

George Bebis et al. (Eds.)

LNCS 6454

Advances in Visual Computing

6th International Symposium, ISVC 2010
Las Vegas, NV, USA, November/December 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

George Bebis Richard Boyle Bahram Parvin
Darko Koracin Ronald Chung Riad Hammound
Muhammad Hussain Tan Kar-Han Roger Crawfis
Daniel Thalmann David Kao Lisa Avila (Eds.)

Advances in Visual Computing

6th International Symposium, ISVC 2010
Las Vegas, NV, USA
November 29 – December 1, 2010
Proceedings, Part II

Volume Editors

George Bebis, E-mail: bebis@cse.unr.edu

Richard Boyle, E-mail: richard.boyle@nasa.gov

Bahram Parvin, E-mail: parvin@hpcrd.lbl.gov

Darko Koracin, E-mail: darko@dri.edu

Ronald Chung, E-mail: rchung@cuhk.edu.hk

Riad Hammound, E-mail: riad.hammound@dynavoxtech.com

Muhammad Hussain, E-mail: mhussain@ccis.edu.sa

Tan Kar-Han, E-mail: karhan.tan@hp.com

Roger Crawfis, E-mail: crawfis@cse.ohio-state.edu

Daniel Thalmann, E-mail: daniel.thalmann@epfl.ch

David Kao, E-mail: davidkao@nas.nasa.gov

Lisa Avila, E-mail: lisa.avila@kitware.com

Library of Congress Control Number: 2010939054

CR Subject Classification (1998): I.3, H.5.2, I.4, I.5, I.2.10, J.3, F.2.2, I.3.5

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

ISSN 0302-9743

ISBN-10 3-642-17273-3 Springer Berlin Heidelberg New York

ISBN-13 978-3-642-17273-1 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

It is with great pleasure that we present the proceedings of the 6th International, Symposium on Visual Computing (ISVC 2010), which was held in Las Vegas, Nevada. ISVC provides a common umbrella for the four main areas of visual computing including vision, graphics, visualization, and virtual reality. The goal is to provide a forum for researchers, scientists, engineers, and practitioners throughout the world to present their latest research findings, ideas, developments, and applications in the broader area of visual computing.

This year, the program consisted of 14 oral sessions, one poster session, 7 special tracks, and 6 keynote presentations. The response to the call for papers was very good; we received over 300 submissions for the main symposium from which we accepted 93 papers for oral presentation and 73 papers for poster presentation. Special track papers were solicited separately through the Organizing and Program Committees of each track. A total of 44 papers were accepted for oral presentation and 6 papers for poster presentation in the special tracks.

All papers were reviewed with an emphasis on potential to contribute to the state of the art in the field. Selection criteria included accuracy and originality of ideas, clarity and significance of results, and presentation quality. The review process was quite rigorous, involving two – three independent blind reviews followed by several days of discussion. During the discussion period we tried to correct anomalies and errors that might have existed in the initial reviews. Despite our efforts, we recognize that some papers worthy of inclusion may have not been included in the program. We offer our sincere apologies to authors whose contributions might have been overlooked.

We wish to thank everybody who submitted their work to ISVC 2010 for review. It was because of their contributions that we succeeded in having a technical program of high scientific quality. In particular, we would like to thank the ISVC 2010 Area Chairs, the organizing institutions (UNR, DRI, LBNL, and NASA Ames), the government and industrial sponsors (Air Force Research Lab, Intel, DigitalPersona, Equinox, Ford, Hewlett Packard, Mitsubishi Electric Research Labs, iCore, Toyota, Delphi, General Electric, Microsoft MSDN, and Volt), the international Program Committee, the special track organizers and their Program Committees, the keynote speakers, the reviewers, and especially the authors that contributed their work to the symposium. In particular, we would like to thank *Air Force Research Lab*, *Mitsubishi Electric Research Labs*, and *Volt* for kindly sponsoring four “best paper awards” this year.

We sincerely hope that ISVC 2010 offered opportunities for professional growth.

Organization

ISVC 2010 Steering Committee

Bebis George	University of Nevada, Reno, USA
Boyle Richard	NASA Ames Research Center, USA
Parvin Bahram	Lawrence Berkeley National Laboratory, USA
Koracin Darko	Desert Research Institute, USA

ISVC 2010 Area Chairs

Computer Vision

Chang Ronald	The Chinese University of Hong Kong, Hong Kong
Hammoud Riad	DynaVox Systems, USA

Computer Graphics

Hussain Muhammad	King Saud University, Saudi Arabia
Tan Kar-Han	Hewlett Packard Labs, USA

Virtual Reality

Crawfis Roger	Ohio State University, USA
Thalman Daniel	EPFL, Switzerland

Visualization

Kao David	NASA Ames Research Lab, USA
Avila Lisa	Kitware, USA

Publicity

Erol Ali	Ocali Information Technology, Turkey
----------	--------------------------------------

Local Arrangements

Regentova Emma	University of Nevada, Las Vegas, USA
----------------	--------------------------------------

Special Tracks

Porikli Fatih	Mitsubishi Electric Research Labs, USA
---------------	--

ISVC 2010 Keynote Speakers

Kakadiaris Ioannis	University of Houston, USA
Hollerer Tobias	University of California at Santa Barbara, USA
Stasko John	Georgia Institute of Technology, USA
Seitz Steve	University of Washington, USA
Pollefeys Marc	ETH Zurich, Switzerland
Majumder Aditi	University of California, Irvine, USA

ISVC 2010 International Program Committee

(Area 1) Computer Vision

Abidi Besma	University of Tennessee, USA
Abou-Nasr Mahmoud	Ford Motor Company, USA
Agaian Sos	University of Texas at San Antonio, USA
Aggarwal J. K.	University of Texas, Austin, USA
Amayeh Gholamreza	Eyecom, USA
Agouris Peggy	George Mason University, USA
Argyros Antonis	University of Crete, Greece
Asari Vijayan	University of Dayton, USA
Basu Anup	University of Alberta, Canada
Bekris Kostas	University of Nevada at Reno, USA
Belyaev Alexander	Max-Planck-Institut fuer Informatik, Germany
Bensrhair Abdelaziz	INSA-Rouen, France
Bhatia Sanjiv	University of Missouri-St. Louis, USA
Bimber Oliver	Johannes Kepler University Linz, Austria
Bioucas Jose	Instituto Superior Tecnico, Lisbon, Portugal
Birchfield Stan	Clemson University, USA
Bourbakis Nikolaos	Wright State University, USA
Brimkov Valentin	State University of New York, USA
Campadelli Paola	Università degli Studi di Milano, Italy
Cavallaro Andrea	Queen Mary, University of London, UK
Charalampidis Dimitrios	University of New Orleans, USA
Chellappa Rama	University of Maryland, USA
Chen Yang	HRL Laboratories, USA
Cheng Hui	Sarnoff Corporation, USA
Cochran Steven Douglas	University of Pittsburgh, USA
Cremers Daniel	University of Bonn, Germany
Cui Jinshi	Peking University, China
Darbon Jerome	CNRS-Ecole Normale Superieure de Cachan, France
Davis James W.	Ohio State University, USA

Debrunner Christian	Colorado School of Mines, USA
Demirdjian David	MIT, USA
Duan Ye	University of Missouri-Columbia, USA
Doulamis Anastasios	National Technical University of Athens, Greece
Dowdall Jonathan	510 Systems, USA
El-Ansari Mohamed	Ibn Zohr University, Morocco
El-Gammal Ahmed	University of New Jersey, USA
Eng How Lung	Institute for Infocomm Research, Singapore
Erol Ali	Ocali Information Technology, Turkey
Fan Guoliang	Oklahoma State University, USA
Ferri Francesc	Universitat de Valencia, Spain
Ferryman James	University of Reading, UK
Foresti GianLuca	University of Udine, Italy
Fowlkes Charless	University of California, Irvine, USA
Fukui Kazuhiro	The University of Tsukuba, Japan
Galata Aphrodite	The University of Manchester, UK
Georgescu Bogdan	Siemens, USA
Gleason, Shaun	Oak Ridge National Laboratory, USA
Goh Wooi-Boon	Nanyang Technological University, Singapore
Guerra-Filho Gutemberg	University of Texas Arlington, USA
Guevara, Angel Miguel	University of Porto, Portugal
Gustafson David	Kansas State University, USA
Harville Michael	Hewlett Packard Labs, USA
He Xiangjian	University of Technology, Sydney, Australia
Heikkilä Janne	University of Oulu, Finland
Heyden Anders	Lund University, Sweden
Hongbin Zha	Peking University, China
Hou Zujun	Institute for Infocomm Research, Singapore
Hua Gang	Nokia Research Center, USA
Imiya Atsushi	Chiba University, Japan
Jia Kevin	IGT, USA
Kamberov George	Stevens Institute of Technology, USA
Kampel Martin	Vienna University of Technology, Austria
Kamberova Gerda	Hofstra University, USA
Kakadiaris Ioannis	University of Houston, USA
Kettebekov Sanzhar	Keane inc., USA
Khan Hameed Ullah	King Saud University, Saudi Arabia
Kim Tae-Kyun	University of Cambridge, UK
Kimia Benjamin	Brown University, USA
Kisacanin Branislav	Texas Instruments, USA
Klette Reinhard	Auckland University, New Zealand
Kokkinos Iasonas	Ecole Centrale Paris, France
Kollias Stefanos	National Technical University of Athens, Greece

Komodakis Nikos	Ecole Centrale de Paris, France
Kozintsev	Igor, Intel, USA
Kuno	Yoshinori, Saitama University, Japan
Kyungnam Kim	HRL Laboratories, USA
Latecki Longin Jan	Temple University, USA
Lee D. J.	Brigham Young University, USA
Li Chunming	Vanderbilt University, USA
Li Fei-Fei	Stanford University, USA
Lin Zhe	Adobe, USA
Lisin Dima	VidoeIQ, USA
Lee Seong-Whan	Korea University, Korea
Leung Valerie	Kingston University, UK
Leykin Alex	Indiana University, USA
Li Shuo	GE Healthcare, Canada
Li Wenjing	STI Medical Systems, USA
Liu Jianzhuang	The Chinese University of Hong Kong, Hong Kong
Loss Leandro	Lawrence Berkeley National Lab, USA
Ma Yunqian	Honyewell Labs, USA
Maeder Anthony	University of Western Sydney, Australia
Makris Dimitrios	Kingston University, UK
Maltoni Davide	University of Bologna, Italy
Mauer Georg	University of Nevada, Las Vegas, USA
Maybank Steve	Birkbeck College, UK
McGraw Tim	West Virginia University, USA
Medioni Gerard	University of Southern California, USA
Melenchón Javier	Universitat Oberta de Catalunya, Spain
Metaxas Dimitris	Rutgers University, USA
Miller Ron	Wright Patterson Air Force Base, USA
Ming Wei	Konica Minolta, USA
Mirmehdi Majid	Bristol University, UK
Monekosso Dorothy	Kingston University, UK
Mueller Klaus	SUNY Stony Brook, USA
Mulligan Jeff	NASA Ames Research Center, USA
Murray Don	Point Grey Research, Canada
Nait-Charif Hammadi	Bournemouth University, UK
Nefian Ara	NASA Ames Research Center, USA
Nicolescu Mircea	University of Nevada, Reno, USA
Nixon Mark	University of Southampton, UK
Nolle Lars	The Nottingham Trent University, UK
Ntalianis Klimis	National Technical University of Athens, Greece
Or Siu Hang	The Chinese University of Hong Kong, Hong Kong
Papadourakis George	Technological Education Institute, Greece

Papanikolopoulos Nikolaos	University of Minnesota, USA
Pati Peeta Basa	First Indian Corp., India
Patras Ioannis	Queen Mary University, London, UK
Petrakis Euripides	Technical University of Crete, Greece
Peyronnet Sylvain	LRDE/EPITA, France
Pinhanez Claudio	IBM Research, Brazil
Piccardi Massimo	University of Technology, Australia
Pietikäinen Matti	LRDE/University of Oulu, Finland
Porikli Fatih	Mitsubishi Electric Research Labs, USA
Prabhakar Salil	DigitalPersona Inc., USA
Prati Andrea	University of Modena and Reggio Emilia, Italy
Prokhorov Danil	Toyota Research Institute, USA
Prokhorov Pylvanainen Timo	Nokia, Finland
Qi Hairong	University of Tennessee at Knoxville, USA
Qian Gang	Arizona State University, USA
Raftopoulos Kostas	National Technical University of Athens, Greece
Reed Michael	Blue Sky Studios, USA
Regazzoni Carlo	University of Genoa, Italy
Regentova Emma	University of Nevada, Las Vegas, USA
Remagnino Paolo	Kingston University, UK
Ribeiro Eraldo	Florida Institute of Technology, USA
Robles-Kelly Antonio	National ICT Australia (NICTA), Australia
Ross Arun	West Virginia University, USA
Salgian Andrea	The College of New Jersey, USA
Samal Ashok	University of Nebraska, USA
Sato Yoichi	The University of Tokyo, Japan
Samir Tamer	Ingersoll Rand Security Technologies, USA
Sandberg Kristian	Computational Solutions, USA
Sarti Augusto	DEI Politecnico di Milano, Italy
Savakis Andreas	Rochester Institute of Technology, USA
Schaefer Gerald	Loughborough University, UK
Scalzo Fabien	University of California at Los Angeles, USA
Scharcanski Jacob	UFRGS, Brazil
Shah Mubarak	University of Central Florida, USA
Shi Pengcheng	The Hong Kong University of Science and Technology, Hong Kong
Shimada Nobutaka	Ritsumeikan University, Japan
Singh Meghna	University of Alberta, Canada
Singh Rahul	San Francisco State University, USA
Skurikhin Alexei	Los Alamos National Laboratory, USA
Souvenir, Richard	University of North Carolina - Charlotte, USA

Su Chung-Yen	National Taiwan Normal University, Taiwan
Sugihara Kokichi	University of Tokyo, Japan
Sun Zehang	Apple, USA
Syeda-Mahmood Tanveer	IBM Almaden, USA
Tan Tieniu	Chinese Academy of Sciences, China
Tavakkoli Alireza	University of Houston - Victoria, USA
Tavares, Joao	Universidade do Porto, Portugal
Teoh Eam Khwang	Nanyang Technological University, Singapore
Thiran Jean-Philippe	Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Tistarelli Massimo	University of Sassari, Italy
Tschepnakis Gabriel	University of Miami, USA
Tsui T.J.	Chinese University of Hong Kong, Hong Kong
Trucco Emanuele	University of Dundee, UK
Tubaro Stefano	DEI, Politecnico di Milano, Italy
Uhl Andreas	Salzburg University, Austria
Velastin Sergio	Kingston University London, UK
Verri Alessandro	Università di Genova, Italy
Wang Charlie	The Chinese University of Hong Kong, Hong Kong
Wang Junxian	Microsoft, USA
Wang Song	University of South Carolina, USA
Wang Yunhong	Beihang University, China
Webster Michael	University of Nevada, Reno, USA
Wolff Larry	Equinox Corporation, USA
Wong Kenneth	The University of Hong Kong, Hong Kong
Xiang Tao	Queen Mary, University of London, UK
Xue Xinwei	Fair Isaac Corporation, USA
Xu Meihe	University of California at Los Angeles, USA
Yang Ruigang	University of Kentucky, USA
Yi Lijun	SUNY at Binghamton, USA
Yu Kai	NEC Labs, USA
Yu Ting	GE Global Research, USA
Yu Zeyun	University of Wisconsin-Milwaukee, USA
Yuan Chunrong	University of Tuebingen, Germany
Zhang Yan	Delphi Corporation, USA
Zhou Huiyu	Queen's University Belfast, UK

