissue.
4. Calculate the distance Δs to the next interaction event

$$\Delta s = \frac{-\ln(\zeta)}{\mu_a + \mu_s}, \quad (1)$$

where:

ζ - random number,

μ_a and μ_s - absorption and scattering coefficients respectively.

5. Move photon by Δs
6. If photon is inside the medium then its weight needs to be decremented due to absorption according to:

$$W = W \frac{\mu_s}{\mu_a + \mu_s}, \quad (2)$$

where:

μ_a and μ_s - absorption and scattering coefficients respectively.

If the resulting weight is smaller than a given threshold then there is no need to continue the photon propagation, as long as it does not contain any useful information. The energy conservation principle must be preserved, so the photon cannot be simply terminated. A technique called Russian roulette provides the solution. It gives one chance in m (e.g. $m = 10$) for a photon to survive with weight mW . Otherwise its energy is set to zero and photon is terminated.

After the weight attenuation, the photon is ready to be scattered. The deflection angle is defined by the angle θ in the interval $[0, \pi]$. Basing on probability density function proposed by the Henyey and Greenstein the following equation holds, that helps to calculate the deflection angle:

$$\cos(\theta) = \begin{cases} \frac{1}{2g} [1 + g^2 - (\frac{1-g^2}{1-g+2g\zeta})^2] & \text{for } g \neq 0 \\ 2\zeta - 1 & \text{for } g = 0 \end{cases} \quad (3)$$

where:

ζ - random number,

g - anisotropy factor characterizing angular distribution of scattering.

Azimuthal angle ψ , however, is defined in interval $[0, 2\pi]$. This time the probability distribution function is constant, so the angle is calculated by simple multiplication:

$$\psi = 2\pi\zeta \quad (4)$$

In this way new direction of propagation was calculated and the photon can be moved again by another Δs . This process goes on until the photon is terminated.

7. If photon is outside the medium then check if it was internally reflected. If this is true, then update its position accordingly and continue the algorithm. Otherwise, if the photon escaped, then store its weight, exit angle and position in photon map.
8. This process continues until desired number of photon is collected [9].

In order to apply this method in computer graphics each photon has a spectrum, that describes its energy or its weight. If the absorption event occurs the spectrum is decreased. Additionally, in order to simulate fluorescence phenomenon the probability d_{fluor} was provided. In such a case the spectrum is not only reduced, but its shape get changed according to EEM i.e excitation-emission matrix. The optical characteristic of protoporphyrin was available, but it should be kept in mind that it can be slightly different than for endo-protoporphyrin IX, which was applied to all patients.

For rendering purpose, the camera is placed at the same position as the light source. Moreover, it was assumed that the tissue surface is strictly diffuse.

Finally, the number of photons that were collected during simulation was equal to 100000. μ_a and μ_s were set to 20cm^{-1} and 200cm^{-1} respectively, anisotropic coefficient g equals to 0.9 and d_{fluor} to 0.2. For such a settings the simulation was started.

4 Processing of Multi-spectral Images

Thanks to cooperation with Medical University of Silesia and the Center of Diagnostics and Laser Therapy of Tumors in Bytom, it was possible to gather multi-spectral images from patients with skin cancer (see Table I). Images for the second patient look very similar.

Table 1. Multi-spectral images of cheek skin for a white light - patient 1

λ	Bcc	Healthy	λ	Bcc	Healthy
400nm			576nm		
416nm			592nm		
432nm			608nm		
448nm			624nm		
464nm			640nm		
480nm			656nm		
496nm			672nm		
512nm			688nm		
528nm			704nm		
544nm			720nm		
560nm					

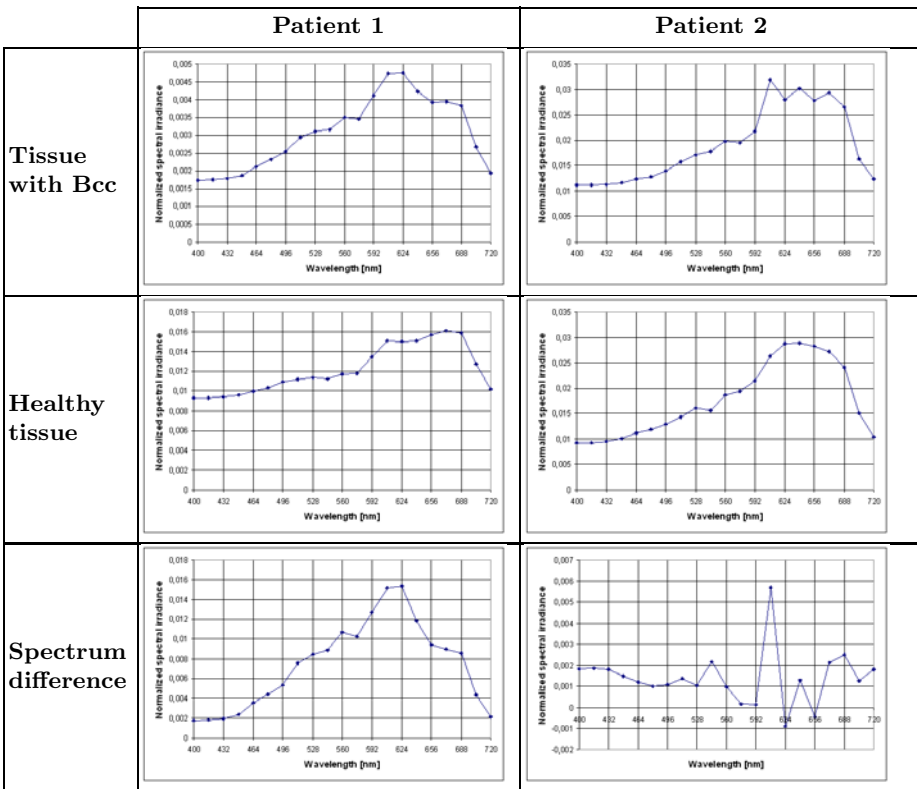
In each case 21 grey scale photos were taken for wavelengths between 400nm and 720nm with step equal to 16nm. The luminance value can be treated as an intensity. All images were non-normalized, which means that they were provided as they are and without any changes. It guarantees that any important information isn't lost during some preprocessing operations. Raw data is useful to analyze the content of pictures, but because of lack of normalization, they cannot be viewed in popular image viewers.

Tone reproduction could be utilized as a tradeoff between real images and limited capabilities of display devices. To localize cancerous changes in human tissues, maximum to white operator was applied. It looks for the pixel with the greatest luminance and converts the rest of them linearly under the assumption that the pixel with highest luminance is mapped to maximum value of a display - in this case white color [10].

The reflectance spectrum can be created by choosing a point on multi-spectral images and calculating its intensity.

The comparison of obtained graphs for two different patients with Basal Cell Cancer is presented in Table 2. It is remarkable for both patients that for BCC

Table 2. Light spectrum obtained from multi-spectral images



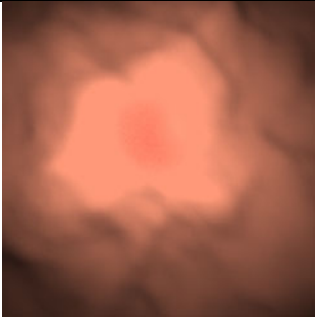
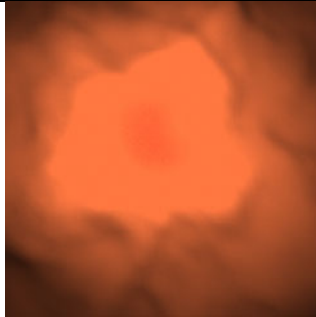
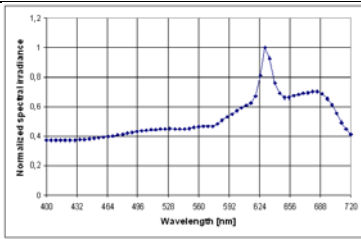
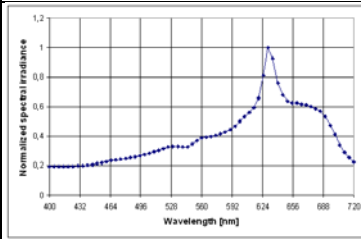
spectrum a high peak at 608nm is visible. This way diseased and healthy tissue can be distinguished. After subtracting both spectras from each other the graph showing only differences occurs. It worth to mention that for patient 1 changes are closely related to morphological cancer deformations in tissue and with spectrum of endo-protoporphyrin IX. Data obtained after spectrum subtraction for patient 2 are not of good quality. However they again show that thanks to fluorescent properties of tissue, cancer can be recognized by excessive intensity around 608nm.