(Area 2) Computer Graphics

Abd Rahni Mt Piah	Universiti Sains Malaysia, Malaysia
Abram Greg	IBM T.J.Watson Reseach Center, USA
Adamo-Villani Nicoletta	Purdue University, USA

Agu Emmanuel	Worcester Polytechnic Institute, USA
Andres Eric	Laboratory XLIM-SIC, University of Poitiers, France
Artusi Alessandro	CaSToRC Cyprus Institute, Cyprus
Baciu George	Hong Kong PolyU, Hong Kong
Balcisoy Selim Saffet	Sabanci University, Turkey
Barneva Reneta	State University of New York, USA
Bartoli Vilanova Anna	Eindhoven University of Technology, The Netherlands
Belyaev Alexander	Max Planck-Institut fuer Informatik, Germany
Benes Bedrich	Purdue University, USA
Berberich Eric	Max-Planck Institute, Germany
Bilalis Nicholas	Technical University of Crete, Greece
Bimber Oliver	Johannes Kepler University Linz, Austria
Bohez Erik	Asian Institute of Technology, Thailand
Bouatouch Kadi	University of Rennes I, IRISA, France
Brimkov Valentin	State University of New York, USA
Brown Ross	Queensland University of Technology, Australia
Callahan Steven	University of Utah, USA
Chen Min	University of Wales Swansea, UK
Cheng Irene	University of Alberta, Canada
Chiang Yi-Jen	Polytechnic Institute of New York University, USA
Choi Min	University of Colorado at Denver, USA
Comba Joao	Univ. Fed. do Rio Grande do Sul, Brazil
Cremer Jim	University of Iowa, USA
Culbertson Bruce	HP Labs, USA
Debattista Kurt	University of Warwick, UK
Deng Zhigang	University of Houston, USA
Dick Christian	Technical University of Munich, Germany
DiVerdi Stephen	Adobe, USA
Dingliana John	Trinity College, Ireland
El-Sana Jihad	Ben Gurion University of The Negev, Israel
Entezari Alireza	University of Florida, USA
Fiorio Christophe	Université Montpellier 2, LIRMM, France
Floriani Leila De	University of Genoa, Italy
Gaither Kelly	University of Texas at Austin, USA
Gao Chunyu	Epson Research and Development, USA
Geist Robert	Clemson University, USA
Gelb Dan	Hewlett Packard Labs, USA
Gotz David	IBM, USA
Gooch Amy	University of Victoria, Canada

Gu David	State University of New York at Stony Brook, USA
Guerra-Filho Gutemberg	University of Texas Arlington, USA
Habib Zulfiqar	National University of Computer and Emerging Sciences, Pakistan
Hadwiger Markus	KAUST, Saudi Arabia
Haller Michael	Upper Austria University of Applied Sciences, Austria
Hamza-Lup Felix	Armstrong Atlantic State University, USA
Han JungHyun	Korea University, Korea
Hao Xuejun	Columbia University and NYSPI, USA
Hernandez Jose Tiberio	Universidad de los Andes, Colombia
Huang Mao Lin	University of Technology, Australia
Huang Zhiyong	Institute for Infocomm Research, Singapore
Joaquim Jorge	Instituto Superior Tecnico, Portugal
Ju Tao	Washington University, USA
Julier Simon J.	University College London, UK
Kakadiaris Ioannis	University of Houston, USA
Kamberov George	Stevens Institute of Technology, USA
Kim Young	Ewha Womans University, Korea
Klosowski James	AT&T Labs, USA
Kobbelt Leif	RWTH Aachen, Germany
Kuan Lee Hwee	Bioinformatics Institute, ASTAR, Singapore
Lai Shuhua	Virginia State University, USA
Lakshmanan Geetika	IBM T.J. Watson Reseach Center, USA
Lee Chang Ha	Chung-Ang University, Korea
Lee Tong-Yee	National Cheng-Kung University, Taiwan
Levine Martin	McGill University, Canada
Lewis Bob	Washington State University, USA
Li Frederick	University of Durham, UK
Lindstrom Peter	Lawrence Livermore National Laboratory, USA
Linsen Lars	Jacobs University, Germany
Loviscach Joern	Fachhochschule Bielefeld (University of Applied Sciences), Germany
Magnor Marcus	TU Braunschweig, Germany
MaJumder Aditi	University of California, Irvine, USA
Mantler Stephan	VRVis Research Center, Austria
Martin Ralph	Cardiff University, UK
McGraw Tim	West Virginia University, USA
Meenakshisundaram Gopi	University of California-Irvine, USA
Mendoza Cesar	NaturalMotion Ltd., USA
Metaxas Dimitris	Rutgers University, USA
Myles Ashish	University of Florida, USA
Nait-Charif Hammadi	University of Dundee, UK

Nasri Ahmad	American University of Beirut, Lebanon
Noma Tsukasa	Kyushu Institute of Technology, Japan
Okada Yoshihiro	Kyushu University, Japan
Olague Gustavo	CICESE Research Center, Mexico
Oliveira Manuel M.	Univ. Fed. do Rio Grande do Sul, Brazil
Ostromoukhov Victor M.	University of Montreal, Canada
Pascucci Valerio	University of Utah, USA
Peters Jorg	University of Florida, USA
Qin Hong	State University of New York at Stony Brook, USA
Razdan Anshuman	Arizona State University, USA
Reed Michael	Columbia University, USA
Renner Gabor	Computer and Automation Research Institute, Hungary
Rosenbaum Rene	University of California at Davis, USA
Rushmeier	Holly, Yale University, USA
Sander Pedro	The Hong Kong University of Science and Technology, Hong Kong
Sapidis Nickolas	University of Western Macedonia, Greece
Sarfraz Muhammad	Kuwait University, Kuwait
Scateni Riccardo	University of Cagliari, Italy
Schaefer Scott	Texas A&M University, USA
Sequin Carlo	University of California-Berkeley, USA
Shead Timothy	Sandia National Laboratories, USA
Sorkine Olga	New York University, USA
Sourin Alexei	Nanyang Technological University, Singapore
Stamminger Marc	REVES/INRIA, France
Su Wen-Poh	Griffith University, Australia
Staad Oliver	University of Rostock, Germany
Tarini Marco	Università dell'Insubria (Varese), Italy
Teschner Matthias	University of Freiburg, Germany
Tsong Ng Tian	Institute for Infocomm Research, Singapore
Umlauf Georg	HTWG Constance, Germany
Wald Ingo	University of Utah, USA
Wang Sen	Kodak, USA
Wimmer Michael	Technical University of Vienna, Austria
Wylie Brian	Sandia National Laboratory, USA
Wyman Chris	University of Iowa, USA
Yang Qing-Xiong	University of Illinois at Urbana, Champaign, USA
Yang Ruigang	University of Kentucky, USA
Ye Duan	University of Missouri-Columbia, USA
Yi Beifang	Salem State College, USA
Yin Lijun	Binghamton University, USA

Yoo Terry
Yuan Xiaoru
Zabulis Xenophon

Zhang Eugene
Zhang Jian Jun
Zordan Victor

National Institutes of Health, USA
Peking University, China
Foundation for Research and
Technology - Hellas (FORTH), Greece
Oregon State University, USA
Bournemouth University, UK
University of California at Riverside, USA

(Area 3) Virtual Reality

Alcañiz Mariano
Arns Laura
Balcisoy Selim
Behringer Reinhold
Benes Bedrich
Bilalis Nicholas
Blach Roland

Blom Kristopher
Borst Christoph
Brady Rachael
Brega Jose Remo Ferreira
Brown Ross

Bruce Thomas

Bues Matthias
Chen Jian
Cheng Irene
Coquillart Sabine
Craig Alan

Cremer Jim
Egges Arjan
Encarnacao L. Miguel
Figueroa Pablo
Fox Jesse
Friedman Doron
Froehlich Bernd
Gregory Michelle
Gupta Satyandra K.
Hachet Martin
Haller Michael
Hamza-Lup Felix
Hinkenjann Andre

Technical University of Valencia, Spain
Purdue University, USA
Sabanci University, Turkey
Leeds Metropolitan University UK
Purdue University, USA
Technical University of Crete, Greece
Fraunhofer Institute for Industrial
Engineering, Germany
University of Hamburg, Germany
University of Louisiana at Lafayette, USA
Duke University, USA
Universidade Estadual Paulista, Brazil
Queensland University of Technology,
Australia
The University of South Australia,
Australia
Fraunhofer IAO in Stuttgart, Germany
Brown University, USA
University of Alberta, Canada
INRIA, France
NCSA University of Illinois at
Urbana-Champaign, USA
University of Iowa, USA
Universiteit Utrecht, The Netherlands
Humana Inc., USA
Universidad de los Andes, Colombia
Stanford University, USA
IDC, Israel
Weimar University, Germany
Pacific Northwest National Lab, USA
University of Maryland, USA
INRIA, France
FH Hagenberg, Austria
Armstrong Atlantic State University, USA
Bonn-Rhein-Sieg University of Applied
Sciences, Germany

Hollerer Tobias	University of California at Santa Barbara, USA
Huang Jian	University of Tennessee at Knoxville, USA
Julier Simon J.	University College London, UK
Klinker Gudrun	Technische Universität München, Germany
Klosowski James	AT&T Labs, USA
Kozintsev	Igor, Intel, USA
Kuhlen Torsten	RWTH Aachen University, Germany
Liere Robert van	CWI, The Netherlands
Majumder Aditi	University of California, Irvine, USA
Malzbender Tom	Hewlett Packard Labs, USA
Mantler Stephan	VRVis Research Center, Austria
Meyer Joerg	University of California, Irvine, USA
Molineros Jose	Teledyne Scientific and Imaging, USA
Muller Stefan	University of Koblenz, Germany
Paelke Volker	Leibniz Universität Hannover, Germany
Pan Zhigeng	Zhejiang University, China
Papka Michael	Argonne National Laboratory, USA
Peli Eli	Harvard University, USA
Pettifer Steve	The University of Manchester, UK
Pugmire Dave	Los Alamos National Lab, USA
Qian Gang	Arizona State University, USA
Raffin Bruno	INRIA, France
Reiners Dirk	University of Louisiana, USA
Richir Simon	Arts et Metiers ParisTech, France
Rodello Ildeberto	University of Sao Paulo, Brazil
Santhanam Anand	MD Anderson Cancer Center Orlando, USA
Sapidis Nickolas	University of Western Macedonia, Greece
Schulze	Jurgen, University of California - San Diego, USA
Sherman Bill	Jurgen, Indiana University, USA
Slavik Pavel	Czech Technical University in Prague, Czech Republic
Sourin Alexei	Nanyang Technological University, Singapore
Stamminger Marc	REVES/INRIA, France
Srikanth Manohar	Indian Institute of Science, India
Staad Oliver	University of Rostock, Germany
Swan Ed	Mississippi State University, USA
Stefani Oliver	COAT-Basel, Switzerland
Sun Hanqiu	The Chinese University of Hong Kong, Hong Kong
Varsamidis Thomas	Bangor University, UK
Vercher Jean-Louis	Université de la Méditerranée, France
Wald Ingo	University of Utah, USA

Yu Ka Chun	Denver Museum of Nature and Science, USA
Yuan Chunrong	University of Tuebingen, Germany
Zachmann Gabriel	Clausthal University, Germany
Zara Jiri	Czech Technical University in Prague, Czech Republic
Zhang Hui	Indiana University, USA
Zhao Ye	Kent State University, USA
Zyda Michael	University of Southern California, USA

(Area 4) Visualization

Andrienko Gennady	Fraunhofer Institute IAIS, Germany
Apperley Mark	University of Waikato, New Zealand
Balázs Csébfalvi	Budapest University of Technology and Economics, Hungary
Bartoli Anna Vilanova	Eindhoven University of Technology, The Netherlands
Brady Rachael	Duke University, USA
Benes Bedrich	Purdue University, USA
Bilalis Nicholas	Technical University of Crete, Greece
Bonneau Georges-Pierre	Grenoble Université , France
Brown Ross	Queensland University of Technology, Australia
Bühler Katja	VRVIS, Austria
Callahan Steven	University of Utah, USA
Chen Jian	Brown University, USA
Chen Min	University of Wales Swansea, UK
Cheng Irene	University of Alberta, Canada
Chiang Yi-Jen	Polytechnic Institute of New York University, USA
Chourasia Amit	University of California - San Diego, USA
Coming Daniel	Desert Research Institute, USA
Dana Kristin	Rutgers University, USA
Dick Christian	Technical University of Munich, Germany
DiVerdi Stephen	Adobe, USA
Doleisch Helmut	VRVis Research Center, Austria
Duan Ye	University of Missouri-Columbia, USA
Dwyer Tim	Monash University, Australia
Ebert David	Purdue University, USA
Entezari Alireza	University of Florida, USA
Ertl Thomas	University of Stuttgart, Germany
Floriani Leila De	University of Maryland, USA
Fujishiro Issei	Keio University, Japan
Geist Robert	Clemson University, USA
Goebel Randy	University of Alberta, Canada

Gotz David	IBM, USA
Grinstein Georges	University of Massachusetts Lowell, USA
Goebel Randy	University of Alberta, Canada
Gregory Michelle	Pacific Northwest National Lab, USA
Hadwiger Helmut Markus	VRVis Research Center, Austria
Hagen Hans	Technical University of Kaiserslautern, Germany
Hamza-Lup Felix	Armstrong Atlantic State University, USA
Heer Jeffrey	Armstrong University of California at Berkeley, USA
Hege Hans-Christian	Zuse Institute Berlin, Germany
Hochheiser Harry	University of Pittsburgh, USA
Hollerer Tobias	University of California at Santa Barbara, USA
Hong Lichan	Palo Alto Research Center, USA
Hotz Ingrid	Zuse Institute Berlin, Germany
Jiang Ming	Lawrence Livermore National Laboratory, USA
Joshi Alark	Yale University, USA
Julier Simon J.	University College London, UK
Kohlhammer Jörn	Fraunhofer Institut, Germany
Kosara Robert	University of North Carolina at Charlotte, USA
Laramee Robert	Swansea University, UK
Lee Chang Ha	Chung-Ang University, Korea
Lewis Bob	Washington State University, USA
Liere Robert van	CWI, The Netherlands
Lim Ik Soo	Bangor University, UK
Linsen Lars	Jacobs University, Germany
Liu Zhanping	Kitware, Inc., USA
Ma Kwan-Liu	University of California-Davis, USA
Maeder Anthony	University of Western Sydney, Australia
Majumder Aditi	University of California, Irvine, USA
Malpica Jose	Alcala University, Spain
Masutani Yoshitaka	The University of Tokyo Hospital, Japan
Matkovic Kresimir	VRVis Forschungs-GmbH, Austria
McCaffrey James	Microsoft Research / Volt VTE, USA
McGraw Tim	West Virginia University, USA
Melançon Guy	CNRS UMR 5800 LaBRI and INRIA Bordeaux Sud-Ouest, France
Meyer Joerg	University of California, Irvine, USA
Miksch Silvia	Vienna University of Technology, Austria
Monroe Laura	Los Alamos National Labs, USA
Morie Jacki	University of Southern California, USA