5 Human Skin Tissue Rendering

Now its time to use everything what was described so far. Monte Carlo simulation utilizing photons penetrating human skin combined with healthy tissue spectrum is going to display the predicted appearance of analyzed area. The process of rendering was performed basing on reflectance spectrum of normal skin. Tissue surface was modeled as a flat semi-infinite plane with random distortions making it more realistic. In the middle of the tissue surface a small structure containing highly fluorescent substance - endo-protoporphyrin IX - was injected to simulate the reflectance spectrum of a cancer. Obtained results are presented on Table 3.

Depending on how much light is reflected from the tissue, the area of highest light intensity has different size. This effect causes that the details of a surface are missing and only illumination is visible. The color of the skin for both patients differs as well. As long as spectrum is calculated from multi-spectral images it is

Table 3. Image rendering of tissue with BCC for 2 patients

	Patient 1	Patient 2
Image		
Simulated spectrum		

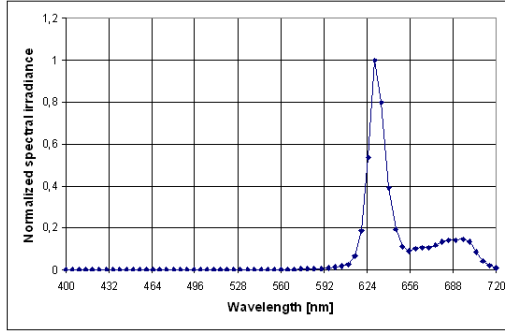


Fig. 1. The spectrum of endo-protoporphyrin IX obtained for white light

not ideal. For patient 1 the color of a tissue is similar to that of a skin, but for the second one it is closer to orange color. Actually it is not a problem at all, because the peak effect in the reflectance spectrum is still observable.

The results of simulation can be compared with multi-spectral images obtained for real tissues (see Table 2). They are in good agreement with each other and the influence of fluorophores are meaningful for final image appearance. In both cases the spectral highest value can be observed somewhere around wavelength equal to 610nm. Other differences between original and rendered images are coming from the morphological changes caused by a tumor and need further investigations.

Additionally the spectrum of the only fluorophore was extracted from the image after illuminating with white light source (see Fig. 1). It has an important property that the spectrum value for wavelengths less than about 590nm is equal to zero. Then it goes to its maximum and after that it remains at almost constant level around red color. Its influence is clearly observable on the simulated cancer spectrum in Table 3. Having this shape well defined it is important now to find other absorbers that influence the human skin reflectance spectrum. The complete model would improve the analysis, but even now it is very useful and shows the potential of multi-spectral images.

6 Conclusions

This paper had many goals. First, it described shortly photon mapping algorithm applied to turbid medium like human tissue. Then the idea of multi-spectral images were presented and the way for obtaining a reflectance spectrum at a given point. Finally, the technique of rendering images as a powerful tool to predict tissue appearance under light illumination was introduced. This information would be helpful to find such a light properties that are more efficient in cancer recognition process. It was also verified that simulated images are in relationship with real tissue optical properties.

Missing the accurate excitation-emission matrix for endo-protoporphyrin IX and using EEM of simple protoporphyrin instead, can be one of the sources for

many errors. Moreover, it was assumed that absorbing and scattering properties of a tissue are the same for all possible wavelengths. This is not true in a real world, however this phenomenon was not implemented in the human skin model. Additionally changes in morphological structure for healthy and diseased tissue were not taken into account to keep the simulation as simple as possible. Despite of all those drawbacks, the described model of a skin and Monte Carlo approach to imitate tissue properties under light illumination is valid and can be compared with reality.

References

1. Bradford, P.T.: Skin Cancer in Skin of Color. *Dermatology nursing* 21(4), 170–206 (2009)
2. Yavari, N.: Optical spectroscopy for tissue diagnostics and treatment control. Doctoral Thesis, Department of Physics and Technology University of Bergen (2006)
3. Jacques, S.L.: Light Distributions from Point, Line and Plane Sources for Photochemical Reactions and Fluorescence in Turbid Biological Tissues. *Photochemistry and Photobiology* 67(1), 23–32 (1998)
4. Long, R., McShane, M.J.: Modeling of Selective Photon Capture for Collection of Fluorescence Emitted from Dermally-Implanted Microparticle Sensors. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 21(4), pp. 2972–2975 (2007)
5. Zeng, H., MacAulay, C., McLean, D.I., Palcic, B.: Reconstruction of in vivo skin autofluorescence spectrum from microscopic properties by Monte Carlo simulation. *Journal of Photochemistry and Photobiology B* 38(4), 234–240 (1997)
6. Sokolov, K., Follen, M., Richards-Kortum, R.: Optical spectroscopy for detection of neoplasia. *Current Opinion in Chemical Biology* 6(5), 651–658 (2002)
7. Li, Y., Li, M., Xu, T.: Quantitative Time-Resolved Fluorescence Spectrum of the Cortical Sarcoma and the Adjacent Normal Tissue. *Journal of fluorescence* 17(6), 643–648 (2007)
8. Perelman, L.T., Wu, J., Itzkan, I., Feld, M.S.: Photon Migration in Turbid Media Using Path Integrals. *Physical Review Letters* 72(9), 1341–1344 (1994)
9. Wang, L., Jacques, S.L.: Monte Carlo Modeling of Light Transport in Multi-layered Tissues in Standard C. University of Texas M. D. Anderson Cancer Center (1992)
10. Pharr, M., Humphreys, G.: Physically based rendering. From theory to implementation. Morgan Kaufmann, San Francisco (2004)

The Spectral Analysis of Human Skin Tissue Using Multi-spectral Images*

Andrzej Zacher

The Silesian University of Technology, Institute of Informatics
Akademicka 2, 44-100 Gliwice, Poland
Andrzej.Zacher@polsl.pl

Abstract. This paper analyses the properties of human skin, when the light source from different angles and positions was applied. It answers the question how the tissue appearance varies and how the changes of radiance distribution can be observed on multi-spectral images. As long as they contain raw data, that wasn't changed by any postprocessing algorithm, they directly represent light spectrum and its intensity. All images were taken at wavelengths ranging from 400nm to 720nm with step 16nm long. As the next experiment the reflectance spectrum of healthy and diseased tissue was compared. Due to fluorophore concentration and utilizing multi-spectra images, it was possible to unambiguously determine cancerous part of human skin tissue.

1 Introduction

Images taken across the electromagnetic spectrum at different wavelengths are called multi-spectral. Special filters or instruments sensitive to the required wavelength, including ultra-violet or infrared light, are able to register only those data, that are necessary for the desired image component. They are represented by using gray-scale images. Multi-spectral images help to detect additional surface features and to capture information that are almost invisible for human eye. By combining the contribution of each image, it is possible to obtain the discrete color spectrum of each pixel.

This technique can be applied to identify cancerogeneous changes in the human skin. The physical deformation influences the structure of the tissue, so that it reflects light in different way. The optics of human skin was widely described in [1]. The tissue was divided into layers with different inherent optical properties, caused by various concentration of blood, melanin and keratin. Skin without those substances appears diffusely white. The reason for such a color results from the fact, that many skin components have dimensions larger than the wavelengths of visible light.

All substances have different optical properties. Melanins and keratins in the epidermis, and elastin and collagen fibers in the dermis were identified as the primary scatterers in skin. Especially melanins since they can be characterized

* Supported by Ministry of Science and Higher Education grant R13 046 02.

with higher refractive index. Moreover, scattering efficiency does not change rapidly with wavelength. To the group of main absorbers was included blood, melanosomes and keratin. For the wavelength greater than 600nm, absorption of blood is very low resulting in red color. The concentration of the melanosome influences the skin reflectance at all wavelengths. Keratin as a main absorber builds nails and hairs, and can be found in epidermis and in stratum corneum.

Different concentration and localization of absorbers and scatterers in human tissue results in various spectral reflectance. The color depends on the skin illumination i.e. the position and incident angle of the light rays. Changes in tissue structure caused by tumor activities, can be recognized thanks to fluorescent properties of substances the human skin is made of. Multi-spectral images help to understand how the investigated area looks like in different wavelength, which one is the best to study deeper and which one gives the information that could be useful in case of photodynamic diagnosis. Each image contains data, that wouldn't be visible with normal camera.

2 Previous Work

An excited electron returning to its ground state after light absorption, emits light of different wavelength. This physical phenomenon that happens in many substances including biological tissues is called fluorescence. Human skin containing such fluorophores as NADH, tryptophan (aromatic amino acid), collagen and elastin (structural proteins) is sensitive to the environment and the metabolic status of the tissue. This optical behavior makes the fluorescence spectroscopy a valuable tool to study the condition of biological tissues. In [2] fluorescence lifetime measurements were presented. Time-correlated single photon counting method was applied, when human skin was exposed to light pulse of wavelength 375nm. Recorded fluorescence decays were compared between males and females and among people in age ranging from 10 to 70 years. Investigations were performed on different locations i.e. the arm, the palm and the cheek. Obtained results perfectly fit to three-exponential fluorescence decay model. This paper is of particular interest with the possibility to monitor diseases and skin conditions such as skin cancer and ulcers.

The reflectance spectra of the human skin have been calculated using Monte Carlo method in [3]. The model of skin tissue was described as semi-infinite three dimensional medium divided into seven layers. They concentrated on calculating the absorption coefficients of oxy- and deoxy-hemoglobin, water and other parts of tissue. All those factors contributed to compute total diffuse reflectance for different wavelengths. Finally, they compared the simulation outcome with in vivo experimental results and showed they are in good agreement with each other. Moreover, they remark that basing on reflectance spectra, the model fitting analysis using multi-linear regression, could give quantitative amount of chromophores in human skin.

Some ways of rendering fluorescent phenomenon in computer graphics were described in [4]. They paid attention to the fact that the reflected light from

the fluorescent material depends on the angle of the incident light. The greater the angle, the more visible the effect of frequency and color shifting. The phenomenon occurs only if a photon interacts with colorant molecules. This implies a penetration of the diffuse substrate resulting in radiant energy and spectrum changes. Finally, they provides a BRDF model based on the idea of a layered microfacet technique. They also compared a combination of Phong lobes and Lambert model with layered varnish model, which gave the best results.

Extracting the intrinsic fluorescence properties of tissue is the main goal of [5]. Monte-Carlo-based model was developed to obtain fluorescence spectra from combined fluorescence and diffuse reflectance spectra i.e. independently from absorption and scattering properties of investigated medium. The proposed model was verified using 11 phantom configurations containing blood as an absorber, polystyrene as a scatterer and furan-2 as a fluorophore. They were able to retrieve the intrinsic fluorescence spectra and intrinsic fluorophore intensity with an error not greater than 10%. Their approach could be used in a number of biomedical applications, especially cancer diagnosis and observing the physiological tissue response - both useful in therapy.

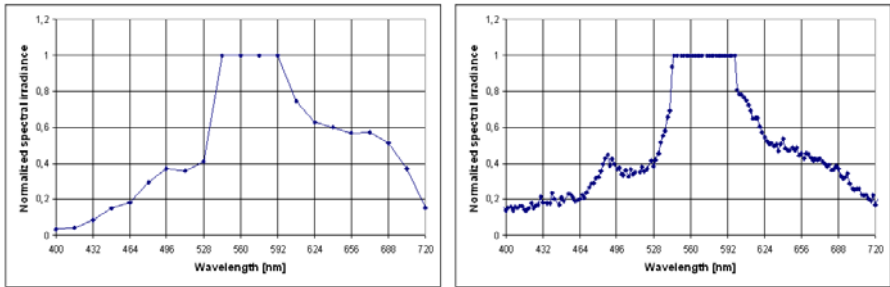
Very precise optical model of port wine stain was presented in [6]. The data was extracted from $6\mu\text{m}$ thick histology sections, which then were photographed and digitized with $2\mu\text{m}$ per pixel resolution. The epidermis, dermis and blood were manually identified to create a three-dimensional matrix with values representing tissue type. Having described internal structure of a skin deformation, they were able to perform Monte Carlo simulation and collect the energy deposition in different parts of a tissue grid. The simulation executed for wavelengths 532nm and 585nm showed that the greatest energy was accumulated in superficial blood vessels. Moreover, more energy was deposited in dermis and blood for 532nm rather than for 585nm. They also noted that big complexity of the model make it difficult to interpret the results.

Many biological cells and large subcellular structures like chloroplasts or mitochondria in suspension change their volume, but without changing their dry weight. This transformation is related to the rate of cellular processes, which results in various scattering angles of incident light. [7] applied Mie theory to calculate light intensity for different scattering angles. For 10° the model followed the value of total scattering, but for 45° , 90° and 135° distortions were observable. It was showed that the light scattered at different angles contains valuable information about the volume of the sample particle.

3 Spectral Analysis of Human Skin Tissue

The aim of this experiment was to analyze the light propagation in the human skin and its reflectance. The internal structure of a tissue and its fluorescent properties may change the outgoing spectrum. The final appearance of human skin strongly depends on the light position and its color. In Medical University of Silesia and the Center of Diagnostics and Laser Therapy of Tumors in Bytom, three separate measurements were performed. The data was collected for healthy wrist tissue illuminated by white light source.

It was possible to obtain multi-spectral images for 21 different wavelengths, ranging from 400nm to 720nm with 16nm step. Images were not normalized, it means that the intensity of a color directly represents a spectral radiance at a particular point. The choice of a pixel is important as long as the tissue surface is not smooth. In the center of the picture the grey value of 10x10 points were summed up and averaged giving the intensity of a described region. In such a way the whole reflectance spectrum at the investigated area was obtained. This method was verified with data obtained from the spectrometer.



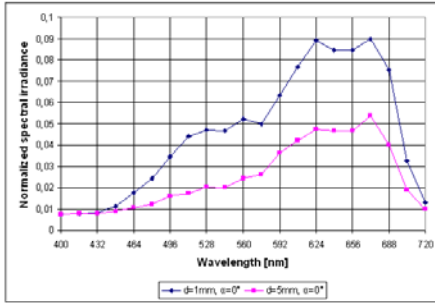
(a) Spectrum from the multi-spectral images (b) Spectrum from the spectrometer

Fig. 1. Comparison of two method for obtaining light spectrum

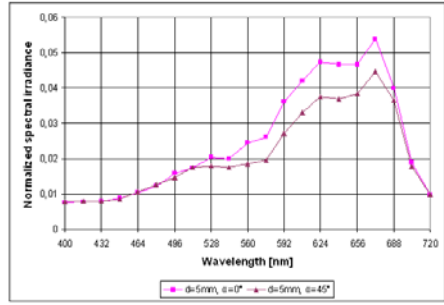
It can be clearly seen on Fig. 1 that both graphs show rather good correlation with each other. Data from spectrometer were sampled with greater frequency. This is why some information is missing in multi-spectral images approach. The smaller peak at 488nm wavelength is well visible at the graph from spectrometer, while on the other one, where the resolution is much lower, only slight slope changes were observed. However, the region of saturation is almost perfectly covered. It starts in both cases at 544nm and ends at 592nm for spectrum a) and 596nm for spectrum b). Finally, plots are going down and the value of spectral irradiance at 720nm wavelength was equal to 0,15. Such a good agreement between graphs made it possible to use multi-spectral images in further analysis.

Having a tool for obtaining reflectance spectrum of a given area, a set of multi-spectral images was taken for human wrist tissue. Experiment was performed for three different situations. It needs to be noted, that the camera is mounted just below the light source and they always point in the same direction. First, both devices were placed perpendicularly to the surface and 1mm away from the skin. Then the position of light was moved 5mm away from the object without changing its direction. Finally, the incident angle was set to 45°, but this time the position remained unchanged. In each situation multi-spectral images were taken and the comparison of resulting spectrum were presented on Fig. 2 and Fig. 3.