Mueller Klaus	SUNY Stony Brook, USA
Museth Ken	Linköping University, Sweden
Paelke Volker	Leibniz Universität Hannover, Germany
Papka Michael	Argonne National Laboratory, USA
Pettifer Steve	The University of Manchester, UK
Pugmire Dave	Los Alamos National Lab, USA
Rabin Robert	University of Wisconsin at Madison, USA
Raffin Bruno	INRIA, France
Razdan Anshuman	Arizona State University, USA
Rhyne Theresa-Marie	North Carolina State University, USA
Rosenbaum Rene	University of California at Davis, USA
Santhanam Anand	MD Anderson Cancer Center Orlando, USA
Scheuermann Gerik	University of Leipzig, Germany
Shead Timothy	Sandia National Laboratories, USA
Shen Han-Wei	Ohio State University, USA
Silva Claudio	University of Utah, USA
Sips Mike	Stanford University, USA
Slavik Pavel	Czech Technical University in Prague, Czech Republic
Sourin Alexei	Nanyang Technological University, Singapore
Swan Ed	Mississippi State University, USA
Theisel Holger	University of Magdeburg, Germany
Thiele Olaf	University of Mannheim, Germany
Toledo de Rodrigo	Petrobras PUC-RIO, Brazil
Tricoche Xavier	Purdue University, USA
Umlauf Georg	HTWG Constance, Germany
Viegas Fernanda	IBM, USA
Wald Ingo	University of Utah, USA
Wan Ming	Boeing Phantom Works, USA
Weinkauf Tino	Courant Institute, New York University, USA
Weiskopf Daniel	University of Stuttgart, Germany
Wischgoll Thomas	Wright State University, USA
Wylie Brian	Sandia National Laboratory, USA
Yeasin Mohammed	Memphis University, USA
Yuan Xiaoru	Peking University, China
Zachmann Gabriel	Clausthal University, Germany
Zhang Eugene	Oregon State University, USA
Zhang Hui	Indiana University, USA
Zhao Ye	Kent State University, USA
Zhukov Leonid	Caltech, USA

ISVC 2010 Special Tracks

1. 3D Mapping, Modeling and Surface Reconstruction

Organizers

Nefian Ara	Carnegie Mellon University/NASA Ames Research Center, USA
Broxton Michael	Carnegie Mellon University/NASA Ames Research Center, USA
Huertas Andres	NASA Jet Propulsion Lab, USA

Program Committee

Hancher Matthew	NASA Ames Research Center, USA
Edwards Laurence	NASA Ames Research Center, USA
Bradski Garry	Willow Garage, USA
Zakhor Avidel	University of California at Berkeley, USA
Cavallaro Andrea	University Queen Mary, London, UK
Bouguet Jean-Yves	Google, USA

2. Best Practices in Teaching Visual Computing

Organizers

Albu Alexandra Branzan	University of Victoria, Canada
Bebis George	University of Nevada, Reno, USA

Program Committee

Bergevin Robert	University of Laval, Canada
Crawfis Roger	Ohio State University, USA
Hammoud Riad	DynaVox Systems, USA
Kakadiaris Ioannis	University of Houston, USA, USA
Laurendeau Denis	Laval University, Quebec, Canada
Maxwell Bruce	Colby College, USA
Stockman George	Michigan State University, USA

3. Low-Level Color Image Processing

Organizers

Celebi M. Emre	Louisiana State University, USA
Smolka Bogdan	Silesian University of Technology, Poland
Schaefer Gerald	Loughborough University, UK
Plataniotis Konstantinos	University of Toronto, Canada
Horiuchi Takahiko	Chiba University, Japan

Program Committee

Aygun Ramazan	University of Alabama in Huntsville, USA
Battiato Sebastiano	University of Catania, Italy
Hardeberg Jon	Gjøvik University College, Norway
Hwang Sae	University of Illinois at Springfield, USA
Kawulok Michael	Silesian University of Technology, Poland
Kockara Sinan	University of Central Arkansas, USA
Kotera Hiroaki	Kotera Imaging Laboratory, Japan
Lee JeongKyu	University of Bridgeport, USA
Lezoray Olivier	University of Caen, France
Mete Mutlu	Texas A&M University - Commerce, USA
Susstrunk Sabine	Swiss Federal Institute of Technology in Lausanne, Switzerland
Tavares Joao	University of Porto, Portugal
Tian Gui Yun	Newcastle University, UK
Wen Quan	University of Electronic Science and Technology of China, China
Zhou Huiyu	Queen's University Belfast, UK

4. Low Cost Virtual Reality: Expanding Horizons

Organizers

Sherman Bill	Indiana University, USA
Wernert Eric	Indiana University, USA

Program Committee

Coming Daniel	Desert Research Institute, USA
Craig Alan	University of Illinois/NCSA, USA
Keefe Daniel	University of Minnesota, USA
Kreylos Oliver	University of California at Davis, USA
O'Leary Patrick	Idaho National Laboratory, USA
Smith Randy	Oakland University, USA
Su Simon	Princeton University, USA
Will Jeffrey	Valparaiso University, USA

5. Computational Bioimaging

Organizers

Tavares João Manuel R. S.	University of Porto, Portugal
Jorge Renato Natal	University of Porto, Portugal
Cunha Alexandre	Caltech, USA

Program Committee

Santis De Alberto	Università degli Studi di Roma “La Sapienza”, Italy
Reis Ana Mafalda	Instituto de Ciencias Biomedicas Abel Salazar, Portugal
Barrutia Arrate Muñoz	University of Navarra, Spain
Calvo Begoña	University of Zaragoza, Spain
Constantinou Christons	Stanford University, USA
Iacoviello Daniela	Università degli Studi di Roma “La Sapienza”, Italy
Ushizima Daniela	Lawrence Berkeley National Lab, USA
Ziou Djemel	University of Sherbrooke, Canada
Pires Eduardo Borges	Instituto Superior Tecnico, Portugal
Sgallari Fiorella	University of Bologna, Italy
Perales Francisco	Balearic Islands University, Spain
Qiu Guoping	University of Nottingham, UK
Hanchuan Peng	Howard Hughes Medical Institute, USA
Pistori Hemerson	Dom Bosco Catholic University, Brazil
Yanovsky Igor	Jet Propulsion Laboratory, USA
Corso Jason	SUNY at Buffalo, USA
Maldonado Javier Melenchón	Open University of Catalonia, Spain
Marques Jorge S.	Instituto Superior Tecnico, Portugal
Aznar Jose M. García	University of Zaragoza, Spain
Vese Luminita	University of California at Los Angeles, USA
Reis Luís Paulo	University of Porto, Portugal
Thiriet Marc	Universite Pierre et Marie Curie (Paris VI), France
Mahmoud El-Sakka	The University of Western Ontario London, Canada
Hidalgo Manuel González	Balearic Islands University, Spain
Gurcan Metin N.	Ohio State University, USA
Dubois Patrick	Institut de Technologie Médicale, France
Barneva Reneta P.	State University of New York, USA
Bellotti Roberto	University of Bari, Italy
Tangaro Sabina	University of Bari, Italy
Silva Susana Branco	University of Lisbon, Portugal
Brimkov Valentin	State University of New York, USA
Zhan Yongjie	Carnegie Mellon University, USA

6. Unconstrained Biometrics: Advances and Trends**Organizers**

Proença Hugo	University of Beira Interior, Portugal
Du Yingzi	Indiana University-Purdue University Indianapolis, USA

XXIV Organization

Scharcanski Jacob	Federal University of Rio Grande do Sul Porto Alegre, Brazil
Ross Arun	West Virginia University, USA
Amayeh Gholamreza	EyeCom Corporation, USA

Program Committee

Júnior Adalberto Schuck	Federal University of Rio Grande do Sul, Brazil
Kwolek Bogdan	Rzeszow University of Technology, Poland
Jung Cláudio R.	Federal University of Rio Grande do Sul, Brazil
Alirezaie Javad	Ryerson University, Canada
Konrad Janusz	Boston University, USA
Kevin Jia	International Game Technologies, USA
Meyer Joceli	Federal University of Santa Catarina, Brazil
Alexandre Luís A.	University of Beira Interior, Portugal
Soares Luis	ISCTE, Portugal
Coimbra Miguel	University of Porto, Portugal
Fieguth Paul	University of Waterloo, Canada
Xiao Qinghan	Defense Research and Development Canada, Canada
Ives Robert	United States Naval Academy, USA
Tamir Samir	Ingersoll Rand Security, USA

7. Behavior Detection and Modeling

Organizers

Miller Ron	Wright-Patterson Air Force Base, USA
Bebis George	University of Nevada, USA
Rosen Julie	Science Applications International Corporation, USA
Davis Jim	Ohio State University, USA
Lee Simon	Army Research Laboratory, USA
Zandipour Majid	BAE Systems, USA

Organizing Institutions and Sponsors



Table of Contents – Part II

Calibration, Pose Estimation, and Reconstruction

Multiple Camera Self-calibration and 3D Reconstruction Using Pedestrians	1
<i>Michael Hödlmoser and Martin Kampel</i>	
Robust Radial Distortion from a Single Image	11
<i>Faisal Bukhari and Matthew N. Dailey</i>	
Projective Reconstruction of General 3D Planar Curves from Uncalibrated Cameras	21
<i>X.B. Zhang, A.W.K. Tang, and Y.S. Hung</i>	
A Novel Photometric Method for Real-Time 3D Reconstruction of Fingerprint	31
<i>Wuyuan Xie, Zhan Song, and Xiaolong Zhang</i>	
3D Camera Pose Estimation Using Line Correspondences and 1D Homographies	41
<i>Irene Reisner-Kollmann, Andreas Reichinger, and Werner Purgathofer</i>	
Near-Optimal Selection of Views and Surface Regions for ICP Pose Estimation	53
<i>L.H. Mark, G. Okouneva, P. Saint-Cyr, D. Ignakov, and C. English</i>	

Segmentation

Region and Edge-Adaptive Sampling and Boundary Completion for Segmentation	64
<i>Scott E. Dillard, Lakshman Prasad, and Jacopo Grazzini</i>	
Universal Seed Skin Segmentation	75
<i>Rehanullah Khan, Allan Hanbury, and Julian Stöttinger</i>	
A Sharp Concentration-Based Adaptive Segmentation Algorithm	85
<i>Christophe Fiorio and Andre Mas</i>	
Segmentation for Hyperspectral Images with Priors	97
<i>Jian Ye, Todd Wittman, Xavier Bresson, and Stanley Osher</i>	
The Curve Filter Transform – A Robust Method for Curve Enhancement	107
<i>Kristian Sandberg</i>	

Split Bregman Method for Minimization of Region-Scalable Fitting Energy for Image Segmentation	117
<i>Yunyun Yang, Chunming Li, Chiu-Yen Kao, and Stanley Osher</i>	

Stereo

A Correlation-Based Approach for Real-Time Stereo Matching	129
<i>Raj Kumar Gupta and Siu-Yeung Cho</i>	
Photometric Stereo under Low Frequency Environment Illumination	139
<i>Rui Huang and William A.P. Smith</i>	
Simultaneous Vanishing Point Detection and Camera Calibration from Single Images	151
<i>Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha</i>	
Inferring Planar Patch Equations from Sparse View Stereo Images	161
<i>Rimon Elias</i>	
Single Camera Stereo System Using Prism and Mirrors	170
<i>Gowri Somanath, Rohith MV, and Chandra Kambhamettu</i>	
A Region-Based Randomized Voting Scheme for Stereo Matching	182
<i>Guillaume Gales, Alain Crouzil, and Sylvie Chambon</i>	

Virtual Reality II

Adaptive Neighbor Pairing for Smoothed Particle Hydrodynamics	192
<i>Brandon Pelfrey and Donald House</i>	
System Structures for Efficient Rendering in Virtual Worlds and Virtual Testbeds	202
<i>Jürgen Rossmann and Nico Hempe</i>	
Prismfields: A Framework for Interactive Modeling of Three Dimensional Caves	213
<i>Matt Boggus and Roger Crawfis</i>	
Efficient Marker Matching Using Pair-Wise Constraints in Physical Therapy	222
<i>Gregory Johnson, Nianhua Xie, Jill Slaboda, Y. Justin Shi, Emily Keshner, and Haibin Ling</i>	
Learning and Prediction of Soft Object Deformation Using Visual Analysis of Robot Interactions	232
<i>Ana-Maria Cretu, Pierre Payeur, and Emil M. Petriu</i>	

Registration

A Novel Consistency Regularizer for Meshless Nonrigid Image Registration	242
<i>Wei Liu and Eraldo Ribeiro</i>	
Robust Rigid Shape Registration Method Using a Level Set Formulation	252
<i>Muayed S. Al-Huseiny, Sasan Mahmoodi, and Mark S. Nixon</i>	
A Meshless Method for Variational Nonrigid 2-D Shape Registration . . .	262
<i>Wei Liu and Eraldo Ribeiro</i>	
A New Simple Method to Stitch Images with Lens Distortion	273
<i>Myung-Ho Ju and Hang-Bong Kang</i>	
Robust Mosaicking of Stereo Digital Elevation Models from the Ames Stereo Pipeline	283
<i>Taemin Kim, Zachary Moratto, and Ara V. Nefian</i>	

Medical Imaging

Tissue Fate Prediction in Acute Ischemic Stroke Using Cuboid Models	292
<i>Fabien Scalzo, Qing Hao, Jeffrey R. Alger, Xiao Hu, and David S. Liebeskind</i>	
3D Vector Flow Guided Segmentation of Airway Wall in MSCT	302
<i>Margarete Ortner, Catalin Fetita, Pierre-Yves Brillet, Françoise Prêteux, and Philippe Grenier</i>	
Graph-Based Segmentation of Lymph Nodes in CT Data	312
<i>Yao Wang and Reinhard Beichel</i>	
Electron Microscopy Image Segmentation with Graph Cuts Utilizing Estimated Symmetric Three-Dimensional Shape Prior	322
<i>Huei-Fang Yang and Yoonsuck Choe</i>	
Retinal Vessel Extraction with the Image Ray Transform	332
<i>Alastair H. Cummings and Mark S. Nixon</i>	
Automatic Liver Segmentation from CT Scans Using Multi-layer Segmentation and Principal Component Analysis	342
<i>Hossein Badakhshannoory and Parvaneh Saeedi</i>	