The first graph shows human skin reflectance spectrum for light source 1mm and 5mm away from the surface. It can be observed that for wavelengths between



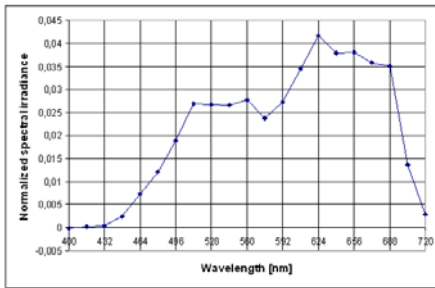
(a) Spectral irradiance for various distance d



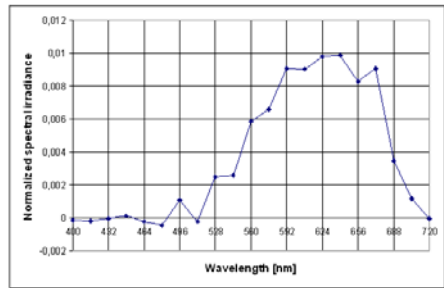
(b) Spectral irradiance for various incident angle α

Fig. 2. Spectrum of human skin tissue for various light source settings

500nm and 600nm plots differ by constant value. The same was noticed for range 600nm and 700nm, but this time the difference was greater. It means, that moving away the light source from the investigated tissue surface, the changes in the reflectance spectrum do not occur linearly, but are strongly wavelength dependent. It pays attention to the fact that probably there are 2 dominant absorbers in the human skin, which attenuate the light intensity in a different way.



(a) Spectral difference for various distance d



(b) Spectral difference for various incident angle α

Fig. 3. Spectral difference of human skin tissue for various light source settings

Another observation was made for reflectance spectrum, when the incident angle of light source was changed from the direction completely perpendicular i.e. 0° to 45° . It is worth to stress out that also in this case there are remarkable changes in skin color. As it can be seen the intensity for wavelengths between 400nm and 550nm remains almost the same. The changes are only noticeable for range 550nm to 675nm. It is probably due to the fact that scattering properties of human skin tissue also depends on the wavelength. Moreover, light entering the tissue at greater angle penetrates the medium only shallowly before it escapes. This results in the smaller photon absorption than for the case where the angle was equal to 0° and the mean free path of photon was much longer.

4 Application of Multi-spectral Images

Multi-spectral images can be also utilized in the other way. Two patients with BCC (Basal Cell Cancer) - very popular skin cancer - were investigated. Images for different wavelength were taken to obtain the whole reflectance spectrum, exactly in the same way as it was before for healthy human skin. A special care was needed to record only diseased part of a tissue. Both graphs for healthy and diseased skin are presented in Fig. 4 and Fig. 5.

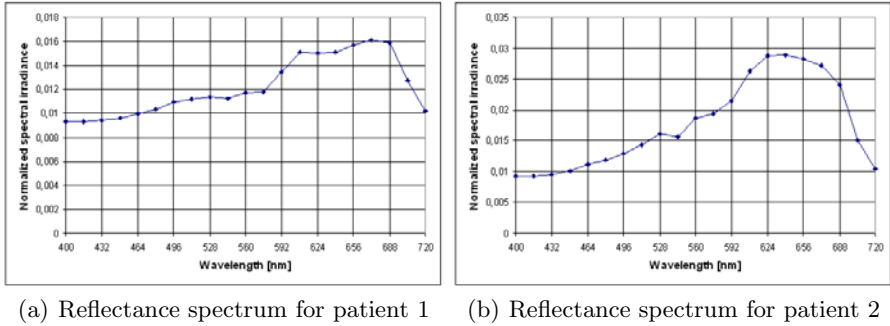


Fig. 4. Reflectance spectrum of human healthy skin tissue obtained from 2 patients

It is important to remark that fluorescent properties of a tissue can be enhanced additionally by fluorophores like endo-protoporphyrin IX. It induces functional and structural tissue modifications by penetrating into human red blood cells, hemoglobin and myoglobin, and releasing oxygen.

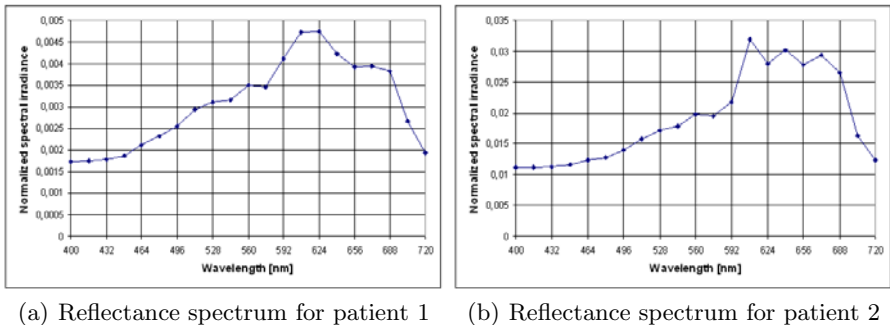



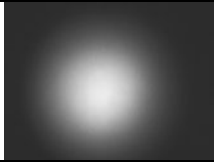
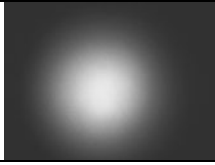
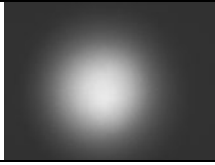
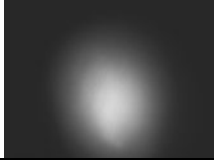
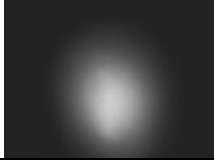


Fig. 5. Reflectance spectrum of human skin tissue with tumor obtained from 2 patients

This leads to morphological changes of the cells and increased sensitivity to light illumination. This ability is very widely used during cancer monitoring and detection. Photodynamic diagnosis is of great interest for medicine [8].

Both graphs in Fig. 5 looks very similar. For wavelengths between 400nm and 450nm spectrum is almost constant, then it starts to grow reaching its

maximal value somewhere around 610nm, finally it falls down a little and remains constant till almost 700nm. This highest peak visible for both patients allows to distinguish between healthy and diseased tissue. It is strictly related to the fluorescent properties of endo-protoporphyrin IX that was collected in cancer cells. However, that highest value is difficult to find in healthy tissue images (see Fig. 4). This property cannot be simply recognized during medical examination. For this particular reason multi-spectral images are very handy. Images below are of special interest (see Table 1).

Table 1. Multi-spectral images of human skin tissue with cancer

Wavelengths			
608nm	640nm	656nm	672nm
Patient 1			
			
Patient 2			
			

The idea behind them is to use the property of both plots where for 608nm the maximal value is observed, but for values between 640nm and 672nm the spectrum seems to be constant. Applying some simple image transformations it was possible to unambiguously determine whether the observed tissue is diseased or healthy. The following formula for calculating final image was proposed:

$$Img = Img_{608} - \frac{Img_{640} + Img_{656} + Img_{672}}{3}, \tag{1}$$




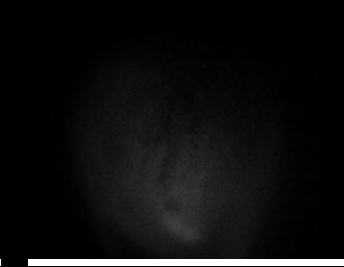
where:

Img_x - multi-spectral image for wavelength x,

It is the average of three multi-spectral images from range 640nm-672nm subtracted from the image with maximal peak. Table 2 presents resulting images for healthy tissue and tissue with BCC.

Obtained images are very satisfactory. For the healthy parts of the skin completely black pictures are identified, which means that reflectance spectrum between 608nm and 672nm is mainly constant. Next resulting images for BCC tissue were presented. They are not black any more, but some illuminations are visible.

Table 2. Images obtained after transformation from Eq. (1)

	Patient 1	Patient 2
Healthy tissue		
Tissue with Bcc		

This situation can only happen for 608nm wavelength highest intensity value. Described observation applied to the real photodynamic diagnosis can be very useful and can increase the cancer detection rate during medical examination.

5 Conclusions

Multi-spectral images have a very big potential and can be used in a number of different applications. It was showed that they can be utilized to obtain full reflectance spectrum of the investigated area that is comparable to data taken from the spectrometer. In fact the resolution of obtained graphs using this method is rather low, but gives a possibility to analyze them by image examination. Having this tool available in the laboratory, some investigations of human skin optical properties were performed. It turned up that moving light spectrum away from the tissue surface or changing the angle of incidence highly influence the reflectance spectrum. Finally, multi-spectral images were applied to cancer recognition process. Additional precessing and image manipulation techniques gave very promising results. They extract only important features of the pictures, which allow to clearly identify suspected part of the skin. The only thing that left to do is to verify this model with larger number of patients. Unfortunately multi-spectral images of other BCC cases were not available. The more accurate validation is going to be done in the future.

References

1. Nielsen, K.P., Zhao, L., Stamnes, J.J., Stamnes, K., Moan, J.: The optics of human skin: Aspects important for human health. In: Proceedings from the symposium Solar Radiation and Human Health (2008)
2. Katikaa, K.M., Piona, L., Dippleb, K., Levinc, S., Blackwell, J., Berberoglu, H.: In-Vivo Time-Resolved Autofluorescence Measurements on Human Skin. In: Proceedings of the SPIE, vol. 6078, pp. 83–93 (2006)
3. Meglinski, I.V., Matcher, S.J.: Computer simulation of the skin reflectance spectra. *Computer Methods and Programs in Biomedicine* 70(2), 179–186 (2003)
4. Wilkie, A., Weidlich, A., Larboulette, C., Purgathofer, W.: A Reflectance Model for Diffuse Fluorescent Surfaces. In: Proceedings of Graphite 2006, pp. 321–328 (2006)
5. Palmer, G.M., Ramanujam, N.: Monte-Carlo-based model for the extraction of intrinsic fluorescence from turbid media. *Journal of Biomedical Optics* 13(2) (2008)
6. Barton, J.K., Pfefer, T.J., Welch, A.J., Smithies, D.J., Nelson, J.S., van Gemert, M.J.C.: Optical Monte Carlo modeling of a true port wine stain anatomy. *Optics Express* 2(9), 391–396 (1998)
7. Latimer, P., Pyle, B.E.: Light Scattering at various angles. Theoretical predictions of the effects of particle volume changes. *Biophysical Journal* 12, 764–773 (1972)
8. Sil, S., Bose, T., Roy, D., Chakraborti, A.S.: Protoporphyrin IX-induced structural and functional changes in human red blood cells, haemoglobin and myoglobin. *Journal of Biosciences* 29(3), 281–291 (2004)

Can the Coronary Artery Centerline Extraction in Computed Tomography Images Be Improved by Use of a Partial Volume Model?

Maria A. Zuluaga^{1,2}, Edgar J.F. Delgado Leyton¹, Marcela Hernández Hoyos¹,
and Maciej Orkisz²

¹ Grupo IMAGINE, Grupo de Ingeniería Biomédica
Universidad de los Andes, Bogotá, Colombia
{ale-zulu,e-delga,marc-her}@uniandes.edu.co

² Université de Lyon; Université Lyon 1; INSA-Lyon;
CNRS UMR5220; INSERM U630; CREATIS, F-69621 Villeurbanne, France
{zuluaga,orkisz}@creatis.insa-lyon.fr

Abstract. We propose the use of a statistical partial volume (PV) model to improve coronary artery tracking in 3D cardiac computed tomography images, combined with a modified centerline extraction algorithm. PV effect is a challenge when trying to separate arteries from blood-filled cardiac cavities, causing leakage and erroneous segmentations. We include a Markov Random Field with a modified weighting scheme. First, synthetic phantoms were used to evaluate the robustness and accuracy of PV detection, as well as to determine the best settings. Average Dice similarity index obtained for PV voxels was 86%. Then cardiac images from eight patients were used to evaluate the usefulness of PV detection to separate real arteries from cavities, compared to Fuzzy C-means classification. Our PV detection scheme reduced approximately by half the number of leakages between artery and cavity. The new version of artery centerline extraction algorithm takes advantage of the PV detection capacity to separate arteries from cavities and to retrieve low-signal small vessels. We show some preliminary qualitative results of the complete method.

1 Introduction

Vascular diseases remain one of the leading causes of death throughout the world. Non-invasive (magnetic resonance angiography - MRA) and minimally-invasive (computed tomography angiography - CTA) 3D imaging techniques are increasingly used to assess the severity of atherosclerotic lesions that lead to arterial lumen narrowing (stenosis) and wall vulnerability. However, the assessment of such images requires appropriate processing tools. One of the most useful tools is the extraction of the vascular centerline. Many algorithms have been published during the last fifteen years (see [1,2] for surveys). Most of them assume that the vascular lumen is brighter than the surrounding tissues, which is achieved

in MRA images of still arterial regions (cerebral, peripheral, etc.) by use of a contrast agent injected into the blood, and by a digital subtraction of the image volume acquired before and after contrast injection. In CTA, double acquisition is not available owing to irradiation limitations, so the subtraction cannot be performed. Similarly, the subtraction does not remove the surrounding tissues in moving regions, namely coronary arteries. In coronary disease assessment, multi-detector CTA is often preferred to MRA, owing to its finer spatial resolution. Another advantage of the CTA is an approximate previous knowledge of image intensities corresponding to different tissues densities, expressed in Hounsfield units. However, a recent international competition, MICCAI Workshop - Coronary Artery Tracking Grand Challenge [3] (<http://coronary.bigr.nl/>), demonstrated the limits of the existing methods in terms of accuracy, computational time and quantity of interaction. We participated in this competition with a method initially developed for MRA images, based on an elastic model of the centerline and multi-scale analysis of image moments [1]. To be applicable to CTA coronary images, the method required a careful separation of the arteries from the cavities, by use of an interactive thresholding [4]. The goal of the current work is the automation of this stage, so as to reduce the user-dependence and make the processing time compatible with the clinical practice.

One of the main challenges when trying to separate the arteries from blood-filled cardiac cavities, is the boundary blurring owing to partial volume (PV) effect, which means that voxel intensities are formed by a mixture of different tissues. PV effect is reinforced in cardiac CTA images due to motion artifacts. Attempts to achieve artery/cavity separation include such approaches as leakage detection during region growing [5,6], mathematical morphology [7] and classification based on metrics combining intensities and image moments [8].

Statistical modeling of PV has been used in brain imaging [9,10]. We recently showed that the improvement of separation between coronary arteries and heart cavities by use of PV models is feasible [11]. Here, we report new results and show how the PV detection can be combined with the centerline extraction algorithm. The remaining sections first summarize the respective components of the method: PV detection [2] and centerline extraction [3]. Then new quantitative results of PV detection are reported [4], which justify its usefulness in the subsequent segmentation task. Finally, the combination with centerline extraction is described along with first qualitative results [5].