ST: Low Cost Virtual Reality: Expanding Horizons

Low Cost VR Meets Low Cost Multi-touch	351
<i>Dane Coffey, Fedor Korsakov, and Daniel F. Keefe</i>	

IQ-Station: A Low Cost Portable Immersive Environment	361
<i>William R. Sherman, Patrick O’Leary, Eric T. Whiting, Shane Grover, and Eric A. Wernert</i>	
A Fiducial-Based Tangible User Interface for White Matter Tractography	373
<i>Steven R. Gomez, Radu Jianu, and David H. Laidlaw</i>	
Immersive Molecular Visualization and Interactive Modeling with Commodity Hardware	382
<i>John E. Stone, Axel Kohlmeyer, Kirby L. Vandivort, and Klaus Schulten</i>	
ST: Best Practices in Teaching Visual Computing	
Multi-institutional Collaboration in Delivery of Team-Project-Based Computer Graphics Studio Courses	394
<i>Tim McLaughlin, B. Adán Peña, Todd A. Fechter, Anton Markus Pasing, Judith Reitz, and Joseph A. Vidal</i>	
A Workflow Based Process Visual Analyzer (ProVisZer) for Teaching and Learning	406
<i>Nathaniel Rossol, Irene Cheng, and Mrinal Mandal</i>	
Teaching Geometric Modeling Algorithms and Data Structures through Laser Scanner Acquisition Pipeline	416
<i>S. Gueorguieva, R. Synave, and Ch. Couture-Veschambre</i>	
Creating Passion for Augmented Reality Applications – A Teaching Concept for a Lab Course	429
<i>Christian Waechter, Eva Artinger, Markus Duschl, and Gudrun Klinker</i>	
Applications	
Object Material Classification by Surface Reflection Analysis with a Time-of-Flight Range Sensor	439
<i>Md. Abdul Mannan, Dipankar Das, Yoshinori Kobayashi, and Yoshinori Kuno</i>	
Retrieving Images of Similar Geometrical Configuration	449
<i>Xiaolong Zhang and Baoxin Li</i>	
An Analysis-by-Synthesis Approach to Rope Condition Monitoring	459
<i>Esther-Sabrina Wacker and Joachim Denzler</i>	
Fast Parallel Model Estimation on the Cell Broadband Engine	469
<i>Ali Khalili, Amir Fijany, Fouzhan Hosseini, Saeed Safari, and Jean-Guy Fontaine</i>	

Organizing and Browsing Image Search Results Based on Conceptual and Visual Similarities	481
<i>Grant Strong, Enamul Hoque, Minglun Gong, and Orland Hoerber</i>	
Evaluation of a Difference of Gaussians Based Image Difference Metric in Relation to Perceived Compression Artifacts	491
<i>Gabriele Simone, Valentina Caracciolo, Marius Pedersen, and Faouzi Alaya Cheikh</i>	

Visualization II

Distance Field Illumination: A Rendering Method to Aid in Navigation of Virtual Environments	501
<i>Matt Boggus and Roger Crawfis</i>	
Indirect Shader Domain Rendering	511
<i>Daqing Xue and Roger Crawfis</i>	
Visual Exploration of Stream Pattern Changes Using a Data-Driven Framework	522
<i>Zaixian Xie, Matthew O. Ward, and Elke A. Rundensteiner</i>	
RibbonView: Interactive Context-Preserving Cutaways of Anatomical Surface Meshes	533
<i>T. McInerney and P. Crawford</i>	
Interactive Visualisation of Time-Based Vital Signs	545
<i>Rhys Tague, Anthony Maeder, and Quang Vinh Nguyen</i>	
Using R-Trees for Interactive Visualization of Large Multidimensional Datasets	554
<i>Alfredo Giménez, René Rosenbaum, Mario Hlawitschka, and Bernd Hamann</i>	
Combining Automated and Interactive Visual Analysis of Biomechanical Motion Data	564
<i>Scott Spurlock, Remco Chang, Xiaoyu Wang, George Arceneaux IV, Daniel F. Keefe, and Richard Souvenir</i>	

Video Analysis and Event Recognition

Human Activity Recognition: A Scheme Using Multiple Cues	574
<i>Samy Sadek, Ayoub Al-Hamadi, Bernd Michaelis, and Usama Sayed</i>	
A Platform for Monitoring Aspects of Human Presence in Real-Time . . .	584
<i>X. Zabulis, T. Sarmis, K. Tzevanidis, P. Koutlemanis, D. Grammenos, and A.A. Argyros</i>	

Egocentric Visual Event Classification with Location-Based Priors	596
<i>Sudeep Sundaram and Walterio W. Mayol-Cuevas</i>	
View Invariant Activity Recognition with Manifold Learning	606
<i>Sherif Azary and Andreas Savakis</i>	
Arm-Hand Behaviours Modelling: From Attention to Imitation	616
<i>Sean R.F. Fanello, Ilaria Gori, and Fiora Pirri</i>	
Hand Detection and Gesture Recognition Exploit Motion Times Image in Complicate Scenarios	628
<i>Zhan Song, Hanxuan Yang, Yanguo Zhao, and Feng Zheng</i>	
Face Verification Using Indirect Neighbourhood Components Analysis	637
<i>Hieu V. Nguyen and Li Bai</i>	
Poster Session	
Efficient Algorithms for Image and High Dimensional Data Processing Using Eikonal Equation on Graphs	647
<i>Xavier Desquesnes, Abderrahim Elmoataz, Olivier L��zoray, and Vinh-Thong Ta</i>	
3D DCT Based Compression Method for Integral Images	659
<i>Ju-Il Jeon and Hyun-Soo Kang</i>	
Plant Texture Classification Using Gabor Co-occurrences	669
<i>James S. Cope, Paolo Remagnino, Sarah Barman, and Paul Wilkin</i>	
A Compressive Sensing Algorithm for Many-Core Architectures	678
<i>A. Borghi, J. Darbon, S. Peyronnet, T.F. Chan, and S. Osher</i>	
An Incremental PCA-HOG Descriptor for Robust Visual Hand Tracking	687
<i>Hanxuan Yang, Zhan Song, and Runen Chen</i>	
Probabilistic Learning of Visual Object Composition from Attended Segments	696
<i>Masayasu Atsumi</i>	
Propagating Uncertainty in Petri Nets for Activity Recognition	706
<i>Gal Lavee, Michael Rudzsky, and Ehud Rivlin</i>	
Mixture of Gaussians Exploiting Histograms of Oriented Gradients for Background Subtraction	716
<i>Tomas Fabian</i>	

Human Pose Recognition Using Chamfer Distance in Reduced Background Edge for Human-Robot Interaction.....	726
<i>Anjin Park and Keechul Jung</i>	
Modeling Clinical Tumors to Create Reference Data for Tumor Volume Measurement.....	736
<i>Adele P. Peskin and Alden A. Dima</i>	
Spectral Image Decolorization	747
<i>Ye Zhao and Zakiya Tamimi</i>	
Author Index	757

Multiple Camera Self-calibration and 3D Reconstruction Using Pedestrians

Michael Hödlmoser and Martin Kampel

Computer Vision Lab, Vienna University of Technology
Favoritenstraße 9/183, 1040 Vienna

Abstract. The analysis of human motion is an important task in various surveillance applications. Getting 3D information through a calibrated system might enhance the benefits of such analysis. This paper presents a novel technique to automatically recover both intrinsic and extrinsic parameters for each surveillance camera within a camera network by only using a walking human. The same feature points of a pedestrian are taken to calculate each camera's intrinsic parameters and to determine the relative orientations of multiple cameras within a network as well as the absolute positions within a common coordinate system. Experimental results, showing the accuracy and the practicability, are presented at the end of the paper.

1 Introduction

According to [1], the task of calibrating a camera can mainly be divided into two approaches, namely calibration using a known calibration object and self-calibration. When the dimensions of an object are known, the 3D informations of a scene can be extracted by establishing correspondences between different views showing the same calibration object. In case of self-calibration, the dimensions of the calibration object are not known and geometric features (e.g. vanishing points (VPs)) are taken for camera calibration.

Surveillance scenarios might deal with the analysis of humans. Therefore, pedestrians are predestined to be used as calibration object for self-calibrating a camera. Calibrating a camera using a pedestrian was first introduced by Lv et. al. in [2]. Top and bottom points are determined and three VPs are extracted. A closed-form geometric solution is used to obtain the intrinsic parameters afterwards. The extrinsic parameters are only determined for one camera.

In this paper we present a novel and practical approach to determine both intrinsic and extrinsic parameters for a network of surveillance cameras by analyzing a pedestrian. We develop a 3D scene reconstruction method based on pairwise determination of relative orientations. Different to [3], the calibration method is replaced by a self-calibration approach similar to the one presented in [2]. We use top and bottom points for calculating three VPs, but instead of using a geometric closed form solution, the intrinsic parameters are then extracted from the image of the absolute conic (IAC), which should lead to faster computational time. In a next step, the extracted top and bottom

points are also used as input for gathering the relative orientations between the cameras within a network. To extract the scaling factor for the absolute translation of all cameras, the user needs to predefine the height of the walking person.

The paper is organized as follows: Section 2 gives an overview on related work concerning self-calibration and scene reconstruction using VPs and pedestrians. Section 3 explains our approach of calculating intrinsic and extrinsic parameters for each camera using a walking human. Experiments using synthetic and real world data are carried out in Section 4.

2 Related Work

Camera self-calibration and 3D reconstruction have been studied extensively in the last few years. Beardsley first described the extraction of intrinsic camera parameters from three VPs in [4]. By the determination of three VPs within one image, the principal point and the focal length can be recovered sequentially. In [5], camera calibration using three VPs of an image is described. Their semi-automatic auto-calibration method uses building facades to determine three VPs. The user needs to select a set of parallel image lines in order to search for correct VP initialization. After initialization, the intrinsic parameters are recovered. The relative rotation between a camera pair is estimated using the calculated points on the plane at infinity, the translation is calculated by using further points of interest in a scene. Based on the work of [2], Junejo presents a quite similar calibration approach for pedestrians walking on uneven terrains in [6]. The VPs do not need to be orthogonal to each other and the intrinsic parameters are estimated by obtaining the infinite homography from all VPs. A direct method for self-calibrating a camera by observing a pedestrian is presented in [7]. Based on top and bottom points of a walking human, the pose matrix is estimated column by column by exploiting cross-ratio constraints. Having two VPs and a vanishing line (VL), the third VP can be calculated. The VPs are then directly used to calculate the pose of the camera.

A scene reconstruction method for two cameras only is presented in [8]. The recover of the intrinsic parameters is based on the algorithm presented in [6]. The relative orientation is afterwards calculated using all VPs and the infinite homography. A pose estimation supporting multiple cameras is presented in [3]. A framework for manual camera calibration and 3D scene reconstruction for teleimmersion purposes using corresponding points of interest is proposed. The determination of intrinsic parameters is performed using Bouguet's Camera Calibration Toolbox for Matlab¹. For reconstructing the scene, two LED markers having a fixed distance and defining a virtual calibration object are used. After the marker is detected on the image plane, the fundamental matrix between two cameras is determined and the relative rotation is calculated.

¹http://www.vision.caltech.edu/bouguetj/calib_doc/,
last retrieved on 07.07.2010.

3 Calibration and 3D Reconstruction from Pedestrians

Calibrating a camera network consists of three steps, namely the determination of intrinsic parameters (focal length f , principal point pp) for each camera, the extraction of pairwise relative orientations ($\Delta R, \Delta t$) between two cameras and the determination of the absolute orientations (R, t) of all cameras in one common coordinate system.

This section describes the extraction of intrinsic and extrinsic parameters from a sequence of a walking human. The pedestrian is observed for at least 2 frames over time and top and bottom points are extracted. These points are taken to calculate three VPs. The intrinsic parameters are then determined from the IAC. The same points of the walking pedestrian are also used to recover the cameras' extrinsic parameters. To extract the scaling factor for the absolute translation of all cameras, the user needs to predefine the height of the walking person.

3.1 Estimation of VPs

By using the extracted top and bottom points, two VPs and a VL can be calculated. Since the third VP can be determined, the IAC can be formed using all three VPs. By applying the Cholesky decomposition [9], the intrinsic parameters can be extracted from the IAC. There are only two constraints related to this approach namely (a) the top and the bottom points need to have a fixed distance in 3D coordinates over time and (b) all top points and all bottom points are coplanar respectively. Both constraints are fulfilled by a pedestrian having a constant height walking on an even ground plane.

Determination of Top and Bottom Points: To determine top points ($T = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$) and bottom points ($B = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$) of a walking human, the human silhouette must be extracted in a first step. This step is crucial for all further calculations as an error of 1-2 pixels in top and bottom locations can result in wrong VP estimations and therefore wrong calibration results (see Section 4). A simple background subtraction is done to determine a rough silhouette of the pedestrian. The resulting foreground bounding box is used as input for the grab cut algorithm [10]. This enables the accurate extraction of the pedestrian from the background. The bounding box for the accurate determined pedestrian is calculated and divided in an upper and a lower part. The center of mass is calculated for the whole bounding box, for the lower and for the upper part respectively. By minimizing the least squared distance to all three center points, a line is fitted through the centers of mass, which gives a vertical VL of the walking human. The top and bottom points are then given by the intersection of the vertical VL and the bounding box. Figure 1(a) shows a random input image from a video sequence used for the experiments in Section 4. Figure 1(b) shows the extracted walking person within a rectangular bounding box and the top and bottom points. Figure 1(c) shows all pedestrian locations and the corresponding top and bottom points, represented by dots, within one frame.

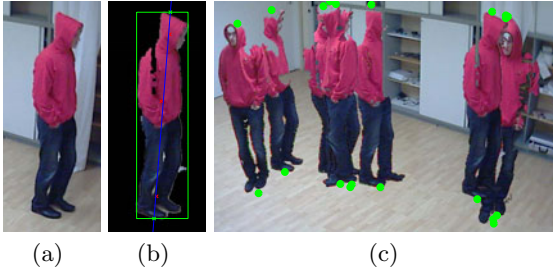


Fig. 1. (a) Example input image, (b) extraction of top and bottom points and (c) all extracted top and bottom locations

Approximation of the Vertical VP: According to [2], vertical VPs vz_{ij} can be calculated by using the top and bottom points of a walking human. Therefore, the human must be shown in a frame sequence consisting of at least two frames, i and j , where $i \neq j$. The vertical VPs can be expressed by $vz_{ij} = \overline{T_i B_i} \times \overline{T_j B_j}$.