2 PV Detection

A CTA image is modeled by a union of *pure* regions, *i.e.* voxels made up of only one tissue type, and of PV layers possibly located on the boundaries between these regions. Our model comprises three *pure* classes: blood (B), lung (L) and muscle (M), corresponding to the modes of the typical cardiac CT histograms, and three *mixed* classes corresponding to the PV effect: BL , BM and ML . Please note that calcifications often present in the CT images of patients with coronary disease, are not considered here as a separate pure class. Indeed, owing

to their relatively small size and large intensity range, a corresponding histogram mode cannot be identified. The probability distributions of the pure classes, $C \in \{B, L, M\}$, can be modeled by Gaussian distributions with respective mean values μ_C and standard deviations σ_C :

$$p(x_i|C) = G(x_i; \mu_C, \sigma_C) = \frac{1}{\sqrt{2\pi}\sigma_C} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right), \tag{1}$$

where x_i denotes the intensity of i -th voxel. The mixed classes can be modeled by a distribution that follows a linear combination of two Gaussians. Let C be any combination of pure classes k and l , i.e. $C \in \{BL, BM, ML\}$, with $\alpha \in [0, 1]$. The marginal probability density [9] can then be expressed as:

$$p(x_i|C) = \int_0^1 G(x_i; \mu_C(\alpha), \sigma_C(\alpha)) d\alpha, \tag{2}$$

$$\mu_C(\alpha) = \alpha\mu_k + (1 - \alpha)\mu_l, \quad \sigma_C(\alpha) = \sqrt{\alpha\sigma_k^2 + (1 - \alpha)\sigma_l^2}. \tag{3}$$

Based on the observations X , a label $e_i \in \{e_B, e_L, e_M, e_{BL}, e_{BM}, e_{ML}\}$ is assigned to each voxel, which attempts to denote the tissue type the voxel i belongs to. The likelihood for all observations in the image, given a labeled image E , can be expressed as:

$$p(X|E) = \prod_i p(x_i|e_i). \tag{4}$$

A Markov random field is included in the model to take into account the spatial correlation of image intensities. Here we use the Potts model [10]:

$$p(E) = \frac{1}{Z} \exp\left(-\beta \sum_i \sum_{j \in N_i} \frac{\varphi(e_i, e_j)}{d(i, j)}\right), \tag{5}$$

where N_i represents the neighborhood of voxel i , $\varphi(e_i, e_j)$ is a weighting function that describes the prior knowledge of different neighboring tissues configurations, $d(i, j)$ is the distance between voxels i and j , β controls the strength of the prior and the partition function Z is considered as a scaling constant.

Typically the weights $\varphi(e_i, e_j)$ are chosen so as to favor neighborhoods containing the same label, which would equally encourage large pure regions and PV regions. Since PV voxels actually form thin layers around bigger pure regions, we make difference between neighborhoods of pure voxels and PV voxels:

$$\varphi(e_i, e_j) = \begin{cases} -3 : e_i = e_j \in \{e_B, e_L, e_M\} \\ -2 : e_i, e_j \text{ share one component} \\ -1 : e_i = e_j \in \{e_{BL}, e_{BM}, e_{ML}\} \\ 1 : \text{otherwise} \end{cases} \tag{6}$$

Labels "share one component" if one of them corresponds to a mixture of tissues k and l , while the other one corresponds to one of the pure tissues (k or l).

Using Bayes' formula, a maximum *a posteriori* classifier is constructed, which maximizes:

$$p(E|X) = \frac{p(X|E)p(E)}{p(X)}. \quad (7)$$

With a data-attachment term defined by equations 1 and 2, for pure and mixed classes respectively, and with a prior defined by equation 5, the labeled image E^* with the highest conditional probability should satisfy the following:

$$E^* = \operatorname{argmax}_E \left[\sum_i \log p(x_i|e_i) + \log p(E) \right]. \quad (8)$$

Its initialization is performed using the Fuzzy C-means algorithm [13] that performs an initial labeling of the image with only pure classes and provides mean values μ_C and standard deviations σ_C for each class.

3 Centerline-Tracking Algorithm

The algorithm extracts the centerline of one vessel of interest, starting from a point within its lumen, and includes the following steps:

- estimation (refinement) of the current point location, based on centroid information and restricted by continuity and smoothness constraints,
- estimation of the local orientation of the vessel, based on inertia matrix eigenvectors,
- prediction of the next point based on the estimated location and orientation.

Image moments are computed within a spherical sub-volume called *analysis cell*. We use a multi-scale framework to determine the cell size that is locally most suitable. The tracking process is carried out in two opposite directions from the starting point, and stops when one of the stopping criteria is encountered. In coronary CTA images, the tracking process is restricted to a subvolume masked with a presegmentation result \mathcal{V}_{pre} . As the details of the method were described elsewhere, here we only focus in the aspects useful to understand the innovation: criteria used to make evolve the cell size and to stop the tracking process.

When the cell is included within a vessel, its content has no privileged orientation and the inertia moments are identical for any axis. Conversely, when the cell is large enough to contain a cylindrical portion of the vessel, there is only one axis, corresponding to the local orientation of the vessel, around which the cylinder would spin with minimum inertia moment. The adaptation of the size ρ of the cell is carried out by "inflating" the cell as long as all three eigenvalues of the inertia matrix are approximately equal to each other, or by "deflating" it when the eigenvalues are significantly different. After sorting the eigenvalues so that $\lambda_1 \leq \lambda_2 \leq \lambda_3$, this is measured by the following function:

$$m_{iso}(\rho) = \psi(\lambda_1, \lambda_2) \cdot \psi(\lambda_2, \lambda_3) \quad \text{with} \quad \psi(\lambda_v, \lambda_w) = \left(\frac{\lambda_v}{\lambda_w} \right)^\xi, \quad (9)$$

the value of which is close to 1 when the three eigenvalues are approximately equal. The function ψ measures the similarity between two eigenvalues $\lambda_v \leq \lambda_w$. The coefficient ξ controls its selectivity. The structure contained in the cell is considered isotropic (spherical) as long as m_{iso} is greater than a threshold.

Two criteria are defined to stop the tracking of an artery when reaching its proximal (aortic) or distal end, respectively. If the cell reaches a radius as large as $\rho_{max} = 15$ voxels, and more than 90% of the voxels within it belong to \mathcal{V}_{pre} , we consider that the proximal end of the coronary artery was reached. If less than 30% of the voxels within the current cell (with a radius adapted to the local size of the vessel and centered at the current end-point of the centerline) belong to \mathcal{V}_{pre} , we consider that the distal end of the artery was reached.

4 Results of PV Detection

The evaluation was carried out using simulated phantoms and real images of patients. The phantoms represented an artery close to a cavity, with variable distance, radius and noise levels. The segmented PV, denoted X was compared to the ground truth Y via the Dice similarity index: $D = 2|X \cap Y| / (|X| + |Y|)$. For all vessel diameters and distances to the cavity, the Dice index with the weighting function defined by equation 6 was higher than with the classical one, *e.g.*: with signal/noise ratio equal to 8.5 these values increased from the range 0.80 – 0.84 to 0.84 – 0.88.

32 coronary arteries in 8 datasets obtained from the Rotterdam Coronary Artery Algorithm Evaluation Framework [14] were used to assess the method in real patients. The detection of PV is expected to improve the separation between vessels and cavities, but may potentially create discontinuities in thin vessels. To evaluate these two aspects, we proceeded as follows. A seed point was chosen within the artery (blood class) and connectivity was applied, starting from this point, in the images with and without (Fuzzy C-means result) PV detection. As

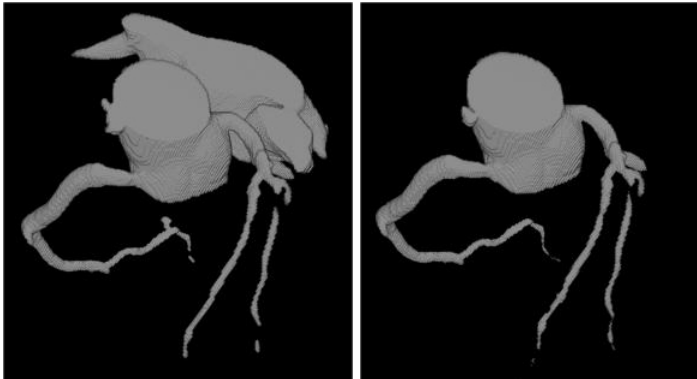


Fig. 1. Example of coronary arteries segmented without (left) and with (right) detection of PV. The use of PV permitted to separate the cavity.

many seeds were added as necessary to extract the whole artery. Each new seed was due to a discontinuity. We also counted the structures "stuck" to the arteries of interest. Without PV detection, in 32 arteries we found 15 discontinuities and 14 adjacent structures not separated. The PV detection did not change the number of discontinuities, while the number of non separated structures was reduced to 12 with a standard weighting function and to 8 with our proposal (Figure III). A visual inspection showed that small low-intensity vessels lost in the Fuzzy C-means classification (labeled as muscle e_M), were often distinguished from the muscle (labeled as PV e_{BM}) with our method.

5 Modified Centerline Extraction with PV Detection

Based on these results, the tracking algorithm was first kept unchanged and the interactive presegmentation step was replaced by the automatic Fuzzy C-means classification followed by PV detection. The mask \mathcal{V}_{pre} was defined as the set of voxels labeled e_B and connected to the seed point. This effectively prevented leakage into cavities, each time PV succeeded in separating the artery from the neighboring structures. However despite the fact that no creation of new discontinuities was observed in the above described evaluation, in additional datasets we encountered this problem: as the PV voxels e_{BM} are basically selected from initial e_B and e_M located on the vessel boundary, in some narrowings PV obstructed the lumen, thus creating discontinuities. This led us to a modification of the tracking algorithm. Actually, we now use two versions: the previously described multi-scale version is first used as long as the distal end of \mathcal{V}_{pre} is not reached, then we switch to a mono-scale version with a different stopping criterion and a different mask.

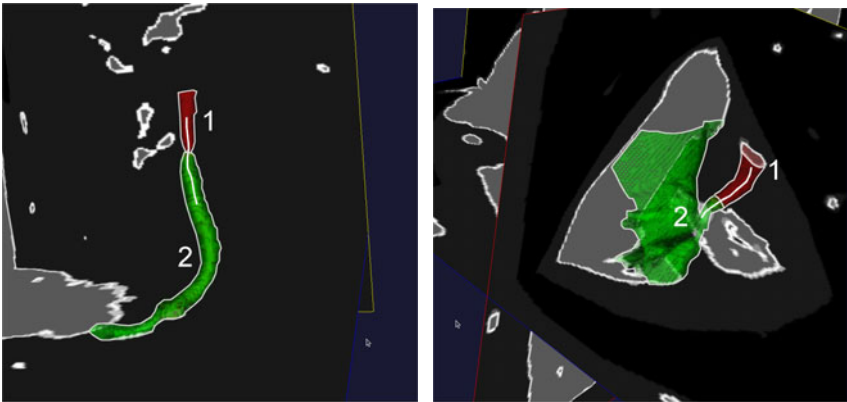


Fig. 2. Centerline tracking in presence of discontinuities created by PV. In red (1): fragment of blood mask used in the first phase. In green (2): PV mask used in the second phase. The centerline correctly continues in the PV-masked artery (left). It stops when arriving to a cavity (right).

The mask \mathcal{V}_{pre} is now composed of e_{BM} voxels connected to the previous centerline points and e_B voxels connected to the predicted point. At each iteration, only the connected voxels of the mask contained within the cell are considered, *i.e.* we discard connections through voxels located outside the cell. In this way, we can use a relatively big cell, without the risk of including nearby structures. The cell size corresponds to twice the average size of the analysis cells used before the narrowing. The tracking continues only in the distal direction, as long as the contents of the cell is cylindrical. The criterion is similar to equation 9, but builds on the fact that in a cylinder $\lambda_1 \cong 0$ and $\lambda_2 \cong \lambda_3$:

$$m_{cyl} = (1 - \psi(\lambda_1, \lambda_2)) \cdot \psi(\lambda_2, \lambda_3). \quad (10)$$

The tracking process stops when m_{cyl} falls below an empirically fixed threshold. Figure 2 shows the behavior of the modified algorithm in a narrowing and near a cavity.

6 Discussion and Conclusion

In this paper we have introduced the use of statistical PV models for the improvement of coronary artery tracking, combined with a modified centerline extraction algorithm. PV detection can help to separate the arteries from the nearby structures of similar intensity. It also permits to detect low signal arteries. However, these are labeled as PV and it is necessary to find a mechanism that can distinguish these arteries from real PV. To overcome this problem in our centerline tracking method, we used a shape constraint that permits to track a PV-labeled artery as long as its shape is cylindrical. This approach sometimes fails on bifurcations, where the cylindrical constraint does not hold. We will investigate the inclusion of bifurcation detection by the analysis of connected components of the mask on the cell surface [8]. The whole method will then be evaluated by use of the framework publicly available at <http://coronary.bigr.nl/>. Another issue that remains to be solved is the high computational cost of the PV labeling. To cope with it, future work will be focused on performing it only locally, within the artery tracking scheme. Let us note that the implementation of the algorithms in a graphical interface was greatly facilitated using CreaTools [1] software suite.

Acknowledgements

This work has been supported by the Région Rhône-Alpes (France) via the Simed project of the ISLE research cluster, by ECOS-Nord project #C07M04, and by Colombian grants: Proyecto CIFI-Uniandes 54-2009 and Programa Jóvenes Investigadores de Colciencias (E. Delgado's MSc project). M.A. Zuluaga's PhD project is supported by a Colciencias grant.

¹ <http://www.creatis.insa-lyon.fr/site/en/creatools>

References

1. Hernández Hoyos, M., Serfaty, J.M., Maghiar, A., Mansard, C., Orkisz, M., Magnin, I.E., Douek, P.C.: Assessment of carotid artery stenoses in 3D contrast-enhanced magnetic resonance angiography, based on improved generation of the centerline. *Machine Graphics and Vision* 14(4), 349–378 (2005)
2. Lesage, D., Angelini, E.D., Bloch, I., Funka-Lea, G.: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis* 13(6), 819–845 (2009)
3. Schaap, M., Metz, C., van Walsum, T., van der Giessen, A.G., Weustink, A.C., Mollet, N.R., et al.: Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. *Medical Image Analysis* 13(5), 701–714 (2009)
4. Hernández Hoyos, M., Zuluaga, M.A., Lozano, M., Prieto, J.C., Douek, P.C., Magnin, I.E., Orkisz, M.: Coronary centerline tracking in CT images with use of an elastic model and image moments. *Midas Journal* (2008), <http://hdl.handle.net/10380/1401>
5. Metz, C., Schaap, M., van der Giessen, A., van Walsum, T., Niessen, W.J.: Semi-automatic coronary artery centerline extraction in computed tomography angiography data. In: *Proc. of ISBI*, pp. 856–859 (2007)
6. Renard, F., Yang, Y.: Image analysis for detection of coronary artery soft plaques in MDCT images. In: *Proc. of ISBI*, pp. 25–28 (2008)
7. Luengo-Oroz, M.A., Ledesma-Carbayo, M.J., Gómez-Diego, J.J., García-Fernández, M.A., Desco, M., Santos, A.: Extraction of the coronary artery tree in cardiac computer tomographic images using morphological operators. In: Sachse, F.B., Seemann, G. (eds.) *FIHM 2007. LNCS*, vol. 4466, pp. 424–432. Springer, Heidelberg (2007)
8. Carrillo, J.F., Hernández Hoyos, M., Dávila, E.E., Orkisz, M.: Recursive tracking of vascular tree axes in 3D medical images. *Int. J. Comp. Assisted Radiol. Surg.* 1(6), 331–339 (2007)
9. Santago, P., Gage, H.D.: Quatification of MR brain images by mixture density and partial volume modeling. *IEEE Trans. Med. Imag.* 12(3), 566–574 (1993)
10. Shattuck, D., Sandor-Leahy, S., Schaper, K., Rottenberg, D., Leahy, R.: Magnetic resonance image tissue classification using a partial volume model. *Neuroimage* 14(5), 856–876 (2001)
11. Zuluaga, M.A., Hernández Hoyos, M., Orkisz, M.: Evaluation of partial volume effects in computed tomography for the improvement of coronary artery segmentation. 23rd CARS - Computer Assisted Radiology and Surgery, *Int. J. Comp. Assisted Radiol. Surg.* 4, 40–41 (June 2009)
12. Besag, J.: On the statistical analysis of dirty picture. *Journal of the Royal Statistical Society, Series B* 48(3), 259–302 (1986)
13. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3, 32–57 (1973)
14. Metz, C., Schaap, M., van Walsum, T., van der Giessen, A., Weustink, A., Mollet, N.R.A., Krestin, G., Niessen, W.J.: 3D Segmentation in the Clinic: A Grand Challenge II - Coronary Artery Tracking. *Midas Journal* (2008), <http://hdl.handle.net/10380/1399>