In a next step, a general vertical VP, $vz = \begin{pmatrix} vz_x \\ vz_y \end{pmatrix}$, needs to be calculated. When using synthetic data, all vertical VPs must converge in one general VP. Since top and bottom location initialization may be noisy and therefore not all vertical VPs must converge to one point, an approximation needs to be performed. Using cross ratio in 2D, the relationship between the vertical VP, top and bottom points of one frame showing a pedestrian is given by [7]

$$vz_x(t_y - b_y) + vz_y(b_x - t_x) = (b_x t_y - t_x b_y). \quad (1)$$

When two or more instances of the pedestrian are given, an equation system $A \cdot vz = r$, where r is the result vector, can be formed to solve the equation. A least squares solution can be gathered by solving

$$vz = (A^T A)^{-1} A^T r. \quad (2)$$

To make the solution more robust, all $\binom{n}{2}$ possible combinations for n frames are used for further calculations.

Determination of the Horizontal VL and the Two Missing VPs: The horizontal VPs can be calculated by $vx_{ij} = \overline{T_i T_j} \times \overline{B_i B_j}$. When having synthetic data, all VPs must lie on one line called the VL. Since top and bottom location initialization may be noisy and therefore not all horizontal VPs must lie on one line, the line is fitted by minimizing the least squared distance to all horizontal VPs. Figure 2(a) shows the general vertical VP and a selection of horizontal VPs of four images of a walking human. As three VPs orthogonal to each other are needed for the intrinsic calibration of a camera, we can use vz and the VL for the determination of two more VPs, vx and vy . If the VL has a gradient equal to zero, it is called the horizontal line. The second VP vx can easily be

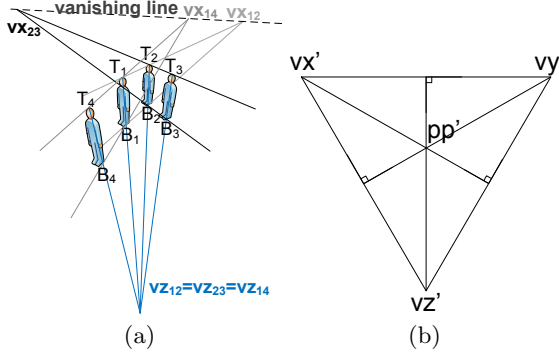


Fig. 2. Determination of (a) VL and a general vertical VP by using all horizontal and vertical VPs and (b) the calculation of two missing VPs by the triangle spanned by vx', vy' and vz'

determined by taking an arbitrary point on the VL. Before calculating vy , the VL needs to be aligned with the horizontal line. This is established by rotating the VL by the negative gradient of the VL. By aligning the VL, the whole scene, including all points needed for further calculations, must also be rotated ($vx = vx', vy = vy', vz = vz'$). The principal point pp can safely be assumed to be the image center and pp is rotated to obtain pp' . As pp' is the orthocenter of the triangle spanned by the three VPs, vy' can be calculated geometrically as shown in Figure 2(b) and described in [2]. To obtain vx, vy, vz , the scene is rotated back around the image origin by the gradient of the VL.

3.2 Retrieving the Camera Calibration Matrix

After the calculation of all three VPs, the IAC, ω , can be determined. Under the assumption of squared pixels and zero skew, ω has the form of

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_2 \\ 0 & \omega_1 & \omega_3 \\ \omega_2 & \omega_3 & \omega_4 \end{bmatrix} \quad (3)$$

According to [1], ω can be determined from three orthogonal VPs. The intrinsic parameters are directly related to ω by $(KK^T)^{-1} = \omega$. By applying the Cholesky decomposition [9], the intrinsic parameters can be extracted from the IAC.

3.3 Calibration of Camera Pairs

Top and bottom points of the walking pedestrian are also used to recover the scene structure, which in other words mean the determination of the cameras' extrinsic parameters. Before calculating an absolute orientation within a common world coordinate system, a relative orientation between each camera pair needs to be determined. The orientation between two arbitrary cameras consists of a relative rotation ΔR and a relative translation Δt . Two cameras are related by

the so called epipolar geometry, which can be described by a fundamental matrix [\[1\]](#). By gathering eight point correspondences between the cameras, the fundamental matrix can be calculated by the normalized 8-point algorithm, presented in [\[1\]](#).

When the intrinsic parameters are already known, the normalized corresponding points can be determined. Performing the 8-point algorithm by using normalized corresponding points leads to a resulting matrix called the essential matrix. The essential matrix is defined by

$$E = [\Delta t]_{\times} \Delta R = \Delta R [\Delta R^T \Delta t]_{\times} \quad (4)$$

The relative rotation and translation can directly be extracted from the essential matrix. As there are four solutions for ΔR and Δt , the solution where a positive depth is gathered for any projected point must be picked, as described in [\[1\]](#).

3.4 3D Reconstruction

After pairwise calibration of the camera network, the absolute orientation for each camera must be determined. Therefore, we determine one camera to be the common world coordinate origin. Having two cameras a, b and their relative orientation, the absolute rotation R_b and absolute translation t_b of camera b is calculated by

$$R_b = \Delta R R_a, \quad t_b = \Delta t + \Delta R t_a \quad (5)$$

Since the essential matrix is only determined up to a scale factor λ , this factor must be calculated to gain a scaled translation vector. Therefore, 3D points are calculated in the normalized world coordinate system using triangulation [\[1\]](#). The scale factor can be calculated by

$$\lambda = \frac{1}{N} \sum_{i=1}^N \frac{H}{\text{distance}(T3D_i, B3D_i)}, \quad (6)$$

where $T3D$ and $B3D$ are the top and bottom points in the normalized 3D space, N is the number of all point pairs respectively, and H is the real height of the pedestrian, which must be provided by the user. The rotation and translation parameters are optimized by running the sparse bundle adjustment algorithm over all cameras within the network, as proposed in [\[11\]](#). This algorithm simultaneously minimizes the reprojection error (RE) and the error of all absolute orientations, when intrinsic parameters are fixed.

4 Experiments

To show the accuracy of our proposed method, we do an evaluation using synthetic and real data. As we want to show that our algorithm is working precisely, we first evaluate it by using a synthetically generated scene. Additionally, the impact of the incorrect extraction of top and bottom points on the calibration

results is demonstrated by introducing noise to the input points. As we also want to show the practical use of the method in real world scenarios, we secondly evaluate it by using real world data. REs and comparisons of the results of our method to calculated ground truth data (synthetic scene) or results of conventional calibration methods (real world scene) are introduced as evaluation metrics.

Synthetic Data: For evaluation purposes, a 3D scene is generated using OpenGL and ten top and bottom points are randomly positioned within the scene. The distance between top and bottom locations is 56 millimeter. Ten cameras are located randomly within the observed scene of 360 degrees where all top and bottom points can be seen from each camera. All cameras have the same focal length and principal point. The positions of the cameras and the calculated camera positions are shown in Table III. Since OpenGL is using a different coordinate system than Matlab does and the coordinate origin is a different one than our implementation uses, we use the relative distance instead of a translation vector for comparisons. ΔR and Δt are the relative parameters for each camera to reference camera one. As can be seen, both the intrinsic and the extrinsic parameters can be extracted precisely. Reference values (RV) from the OpenGL scene are compared to calculated values (CV). All cameras offer a difference between calculated and reference focal length of $< 0.72\%$. The maximum difference of the relative translation is 0.03% , measured at camera seven. Since the top and bottom locations are calculated having no noise, it can be seen that the mean and the standard deviation of the RE (represented by MRE and SRE) is nearly zero. In a next step, we calculate the height of the synthetically generated distance between top and bottom locations. The mean height

Table 1. Comparison of calculated intrinsic / extrinsic parameters and their reference parameters of seven synthetic cameras. Additionally, REs are shown.

Cam #		f (pixel)	u_0 (pixel)	v_0 (pixel)	ΔR (degree)	Δt (mm)	MRE (10^{-3} pixel)	SRE (10^{-3} pixel)
01	CV	659.4054	320.0011	240.0001	(00.0000, 00.0000, 00.0000)	000.0000	0.0040	0.1928
	RV	660.0000	320.0000	240.0000	(00.0000, 00.0000, 00.0000)	000.0000	-	-
02	CV	659.4569	319.9791	239.9978	(60.0023, 00.0002, 00.0000)	371.3528	0.0040	0.2491
	RV	660.0000	320.0000	240.0000	(60.0000, 00.0000, 00.0000)	371.3080	-	-
03	CV	659.4182	319.9973	239.9998	(110.0048, 00.0010,-00.0005)	666.6406	0.0080	0.4019
	RV	660.0000	320.0000	240.0000	(110.0000, 00.0000, 00.0000)	666.5556	-	-
04	CV	659.7868	319.9553	239.9964	(135.0026,-00.0002, 00.0001)	782.5567	0.0060	0.2440
	RV	660.0000	320.0000	240.0000	(135.0000, 00.0000, 00.0000)	782.5418	-	-
05	CV	659.3608	319.9960	239.9997	(179.9988,-00.0002,-00.0001)	854.6013	0.0040	0.2986
	RV	660.0000	320.0000	240.0000	(180.0000, 00.0000, 00.0000)	854.5334	-	-
06	CV	659.4335	320.0258	240.0009	(224.9952, 00.0005, 00.0003)	786.9592	0.0050	0.2789
	RV	660.0000	320.0000	240.0000	(225.0000, 00.0000, 00.0000)	786.9350	-	-
07	CV	659.3384	319.9618	239.9895	(275.0126,-00.0026, 00.0002)	603.9850	0.0230	0.9620
	RV	660.0000	320.0000	240.0000	(275.0000, 00.0000, 00.0000)	603.7959	-	-
08	CV	659.3599	320.0348	240.0053	(290.0016,-00.0004,-00.0000)	492.7822	0.0040	0.2956
	RV	660.0000	320.0000	240.0000	(290.0000, 00.0000, 00.0000)	492.7633	-	-
09	CV	659.2860	319.9794	239.9975	(304.9972, 00.0003,-00.0004)	369.4673	0.0070	0.3678
	RV	660.0000	320.0000	240.0000	(305.0000, 00.0000, 00.0000)	369.4808	-	-
10	CV	659.3601	319.9663	239.9967	(339.9978, 00.0001,180.0000)	134.5118	0.0040	0.2187
	RV	660.0000	320.0000	240.0000	(340.0000, 00.0000,180.0000)	134.5223	-	-

of all ten top/bottom pairs is 56.00 millimeters, having a standard deviation of 2.0248e-004.

To proof the robustness of our algorithm, we run the method again by using a varying standard deviation on the top and bottom points. Table 2 shows the results obtained when using a raising standard deviation. As an example, we pick out the results of camera ten for calculations including noise. As can be seen, intrinsic parameters and the relative rotations remain stable (maximum $\Delta f = 8.71\%$, maximum $\Delta u_0 = 3.34\%$, maximum $\Delta v_0 = 0.60\%$, maximum $\Delta R = (0.07\%, 0.10\%, 3.34\%)$) whereas the distance between camera one and camera ten is getting smaller when noise is introduced (maximum $\Delta t = 26.17\%$).

Table 2. Intrinsic and relative orientation between reference camera one and camera ten when introducing noise

σ	$f(pixel)$	$u_0(pixel)$	$v_0(pixel)$	$\Delta R(degree)$	$\Delta t(mm)$
0.5	638.3583	323.4995	240.2694	(340.2192 -00.0780 179.7621)	123.3432
1.0	621.3275	322.8866	239.8885	(339.7636 00.3883 173.9788)	102.6989
1.5	602.4946	309.2980	238.5548	(340.5865 -00.5360 175.3318)	99.3099

The computation time of the whole calibration procedure using 2,3,4,5 and 10 cameras is 1.5, 2.7, 4.3, 5.6 and 13.8 seconds which means that by each camera used in the network, the computation time increases by approximately 1.4 seconds, which can be negligible in a calibration procedure.

Real Data: In this section we are using real data for evaluating our calibration method. This section should show the practical use of the approach. Basically, we divide our experiments in two different setups.

First three cameras are capturing a moving model figure (Figure 3(a)), having a height of 56 mm. Three Unibrain Fire-i webcams are used for this example as we want to show the remaining accuracy of the method using low cost cameras. The usage of three cameras also enables a observed scene of 360 degrees. This setup enables a more robust estimation of top and bottom locations since the figure is not really walking but only moving on a plane and therefore the two feet positions are always vertical.



Fig. 3. Input data for the evaluation setup using real data: (a) model figure and (b) walking human

Table 3. Intrinsic and extrinsic of three real cameras compared to the ground truth, given by the camera’s data sheets. Additionally, MRE and SRE are shown.

	$f(pixel)$	$u_0(pixel)$	$v_0(pixel)$	$\Delta R(degree)$			$\Delta t(mm)$	MRE(pixel)	SRE(pixel)
Model Figure									
GT	746.6666	320.0000	240.0000	-			-	-	-
cam01	766.0100	331.9616	239.3585	(00.0000	00.0000	00.0000)	00.0000	3.9084	2.4484
cam02	685.4828	336.4431	239.8932	(246.3855	17.7824	21.4211)	362.7431	2.9294	1.7585
cam03	709.1168	286.1695	243.2339	(135.3388	-12.9850	33.9369)	412.5273	3.8092	3.0101
Pedestrian									
GT	782.2222	320.0000	240.0000	-			-	-	-
cam01	781.8839	374.9783	243.3623	(00.0000	00.0000	00.0000)	0000.0000	3.8317	3.0453
cam02	783.5094	303.3124	241.6647	(266.5525	-36.3978	-18.9564)	4221.4870	3.8105	3.2615

Second two cameras (AXIS M1031-W), mounted on the wall of a living room, are capturing a walking human (Figure 3(b)) having the height of 195cm.

As we want to show that the proposed method is also working with a low number of input images, we use ten different moving positions of the model figure and ten walking positions of the pedestrian, respectively. Table 3 shows the calculated intrinsic parameters using the proposed method. The ground truth (denoted as GT in the table) for the intrinsic is given by the data sheet of the used cameras. Since the determination of top and bottom locations is noisy due to the nature of the person detector, it can be seen that the MRE is relatively high in both cases (e.g. 3.9084 and 3.8322 pixels in camera 1 for experiment 1 and 2, respectively). Due to manual measurement errors, it is impossible to recover the absolute translation and rotation precisely. To proof the accuracy of the proposed 3D reconstruction method on real data, we calculate the height of the model figure and the pedestrian. We are using all ten recovered top and bottom positions and calculate a mean height of all ten points, which is 56.00 millimeters (standard deviation = 01.2137 millimeters) for the model figure and 195.0005 centimeters (standard deviation = 4.1674 centimeters) for the pedestrian.

5 Conclusion

In surveillance scenarios, pedestrians are predestined to be used as input for self-calibrating a camera. We proposed a novel self-calibration and 3D reconstruction method to recover both intrinsic and extrinsic parameters of the devices within a camera network by only observing a pedestrian. The user needs to predefine the height of the walking person to determine the scaling of all cameras’ absolute translations. As shown in our experiments, the accuracy of the method does only depend on the noise of top and bottom locations. As we want to use this for surveillance applications where pedestrians need to be classified and heights must be measured, the method is working accurate enough, where measured 195 centimeters offer calibration results having a standard deviation of 4.1674 centimeters. The computation time increases by each camera added to the network by approximately 1.4 seconds.

Acknowledgement

This work was partly supported by the Austrian Research Promotion Agency (FFG) under KIRAS project miniSPOT.net.