Author Index

- Álvarez, Hugo II-235
Arcay, Bernardino I-241
Austvoll, Ivar I-201
Azencot, Jacques I-27
- Babout, Laurent II-57
Barnes, Michael I-209
Barrière, Olivier I-342
Bator, Marcin I-265
Beleznai, Csaba I-283
Beran, Vítězslav I-1
Bernaś, Marcin I-217
Bezin, Richard I-225
Bielecka, Marzena I-11
Bielecki, Andrzej I-11
Borgefors, Gunilla II-243
Borkowski, Dariusz I-233
Borro, Diego II-235
Bouakaz, Saïda I-333
Bóveda, Carmen I-241
Briceño, Fernando II-142
Briceño, Juan C. II-142
Broda, Bartosz II-284
- Caicedo, Sandra L. II-142
Castro, Alfonso I-241
Chan, KinYip I-249
Chevallier, Sylvain I-257
Chmielewski, Leszek J. I-265
Choraś, Michał I-275
Chowdhury, Shyama Prosad I-19
Cielniak, Grzegorz I-209
Crespin, Benoit I-225, I-316
Croonen, Gerardus I-283
Cuperlier, Nicolas I-257
Cyganeck, Bogusław I-292
Czajewski, Witold I-308
Czajka, Lukasz II-1
Czarnecki, Wojciech I-300
- Daniel, Piotr II-33
Darles, Emmanuelle I-316
Das, Amit Kumar I-19
Debled-Rennesson, Isabelle I-59
Delgado Leyton, Edgar J.F. II-385
- Dellepiane, Sara I-324
Dellepiane, Silvana G. I-324
Domański, Marek I-384, II-25
Duckett, Tom I-209
Duran, Olga I-249
Dutreve, Ludovic I-333
- Echelard, Antoine I-342
- Fae'z, Karim I-352
Faridnia, Sa'ed I-352
Ferscha, Alois II-334
Filipowicz, Wiktor I-105
Flizikowski, Adam I-275
Flórez Valencia, Leonardo I-27
Forczmański, Paweł I-114, I-360
Fouladgaran, Maryam Panjeh II-166
Franke, Karl-Heinz II-81
Frejlichowski, Dariusz I-360,
I-368, I-376
- Gaussier, Philippe I-257
Gburek, Szymon I-300
Ghazanfarpour, Djamchid I-316
Gómez, Soledad II-142
Grajek, Tomasz I-384
Grzelak, Piotr I-35
- Habela, Piotr I-105
Harada, Koichi II-300
Hassaballah, M. I-392
Hernández Hoyos, Marcela II-385
Hołubowicz, Witold I-275
Hoyos, Marcela Hernández II-142
- Ibrahim, Rosziati II-134
Ido, Shinobu I-392
Ido, Shun I-392
Iwanowski, Marcin I-308, I-401, II-317
- Jabłoński, Bartosz I-122
Jakóbczak, Dariusz I-411
Janaszewski, Marcin II-57
Jasionowska, Magdalena I-420
Jędrasiak, Karol I-131, I-185, I-193

- Johansson, Carina B. II-243
 Jójczyk, Konrad I-35
 Josiński, Henryk I-131, I-185,
 I-193, II-325
 Józwiak, Rafał I-420
- Kaczmarek, Krzysztof I-105
 Kälviäinen, Heikki II-166
 Kanazawa, Tomonori I-392
 Kapec, Peter II-9
 Kartaszyński, Rafał Henryk II-17
 Karwowski, Damian II-25
 Kasiński, Andrzej II-251, II-260
 Kasprzak, Włodzimierz II-1
 Käyhkö, Jari II-166
 Kerautret, Bertrand I-59
 Klepaczko, Artur II-33
 Kohout, Josef II-41
 Kolingerová, Ivana II-41, II-350
 Komorowski, Jacek II-49
 Konieczny, Jacek II-118
 Korkosz, Mariusz I-11
 Kornev, Anatoly II-57
 Kornuta, Tomasz II-65
 Korzyńska, Anna I-401
 Kozik, Rafał I-275
 Kraft, Marek II-251
 Kruliková, Lenka II-73, II-342
 Krupiński, Robert I-139
 Krzeszowski, Tomasz I-147, I-155
 Kubertschak, Tim II-81
 Kulbacki, Marek I-105, I-122
 Kulikowski, Juliusz L. I-43
 Kurc, Maciej II-118
 Kwasnicka, Halina II-284
 Kwolek, Bogdan I-147, I-155,
 I-201, II-227
- Lachaud, Jacques-Olivier I-59
 Lensu, Lasse II-166
 Lévy, Véhel Jacques I-342
 Li, Longzhuang II-98
 Lindblad, Joakim II-243
 Lipowski, Jarosław Konrad II-89
 Liu, Yonghuai II-98
- Machleidt, Torsten II-81
 Maćkowiak, Przemysław II-110, II-118
 Maćkowiak, Sławomir II-118
 Mankki, Aki II-166
- Mardiak, Michał II-73
 Martinez, Jean-Luc I-163
 Martin, Ralph R. II-98
 Mazurek, Przemysław I-139
 Merienne, Frederic I-163
 Meseure, Philippe I-225
 Meyer, Alexandre I-333
 Michalak, Marcin II-126, II-325
 Miętus, Aleksandra II-150
 Mikołajczak, Paweł II-17
 Minetti, Irene I-324
 Mohd Noor, Helmi Adly II-134
 Moreno, Silvia II-142
- Nestler, Rico II-81
 Nguyen, Thanh Phuong I-59
 Nguyen, Van-Hanh I-163
- Okarma, Krzysztof II-150
 Ordys, Andrzej I-249
 Orgoniková, Tatiana II-342
 Orkisz, Maciej I-27, II-385
 Orvalho, Veronica I-333
- Pamula, Wiesław II-158
 Paradowski, Mariusz II-175
 Pardel, Przemysław Wiktor II-183
 Pavlovic, Juraj II-73
 Pazurek, Marek II-33
 Peyrat, Alexandre I-225
 Piecha, Jan II-193
 Pietruszka, Maria II-203
 Płaczek, Bartłomiej II-211
 Polański, Andrzej I-131, I-185, I-193
 Polec, Jaroslav II-73, II-342
 Porter-Sobieraj, Joanna II-276
 Pryczek, Michał I-35
 Przelaskowski, Artur I-69, I-420
- Rafferty, Karen I-19
 Raniszewski, Marcin II-219
 Rey, Alberto I-241
 Rokita, Przemysław II-49
 Rulf, Michał II-41
 Rymut, Bogusław II-227
- Saeed, Khalid II-334
 Sánchez, Jairo R. II-235
 Sarve, Hamid II-243
 Schmidt, Adam II-251, II-260
 Seweryn, Marcin I-114

- Skalski, Andrzej II-268
Skapin, Xavier I-225
Skomorowski, Marek I-11
Śluzek, Andrzej II-175
Smiatacz, Maciej I-300
Stanek, Michał II-284
Staniek, Marcin II-193
Stankiewicz, Olgierd II-292
Stereńczak, Krzysztof I-265
Strug, Barbara II-309
Strulovic, Tonny II-142
Strzeliński, Paweł I-265
Suciati, Nanik II-300
Świdorski, Rafał II-309
Świechowski, Maciej II-276
Świercz, Michał II-317
Świtoński, Adam I-131, I-185,
I-193, II-126, II-325
Szajerman, Dominik II-203
Szczepaniak, Piotr S. I-35, I-89
Szczepański, Adam II-334
Szczypiński, Piotr II-33
- Talbot, Hugues II-57
Teclaw, Mateusz II-150
Terraz, Olivier I-225
- Tomczyk, Arkadiusz I-35, I-89
Tóthová, Želmira II-342
Trzupek, Mirosław I-81
Turcza, Paweł II-268
- Uher, Václav II-41
- Vomáčka, Tomáš II-350
- Walczak, Stanisław I-89
Wegner, Krzysztof II-292
Wei, Baogang II-98
Wilkowski, Artur II-1
Wojciechowski, Adam II-357
Wojciechowski, Konrad I-131, I-147,
I-155, I-185, I-193, II-183, II-325
Wojciechowski, Wadim I-11
Wójcik, Krzysztof I-97
Wróblewski, Grzegorz II-357
- Zacher, Andrzej II-367, II-376
Zasada, Michał I-265
Zemčík, Pavel I-1
Zieliński, Bartosz I-11
Zieliński, Tomasz II-268
Zuluaga, Maria A. II-385