References

1. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003)
2. Lv, F., Zhao, T., Nevatia, R.: Camera Calibration from Video of a Walking Human. *IEEE Trans. on PAMI* 28, 1513–1518 (2006)
3. Kurillo, G., Li, Z., Bajcsy, R.: Framework for hierarchical calibration of multi-camera systems for teleimmersion. In: *Proc. of the IMMERSCON 2009*, pp. 1–6 (2009)
4. Beardsley, P., Murray, D.: Camera calibration using vanishing points. In: *Proc. of the BMVC, Leeds, UK*, pp. 417–425 (1992)
5. Cipolla, R., Drummond, T., Robertson, D.P.: Camera Calibration from Vanishing Points in Image of Architectural Scenes. In: *Proc. of the BMVC, Nottingham, UK*, vol. 2, pp. 382–391 (1999)
6. Junejo, I.: Using Pedestrians Walking on Uneven Terrains for Camera Calibration. In: *MVA* (2009)
7. Kusakunniran, W., Li, H., Zhang, J.: A Direct Method to Self-Calibrate a Surveillance Camera by Observing a Walking Pedestrian. In: *Proc. of DICTA, Melbourne*, pp. 250–255 (2009)
8. Chen, T., Del Bimbo, A., Pernici, F., Serra, G.: Accurate self-calibration of two cameras by observations of a moving person on a ground plane. In: *Proc. of the AVSS, London, UK*, pp. 129–134. IEEE Computer Society, Los Alamitos (2007)
9. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge (1992)
10. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23, 309–314 (2004)
11. Lourakis, M.I.: levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++ (2004), <http://www.ics.forth.gr/~lourakis/levmar/>

Robust Radial Distortion from a Single Image

Faisal Bukhari and Matthew N. Dailey

Computer Science and Information Management

Asian Institute of Technology

Pathumthani, Thailand

syed.faisal.bukhari@ait.ac.th, mdailey@ait.ac.th

Abstract. Many computer vision algorithms rely on the assumption of the pinhole camera model, but lens distortion with off-the-shelf cameras is significant enough to violate this assumption. Many methods for radial distortion estimation have been proposed, but they all have limitations. Robust automatic radial distortion estimation from a single natural image would be extremely useful for some applications. We propose a new method for radial distortion estimation based on the plumb-line approach. The method works from a single image and does not require a special calibration pattern. It is based on Fitzgibbon's division model, robust estimation of circular arcs, and robust estimation of distortion parameters. In a series of experiments on synthetic and real images, we demonstrate the method's ability to accurately identify distortion parameters and remove radial distortion from images.

1 Introduction

Most computer vision algorithms, particularly structure from motion algorithms, rely on the assumption of a linear pinhole camera model. However, most commercially available cameras introduce sufficiently severe optical distortion that the pinhole assumption is invalid, making distortion correction a must.

Radial distortion is the most significant type of distortion in today's cameras [1,2]. It is most evident in images produced with low-cost, wide-angle lenses. Such lenses are being widely deployed, for example, in automotive applications such as assisting drivers to view a vehicle's blind spots [3,4]. But it is also significant enough in higher-quality cameras to introduce error into 3D reconstruction processes. Radial distortion bends straight lines into circular arcs [2,5], violating the main invariance preserved in the pinhole camera model, that straight lines in the world map to straight lines in the image plane [6,7]. Radial distortion may appear as barrel distortion, usually arising at short focal lengths, or pincushion distortion, usually arising at longer focal lengths.

Methods for radial distortion estimation fall into three major categories: point correspondence [8,19], multiple view autocalibration [10,11,12,13,14], and plumb-line. Plumb-line methods are the most promising for robust distortion

estimation from a single image or a small number of images. Rather than using a known pattern or sequence of images under camera motion, they estimate distortion parameters directly from distorted straight lines in one or more images. Straight lines are frequent enough in most human-made environments to make distortion estimation from a single image possible [25,15]. However, existing methods require human intervention [16,17,18], do not use all available lines for distortion estimation despite the fact that additional lines could minimize estimation error [15,25], or assume the distortion center as the center of the image [2,19], which is in contrast to recommendations [11,20]. The Devernay and Faugeras [6] method is the only existing method that overcomes these limitations. However, it requires a complex process of polygonal approximation of the distorted lines. As we shall see, the distorted line detection process can be dramatically simplified by using an alternative distortion model.

In this paper, we propose a new method based on the plumb-line approach that addresses these limitations. The method works from a single image if the image contains a sufficient number of distorted straight lines. It does not require a calibration pattern or human intervention. We use Fitzgibbon’s division model of radial distortion [12] with a single parameter. Our estimator is similar to that of Strand and Hayman [2] and Wang et al. [5] in that we estimate the parameters of the distortion model from the parameters of circular arcs identified in the distorted image, based on the fact that distorted straight lines can be modeled as circular under the single parameter division model [10]. Our contribution is to make the process fully automatic and robust to outliers using a two-step random sampling process. For the first step, we introduce a sampling algorithm to search the input image for subsequences of contours that can be modeled as circular arcs. For the second step, we introduce a sampling algorithm that finds the distortion parameters consistent with the largest number of arcs. Based on these parameters, we undistort the input image.

To evaluate the new algorithm, we perform a quantitative study of its performance on distorted synthetic images and provide an example of its ability to remove distortion from a real image. We find that the algorithm performs very well, with excellent reconstruction of the original image even under severe distortion, and that it is able to eliminate the visible distortion in real images.

2 Mathematical Model

In this section, we outline the mathematical model of radial distortion assumed in the rest of the paper and show how to estimate the parameters of this model.

2.1 Distortion Model

Although the most commonly used radial distortion model is the *even-order polynomial model*, we use Fitzgibbon’s *division model*, which is thought to be a more accurate approximation to the typical camera’s true distortion function:

$$x_u = \frac{x_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \dots} \quad y_u = \frac{y_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \dots} \quad .$$

(x_u, y_u) and (x_d, y_d) are the corresponding coordinates of an undistorted point and a distorted point, respectively. r_d is the Euclidean distance of the distorted point to the distortion center; if the distortion center is the origin of the distorted image, we can write $r_d^2 = x_d^2 + y_d^2$ or otherwise if (x_0, y_0) is the center, we write $r_d^2 = (x_d - x_0)^2 + (y_d - y_0)^2$. $\lambda_1, \lambda_2, \lambda_3, \dots$ are the distortion parameters, which must be estimated from image measurements. We use the single parameter division model (fixing $\lambda_2 = \lambda_3 = \dots = 0$), because for most cameras, a single term is sufficient [12,5].

2.2 Distortion of a Line under the Single-Parameter Division Model

Wang et al. [5] show that under the single-parameter division model, the distorted image of a straight line is a circular arc. However, they use the slope-y-intercept form of the equation of a line, which we avoid due to its inability to model vertical lines and its undesirable numerical properties [21]. It can be shown (details omitted) that the general line

$$ax_u + by_u + c = 0 \quad (1)$$

is imaged as a circular arc on the circle

$$x_d^2 + y_d^2 + \frac{a}{c\lambda}x_d + \frac{b}{c\lambda}y_d + \frac{1}{\lambda} = 0, \quad (2)$$

under the single parameter division model. It is also possible to come to the same conclusion using the parametric form of a straight line [2]. When the distortion model includes a center of distortion that is not the image center, we obtain a more complex equation that still defines a circle.

2.3 Estimating Distortion Parameters from Circular Arcs

Strand and Hayman [2] and Wang et al. [5] show that it is possible to estimate λ from the parameters of circular arcs identified in an image. However, Rezazadegan and Reza [20] have found that modeling the distortion center in addition to the radial distortion parameter(s) can increase the accuracy of the calibration process. Wang et al. [5] thus further show how both λ and the distortion center (if not assumed to be the center of the image) can be estimated from the parameters of three circular arcs identified in an image. We use their formulation. For each arc $i \in \{1, 2, 3\}$, we rewrite Equation 2 in the form $x_d^2 + y_d^2 + A_i x_d + B_i y_d + C_i = 0$. Then the distortion center can be found by solving the linear system

$$\begin{aligned} (A_1 - A_2)x_0 + (B_1 - B_2)y_0 + (C_1 - C_2) &= 0 \\ (A_1 - A_3)x_0 + (B_1 - B_3)y_0 + (C_1 - C_3) &= 0 \\ (A_2 - A_3)x_0 + (B_2 - B_3)y_0 + (C_2 - C_3) &= 0, \end{aligned} \quad (3)$$

and λ can be estimated from

$$\frac{1}{\lambda} = x_0^2 + y_0^2 + Ax_0 + By_0 + C, \quad (4)$$

using any of the three circular arcs' parameters in place of (A, B, C) . See Wang et al. [5] for details.

3 Robust Radial Distortion Estimation

In this section, we provide the details of our approach, which is based on robust estimation and the mathematical model introduced in Section 2.

3.1 Identifying Circular Arcs

The first step is to robustly identify as many circular arcs as possible in the image. Given an input image, we first extract Canny edges and link adjacent edge pixels into contours. We discard any contour whose length is below a threshold. For each remaining contour, we then attempt to find long pixel subsequences that can be fit by circular arcs. Our method is based on random sampling and inspired by RANSAC [22], but, rather than finding a single model for all the data, we preserve all models (candidate circular arcs) that are not overlapping with other arcs in the same contour that have more support. The termination criterion is to stop once the probability that an arc of minimal length has not yet been found is small. The detailed algorithm is presented in Section 3.5.

3.2 Refining Circular Arc Estimates

After the initial arc identification process is complete, each resulting arc, whose parameters have been calculated directly from the minimum sample of three points, is refined using the inlier pixel contour subsegment supporting that model. The gold standard objective function for circle fitting is

$$\Omega(x_c, y_c, r) = \sum_{i=1}^N d(x_i, y_i, x_c, y_c, r)^2, \quad (5)$$

where (x_c, y_c) is the center of the circle, r is its radius, and $d(x, y, x_c, y_c, r)$ is the orthogonal distance of the measured point (x, y) to the hypothetical circle. Since there is no closed-form solution minimizing this objective function [23], we use an initial guess and the Levenberg-Marquardt nonlinear least squares method to find a local minimum.

As the initial estimate of the circle's parameters, we use either the parameters calculated during the sampling procedure or Taubin's method [24], which is based on algebraic error minimization.

3.3 Estimating Distortion Parameters

Once we have obtained a set of circular arcs as candidate distorted straight lines, we use the estimator of Equations 3 and 4 and a standard RANSAC procedure to find a set of distortion parameters with maximal support. In the sampling loop, we sample three arcs, calculate the model, and count the number of arcs that are inliers by first undistorting them using the estimated distortion parameters then testing for straightness using orthogonal regression. The detailed algorithm is presented in Section 3.6.

Require: Contours C_1, C_2, \dots

Ensure: A is the output arc set

```

1:  $A \leftarrow \emptyset$ 
2: for each contour  $C_i$  do
3:   if  $|C_i| \geq l^{\min}$  then
4:      $N \leftarrow f(l^{\min}, |C_i|)$ 
5:     for  $n = 1$  to  $N$  do
6:       Sample three points  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  from  $C_i$ .
7:       if  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  are not collinear then
8:         Calculate  $x_c, y_c, r$  from  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ .
9:          $A^{\text{new}} \leftarrow$  arc for longest subsequence of  $C_i$  consistent with  $x_c, y_c, r$ 
10:        if  $|A^{\text{new}}| \geq l^{\min}$  then
11:          if  $A^{\text{new}}$  does not overlap with any arc in  $A$  then
12:             $A \leftarrow A \cup \{A^{\text{new}}\}$ 
13:          else if  $A^{\text{new}}$  is longer than every overlapping arc in  $A$  then
14:            Remove arcs overlapping with  $A^{\text{new}}$  from  $A$ 
15:             $A \leftarrow A \cup \{A^{\text{new}}\}$ 
16:          end if
17:        end if
18:      end if
19:    end for
20:  end if
21: end for

```

Algorithm 1. Robust arc identification

3.4 Undistortion

The last step in our procedure is to undistort the input image. We use the optimal distortion parameters and the inverse of the distortion model

$$x_d = x_0 + (1 + \lambda r_u^2)x_u \quad y_d = y_0 + (1 + \lambda r_u^2)y_u$$

with bilinear interpolation and appropriate translation and scale factors to produce the output undistorted image.

3.5 Robust Arc Identification Algorithm

In Algorithm 1, we provide the details of our sampling-based arc identification method. To determine the number of iterations required, the algorithm uses a function $f(l, n)$, which gives the number of trials required to ensure that the probability of not sampling three of l inliers from a set of n points is small. This ensures that we sample a sufficient number of times to find, with high probability, all arcs with sufficient length in each contour.

3.6 Robust Distortion Parameter Estimation Algorithm

In Algorithm 2, we describe our estimation procedure in detail. Once a set of candidate arcs has been identified per Algorithm 1, distortion parameter estimation is a straightforward application of RANSAC [22]. In the sampling loop,

we use adaptive calculation of the number of iterations required based on the number of inlier arcs [7]. The termination criterion uses the same function $f(l, n)$ to determine the number of trials required to ensure that the probability of not sampling three of l inliers from n items is small. An arc is judged to be an inlier if, after undistortion using the candidate distortion parameters λ, x_0 , and y_0 , the pixels of the arc form a straight line, as measured by orthogonal regression.

Require: Arc set A

Ensure: λ^*, x_0^*, y_0^* are the output distortion parameters

```

1:  $(\lambda^*, x_0^*, y_0^*) \leftarrow (\emptyset, \emptyset, \emptyset)$ 
2: if  $|A| \geq 3$  then
3:    $N \leftarrow 0$ 
4:    $s \leftarrow 0$ 
5:   loop
6:      $N \leftarrow N + 1$ 
7:     Sample three distinct arcs  $A_1, A_2, A_3$ 
8:     Estimate  $\lambda, x_0, y_0$  from  $A_1, A_2, A_3$  per Equations 3 and 4
9:     if support for  $(\lambda, x_0, y_0)$  is greater than  $s$  then
10:       $s \leftarrow$  support for  $(\lambda, x_0, y_0)$ 
11:       $(\lambda^*, x_0^*, y_0^*) \leftarrow (\lambda, x_0, y_0)$ 
12:     end if
13:     if  $N \geq f(s, |A|)$  then
14:       break
15:     end if
16:   end loop
17: end if

```

Algorithm 2. Robust distortion parameter estimation

4 Experimental Evaluation

In this section, we describe a detailed quantitative study of the performance of our method on synthetic images and show qualitative results with real images. A sample of the images we used with results is shown in Fig. 1. We used the same original image (Fig. 1(a)) for all experiments. In each experiment, we distort the original image using particular ground truth values of λ, x_0 , and y_0 (Fig. 1(b)), identify circular arcs in the image (Fig. 1(c)), estimate the distortion parameters, and use those parameters to undistort the image (Fig. 1(d)).

We describe two series of experiments with synthetic images. In both cases, we used OpenCV’s Canny and contour extraction algorithms with a low gradient threshold of 50 and a high gradient threshold of 150. We fixed the minimum contour length at 150 pixels. For each contour of sufficient length, our arc extraction procedure (Algorithm 1) pre-calculates the number N of point sampling steps to perform using assuming a minimum number $l^{\min} = 50$ of inlier pixels.

In a first series of runs, we varied λ while keeping the distortion center fixed at $(x_0, y_0) = (320, 240)$, the image center. In a second series of runs, we kept the distortion level fixed ($\lambda = -10^{-6}$) while varying the distortion center. In every

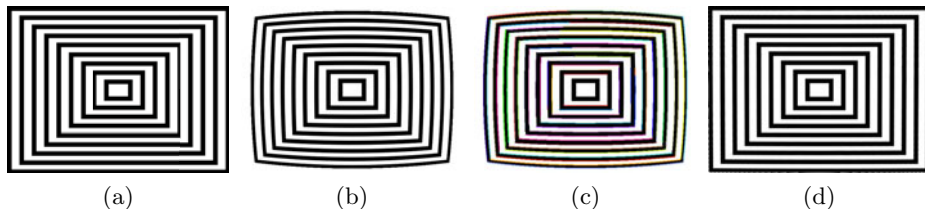


Fig. 1. Example experiment with synthetic image size 640×480 . (a) Original image. (b) Distorted image with $\lambda = -10^{-6}$, $(x_0, y_0) = (320, 240)$ (the image center). (c) Estimated arcs. (d) Undistorted image using estimated values of $\lambda = -9.8097^{-7}$, $x_0 = 319.632$, and $y_0 = 247.75$. Using true distortion parameters, $\text{RMSE} = 3.74103$ and using estimated parameters, $\text{RMSE} = 3.79212$.

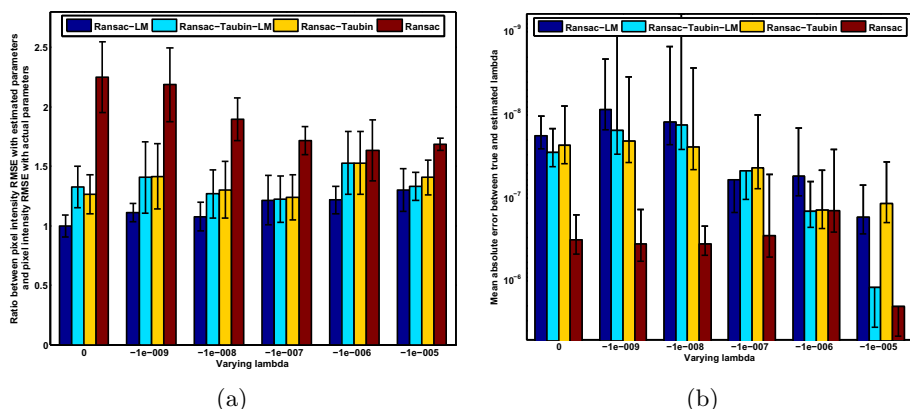


Fig. 2. Results of synthetic image experiments with varying λ . Distortion center is fixed at image center $(x_0, y_0) = (320, 240)$. (a) Noise in the undistorted image relative to the original image, measured by the ratio of the RMSE using estimated parameters to the RMSE using true parameters. (b) Error in estimating λ . Each point is an average over the same 10 runs shown in part (a). Each point is an average over 10 runs. Error bars denote 95% confidence intervals.

case, we estimated all three parameters of the distortion model. We compare four methods for arc estimation. The results for varying λ are shown in Fig. 2, and the results for varying distortion center are shown in Fig. 3. The “Ransac” method means we accept the circular arc model computed from three sample points, without any refinement after calculating the inliers. “Ransac-Taubin” is the result of using the Taubin method to refine the arc model computed from three sample points. “Ransac-LM” is the result of applying the Levenberg-Marquardt method directly to the model computed from three sample points. Under the hypothesis that starting LM from the sample-based estimate might not work as well as an initial estimate closer to the optimum, we also performed one series of experiments in which we first applied the Taubin method to the

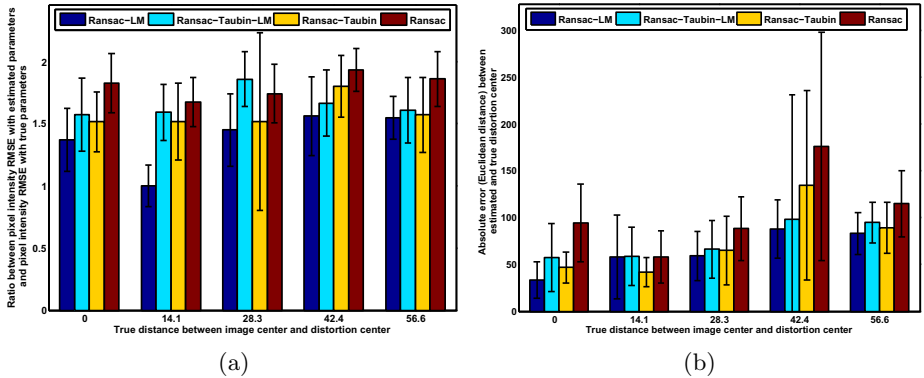


Fig. 3. Results of synthetic image experiments with varying distortion center. Distortion level is fixed at $\lambda = -10^{-6}$. (a) Noise in the undistorted image relative to the original image. (b) Error in estimating the distortion center. Each point is an average over 10 runs. Error bars denote 95% confidence intervals.

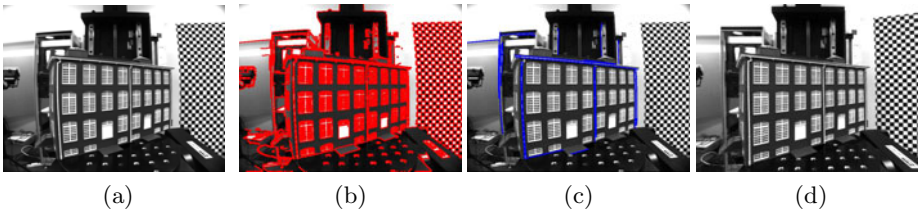


Fig. 4. Example results on real image. (a) Original image. (b) Extracted contours. (c) Identified arcs. (d) Undistorted image using parameters estimated via the “Ransac-LM” circle fitting method.

sample-based model then applied LM to the Taubin estimate. The results from this method are shown as “Ransac-Taubin-LM.”

Over the two series of runs, we observe variability between the actual and estimated parameter values with all of the circle fitting methods, but the performance of the method in terms of RMSE is quite good. The “Ransac-LM” method provides the most stable performance over different levels of distortion and distortion center parameters. Even in the case of severe barrel distortion ($\lambda = 10^{-5}$), the RMSE error introduced when undistorting using the parameters estimated by Ransac-LM is only about 30.06% more than that introduced when using the true distortion parameters.

Finally, in Fig. 4 we provide an example of the proposed method’s ability to identify distortion parameters and undistort a real image [25]. The robust arc selection and parameter estimation method is able to find a consensus set corresponding to distorted straight lines and is successful at removing most of the radial distortion from the image.

5 Conclusion

In this paper, we have introduced a new algorithm for radial distortion estimation and removal based on the plumb-line approach. The method works from a single image and does not require a special calibration pattern. It is based on Fitzgibbon's division model, robust estimation of circular arcs, and robust estimation of distortion parameters. In a series of experiments on synthetic and real images, we have demonstrated the method's ability to accurately identify distortion parameters and remove radial distortion from images.

The main limitation of the current implementation is that some parameters, especially the thresholds for Canny edge extraction, random sampling inlier calculations, and minimum contour length must be specified manually. In future work, we will improve the method to address these limitations.

Acknowledgments

Faisal Bukhari was supported by a graduate fellowship from the Higher Education Commission of Pakistan. We are grateful to Irshad Ali for his valuable feedback and comments.

References

1. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1330–1334 (2000)
2. Strand, R., Hayman, E.: Correcting radial distortion by circle fitting. In: *British Machine Vision Conference, BMVC* (2005)
3. Friel, M., Hughes, C., Denny, P., Jones, E., Glavin, M.: Automatic calibration of fish-eye cameras from automotive video sequences. *Intelligent Transport Systems, IET* 4, 136–148 (2010)
4. Hughes, C., Glavin, M., Jones, E., Denny, P.: Wide-angle camera technology for automotive applications: a review. *Intelligent Transport Systems, IET* 3, 19–31 (2009)
5. Wang, A., Qiu, T., Shao, L.: A simple method of radial distortion correction with centre of distortion estimation. *Journal of Mathematical Imaging and Vision* 35, 165–172 (2009)
6. Devernay, F., Faugeras, O.: Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications* 13, 14–24 (2001)
7. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
8. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Radiometry*, 221–244 (1992)
9. Braüer-Burchardt, C.: A simple new method for precise lens distortion correction of low cost camera systems. In: *German Pattern Recognition Symposium*, pp. 570–577 (2004)
10. Barreto, J.P., Daniilidis, K.: Fundamental matrix for cameras with radial distortion. In: *International Conference on Computer Vision (ICCV)*, pp. 625–632 (2005)

11. Hartley, R., Kang, S.: Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 1309–1321 (2007)
12. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 125–132 (2001)
13. Stein, G.P.: Lens distortion calibration using point correspondences. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 602–608 (1996)
14. Ramalingam, S., Sturm, P., Lodha, S.K.: Generic self-calibration of central cameras. *Computer Vision and Image Understanding* 114, 210–219 (2010)
15. Thormählen, T., Broszio, H., Wassermann, I.: Robust line-based calibration of lens distortion from a single view. In: *Computer Vision / Computer Graphics Collaboration for Model-based Imaging Rendering, Image Analysis and Graphical Special Effects*, pp. 105–112 (2003)
16. Brown, D.C.: Close-range camera calibration. *Photogrammetric Engineering* 37, 855–866 (1971)
17. Swaminathan, R., Nayar, S.: Non-Metric Calibration of Wide-Angle Lenses and Polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1172–1178 (2000)
18. Alvarez, L., Gómez, L., Sendra, J.R.: An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision* 35, 36–50 (2009)
19. Brauer-Burchardt, C., Voss, K.: A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In: *IEEE International Conference on Image Processing*, vol. 1, pp. 225–228 (2001)
20. Tavakoli, H.R., Pourreza, H.R.: Automated center of radial distortion estimation, using active targets. In: *Asian Conference on Computer Vision, ACCV* (2010)
21. Chernov, N.: *Circular and Linear Regression: Fitting Circles and Lines by Least Squares*. Chapman & Hall, Boca Raton (2010)
22. 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)
23. Al-Sharadqah, A., Chernov, N.: Error analysis for circle fitting algorithms. *The Electronic Journal of Statistics* 3, 886–911 (2009)
24. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 1115–1138 (1991)
25. Tomasi, C.: Sample image for CPS 296.1 homework assignment (2007), <http://www.cs.duke.edu/courses/spring06/cps296.1/homework/1/lab.gif>

Projective Reconstruction of General 3D Planar Curves from Uncalibrated Cameras

X.B. Zhang, A.W.K. Tang, and Y.S. Hung

Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong

Abstract. In this paper, we propose a new 3D reconstruction method for general 3D planar curves based on curve correspondences on two views. By fitting the measured and transferred points using spline curves and minimizing the 2D Euclidean distance from measured and transferred points to fitted curves, we obtained an optimum homography which relates the curves across two views. Once two or more homographies are computed, 3D projective reconstruction of those curves can be readily performed. The method offers the flexibility to reconstruct 3D planar curves without the need of point-to-point correspondences, and deals with curve occlusions automatically.

1 Introduction

Computer vision has been a hot topic in the past decades. While 3D reconstruction based on points or lines has been widely studied [1,2,3], 3D reconstruction methods based on curve correspondences have recently drawn the researchers' attention.

The existing literature of 3D reconstruction based on curve correspondences can be classified into three groups: conic reconstruction, high-order curve reconstruction and general 3D curve reconstruction. Among these three groups, the most restrictive one is based on projected conics [4,5,6,7] which, however, has attracted much more attention than the other groups.

Another group is high-order planar curve reconstruction provided by Kaminiski and Shashua. They suggested a closed form solution to the recovery of homography matrix from a single pair high-order curve matched across two views and the recovery of fundamental matrix from two pairs of high-order planar curves [8]. They have also extended this high-order method to general 3D curve reconstruction [9].

The last group is general 3D curve reconstruction from 2D images using affine shape. Berthilsson et al. [10] developed an affine shape method and employed it together with parametric curve for 3D curve reconstruction which demonstrated excellent reconstruction results. While this method is quite applicable, the minimized quantity is the subspace error, which lacks geometric meaning.

In this paper, we proposed a new method of general 3D planar curve reconstruction which solves for the homography between two views by minimizing the sum of squares of Euclidean distances from 2D measured and transferred points

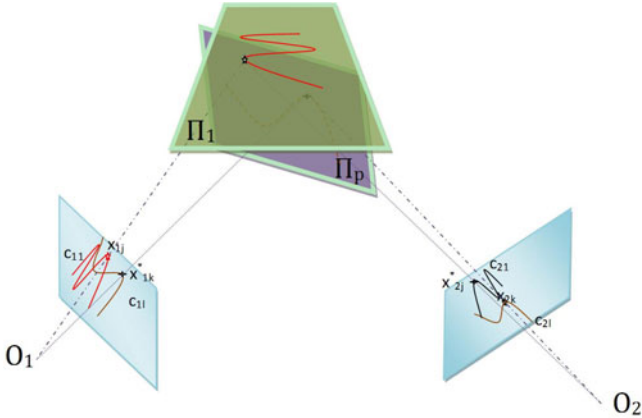


Fig. 1. Projective reconstruction of general curves

to fitted curves. Once 2 or more non-coplanar 3D planar curves are computed, 3D reconstruction for those two views can be readily performed. The curves visible on the two views are not required to be exactly the same portions of the 3D curve, hence the problem of occlusion can be handled. The paper is organized as follows: in Section II, the problem of reconstructing general 3D planar curves is formulated. In Section III, the details of the method is presented, together with the explanation of the algorithm. Experimental results are given in Section IV. In Section V, some concluding remarks are made.

2 Problem Formulation

Suppose that there are m general 3D curves C_l ($l = 1, 2, \dots, m$) lying on P ($P \leq m$) 3D planes Π_p ($p = 1, 2, \dots, P$). Each 3D curve can be seen by at least two views (in this paper, we will only consider 2-view cases but the approach can be extended to multi-views). For the l th 3D curve, the sets of 2D measured points on the first and second views are denoted as $\{x_{1l}\}$ and $\{x_{2l}\}$ respectively. Let $x_{il} = [u_{il}, v_{il}, 1]^T$ ($i = 1, 2$). Note that the index l doesn't indicate point-to-point correspondence between the two sets.

The projections on the two views of a 3D plane Π_p are related by a 3×3 non-singular homography, H_p . Suppose C_l is on Π_p . The set of 2D points $\{x_{2l}\}$ can be transformed to the first view as:

$$x_{1l}^* \sim H_p x_{2l}. \quad (1)$$

where \sim means equivalence up to scale.

The above relationship can be written into an equation by introducing a set of scale factors $\{\lambda_{2l}\}$:

$$\frac{1}{\lambda_{2l}} x_{1l}^* = H_p x_{2l}. \quad (2)$$

Let the projections of the 3D curve C_l on the first and second view be c_{1l} and c_{2l} respectively. The measured points $\{x_{1l}\}$ should ideally lie on c_{1l} ; and since $\{x_{2l}\}$

should lie on c_{2l} which corresponds to c_{1l} , the transferred points $\{x_{1l}^*\}$ should also lie on c_{1l} . Hence, we can obtain the set of curves $\mathbf{c} = \{c_{11}, c_{12}, \dots, c_{1m}\}$ by fitting them to both the measured and transferred points through the minimization problem:

$$E = \min_{\mathbf{H}, \mathbf{c}} \left(\sum_{l=1}^m d(x_{1l}, c_{1l})^2 + \beta \sum_{p=1}^P \sum_{l=1}^m d(\lambda_{2l} H_p x_{2l}, c_{1l})^2 \right). \quad (3)$$

subject to

$$\lambda_{2l} H_p^3 x_{2l} = 1. \quad (4)$$

where $d(x, c)$ is the Euclidean distance from point x to curve c ; β is a weighting factor in $[0, 1]$ and H_p^i is the i th row of the homography matrix H_p .

From (3), we can see that homographies are constrained by the second term while curve fitting relies on both of the terms. By setting a small β at the beginning and increasing it step by step (i.e. β takes value from set $\{\beta : \beta = 1 - e^{-n/a}, n \in \mathbb{N}, a = \text{const}\}$), the influence of the second term on curve fitting can be properly controlled and thus it can deal with the possibly poorly chosen initial homographies.

Instead of solving (3) directly with rigid constraint (4), we relax it to an unconstrained Weighted Least Square (WLS) problem by introducing a weighting factor γ to control the minimization problem from minimizing algebraic error to geometric error as γ increases. The relationship between algebraic error and geometric error is given in [1]. γ_{2l} are added as constant weighting factors [1]. Then the relaxed cost function could be written as:

$$E = \min_{\mathbf{H}, \mathbf{c}} \left\{ \sum_{l=1}^m d(x_{1l}, c_{1l})^2 + \beta \sum_{p=1}^P \sum_{l=1}^m \left(d(\lambda_{2l} H_p x_{2l}, c_{1l})^2 + \gamma \gamma_{2l} (1 - \lambda_{2l} H_p^3 x_{2l})^2 \right) \right\}. \quad (5)$$

The fitted curves on the first view together with optimized homographies provide one-to-one point correspondence along each curve across the two views, and thus facilitate stratified approach for projective reconstruction.

3 Projective Reconstruction

In this section, we will show how the problem can be reformulated into a multi-linear WLS problem which can be solved by an iterative algorithm.

3.1 Spline Curve Construction

According to the definition of the B-spline, a B-spline curve is defined by its order, knot vector and control points. We employ cubic spline for approximating c_{1l} in our algorithm, so the order of spline curve is four in this paper. To further simplify the problem, we use a uniform knot vector in the range of $[0, 1]$, leaving the spline curve to be totally determined by its control points which can be updated iteratively.

3.2 Cost Function for Minimization Algorithm

In Section 2, we have already formulated the problem of projective reconstruction into a minimization problem, as shown in (5). In this part we will rewrite it into a cost function which is suitable for minimization implementation. The following notation is used to describe our newly developed cost function:

- $\mathbf{B}_{lj}(t)$ is the j th spline basis for curve c_{1l} .
- w_{lj} is the j th control point of curve c_{1l} .
- $G_{1l}(t) = \sum_j w_{lj} \mathbf{B}_{lj}(t)$ is the spline description of curve c_{1l} .
- $\{t_{1l}\}$ and $\{t_{2l}\}$ are sets of corresponding parametric values of measured point set $\{x_{1l}\}$ and transferred point set $\{x_{2l}\}$.
- $\mathbf{H} = \{H_1, H_2, \dots, H_P\}$ is the homography set of P 3D planes.
- $\mathbf{w} = \{w_{lj}, l = 1, 2, \dots, m\}$ is the set of control points of all spline curves.
- $\mathbf{t} = \{t_{11}, t_{12}, \dots, t_{1m}, t_{21}, t_{22}, \dots, t_{2m}\}$ is the set of all corresponding parametric values on the spline curves.
- $\lambda = \{\lambda_{21}, \lambda_{22}, \dots, \lambda_{2m}\}$ is the set of homography scaling factors of all transferred points.

The cost function is written as follows:

$$F_c(\mathbf{H}, \mathbf{w}, \mathbf{t}, \lambda, \beta, \gamma) = \sum_{l=1}^{l=m} \|x_{1l} - G_{1l}(t_{1l})\|^2 + \beta \sum_{p=1}^P \sum_{l=1}^m \left\{ \|\lambda_{2l} H_p x_{2l} - G_{1l}(t_{2l})\|^2 + \gamma \gamma_{2l} (1 - \lambda_{2l} H_p^3 x_{2l})^2 \right\}. \quad (6)$$

In (6), when β is small (i.e. $\beta \rightarrow 0$), the influence of the second term on curve fitting will be small so that the curve fitting is faithful to the original measured points of the first view; when $\beta \rightarrow 1$, the cost function has equal weights with regard to the points measured on the first view and those transferred from the second view. This automatically solves the curve occlusion problem. Also, when γ is small (i.e. $\gamma = 1$), the algebraic error between the transferred point and its corresponding point on the curve is minimized; whereas when $\gamma \rightarrow \infty$, the algebraic error becomes the geometric distance from the measured and transferred points to the curves.

3.3 Algorithm Initialization

One of the main advantages of this method is its flexibility of initialization: no point correspondences are needed to be known for this algorithm. However, three initializations must be done for the algorithm to start with. They are determining the first view, estimating the initial homographies and constructing the initial spline curves.

To determine the first view, one suggested strategy is to calculate the arc-length and the enclosed area of each curve (for open curves, connect the start and end points to estimate its area); and take the view with the smallest arc-length to area ratio as the first view.

In the literature, there are a number of methods to estimate the affine transformation between two point-sets, Fitzgibbon [11] suggested an improved Iterative Closest Point (ICP) algorithm which can be used to generate satisfactory initial homographies for our algorithm even there are missing data.

To deal with initial spline curves, we use split-merge algorithm, which is effective in not only determining the number of control points to be used but also providing the initial locations of control points.

3.4 Algorithm for Projective Reconstruction

As stated in Section 2, we are given m pairs of general curves on two views projected from P ($P \geq 2$) 3D planes. The curves on the two views are described as measured points $\{x_{1l}\}$ and $\{x_{2l}\}$, thus $\{x_{1l}\}$ and $\{x_{2l}\}$ in the cost function are already known. As the order of the spline curve is chosen to be four in this paper, the basis $\mathbf{B}_{lj}(t)$ is also determined considering that we have already set the knot vector to be uniform. Thus for fixed β and γ , when we keep \mathbf{t} fixed, the cost function $F_c(\mathbf{H}, \mathbf{w}, \mathbf{t}, \lambda, \beta, \gamma)$ is tri-linear with regard to the homographies \mathbf{H} , the control points \mathbf{w} and the homography scaling factors λ , which could be minimized with standard WLS method; when \mathbf{H} , \mathbf{w} and λ are fixed, minimizing $F_c(\mathbf{H}, \mathbf{w}, \mathbf{t}, \lambda, \beta, \gamma)$ becomes a geometric problem which is to find the nearest points on its corresponding spline curves $G_{1l}(t)$ and can easily be solved in an analytical way; consequently, the cost function F_c can be minimized by iteratively finding the optimums of \mathbf{H} , \mathbf{w} , λ and \mathbf{t} . When β and γ are increasing, the minimized cost function is going from the algebraic error to the geometric error. Thus the equation (6) generates optimum solution \mathbf{H}^* , \mathbf{w}^* , \mathbf{t}^* and λ^* by minimizing the geometric distance from measured and transferred points to fitted curves. Once optimal \mathbf{H} , \mathbf{w} , \mathbf{t} and λ are obtained, the fundamental matrix F relating the two views can be retrieved by either using the homographies [12] or employing the eight-point algorithm [13] with the optimal corresponding points. And the projection matrix can be written as [14]:

$$P_1 = [\mathbf{I}_{3 \times 3}, \mathbf{0}_{3 \times 1}]. \quad (7)$$

$$P_2 = [[e']_{\times} F, e']. \quad (8)$$

where e' is the epipole on the second view.

With the projection matrix and one-to-one point correspondence along the curves, the 3D projective reconstruction can be readily performed. The algorithm is described as follows:

Algorithm 1

1. Initialization

Put $k = 0$ and set $\beta^{(0)} = 0, \gamma^{(0)} = 1$;

Initialize $\mathbf{H}^{(0)}$ and $\mathbf{w}^{(0)}$ according to the strategies described in Section 3.3.

2. Put $k = k + 1$

3. Fix $\mathbf{H}^{(k-1)}$, $\mathbf{w}^{(k-1)}$ and $\lambda^{(k-1)}$, determine t_{1l} and t_{2l} by solving:

$$err'_k = \min_{\mathbf{t}^{(k)}} F_c \left(\mathbf{H}^{(k-1)}, \mathbf{w}^{(k-1)}, \mathbf{t}^{(k-1)}, \lambda^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)} \right). \quad (9)$$

4. Fix $\mathbf{w}^{(k-1)}, \mathbf{t}^{(k)}$ and $\lambda^{(k-1)}$, find $\mathbf{H}^{(k)}$ by solving

$$err_k'' = \min_{\mathbf{H}^{(k)}} F_c \left(\mathbf{H}^{(k)}, \mathbf{w}^{(k-1)}, \mathbf{t}^{(k)}, \lambda^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)} \right). \quad (10)$$

5. Fix $\mathbf{H}^{(k)}, \mathbf{w}^{(k-1)}$ and $\mathbf{t}^{(k)}$, determine $\lambda^{(k)}$ by solving:

$$err_k''' = \min_{\lambda^{(k)}} F_c \left(\mathbf{H}^{(k)}, \mathbf{w}^{(k-1)}, \mathbf{t}^{(k)}, \lambda^{(k)}, \beta^{(k-1)}, \gamma^{(k-1)} \right). \quad (11)$$

6. Fix $\mathbf{H}^{(k)}, \mathbf{t}^{(k)}$ and $\lambda^{(k)}$, find $\mathbf{w}^{(k)}$ by solving:

$$err_k = \min_{\mathbf{w}^{(k)}} F_c \left(\mathbf{H}^{(k)}, \mathbf{w}^{(k)}, \mathbf{t}^{(k)}, \lambda^{(k)}, \beta^{(k-1)}, \gamma^{(k-1)} \right) \quad (12)$$

7. If $|err_k - err_k'| \geq \varepsilon$ (e.g. $\varepsilon = 10^{-4}$), increase β ; else increase γ by $\gamma = 1.1\gamma$.
8. If $k \leq N$ and $\gamma \leq M$ (e.g. $M = 10000, N = 1000$), go to Step 2.
9. Compute the fundamental matrix F using the optimized homographies or new point-to-point correspondences.
10. Compute the projection matrix according to (7) and (8). Output projection matrix P_1 and P_2 .
11. End

The reconstructed curves in a projective 3D space can be upgraded to Euclidean space by enforcing metric constraints on the projection matrices, as proposed by Tang [15].

4 Experimental Results

To demonstrate the performance of our proposed method, both synthetic data and real images are tested in this section.

4.1 Synthetic Data

A synthetic scene is given in Fig. 2(a), which consists of two general curves on two orthogonal planes in 3D space: the blue one is a sine function while the red one is a high-order curve, defined as:

$$\begin{cases} z = 6 \sin(y) \\ x = 0 \end{cases}, \quad y \in [-2\pi \ 2\pi] \quad (13)$$

and

$$\begin{cases} z = 3.5 \times 10^{-4} (x^2 - 25)^3 + 1 \\ y = 0 \end{cases}, \quad x \in [-7.5 \ 7.5] \quad (14)$$

Curves are projected on two views by two cameras at randomly generated locations with focal length 340 and image size 320×320 .

Along each of the projected curves on both of the views, 200 points are sampled randomly; then Gaussian noise with standard deviation 1 pixel is added

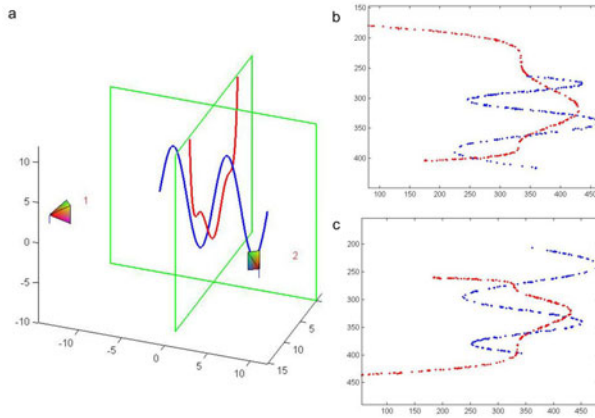


Fig. 2. The synthetic scene together with two synthetic views. (a) The 3D scene used to evaluate the performance of the proposed algorithm. (b) The view taken by camera one. (c) Another view taken by camera two.

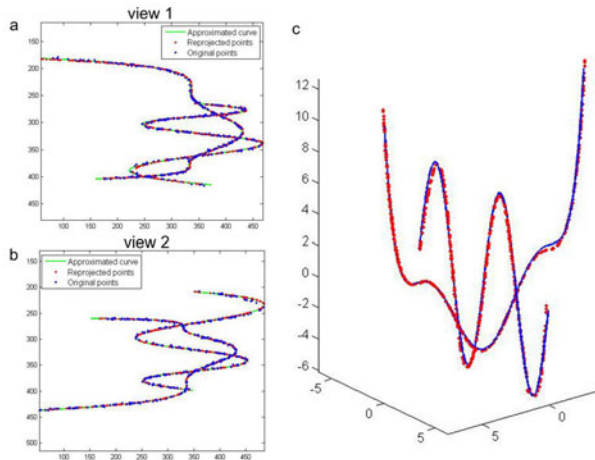


Fig. 3. The reconstruction results. (a) The approximation of curves on view one, where the black dots are measured points on view one; the red dots are transferred points from view two; the green lines are approximated curves. (b) the approximation of curves on view two, with same notation as in (a). (c) 3D reconstruction of measured points upgraded to Euclidean space and super-imposed with ground truth curves.

independently to both the x - and y -coordinates of those points on both of the views, as shown in Fig. 2(b) and (c).

The estimated curves on the first and second view together with reprojected measured points are shown in Fig. 3(a) and (b) respectively. The upgraded reconstruction result is shown in Fig. 3(c). We can see from the result that the reconstructed points lie quite close to that of the ground truth curves,

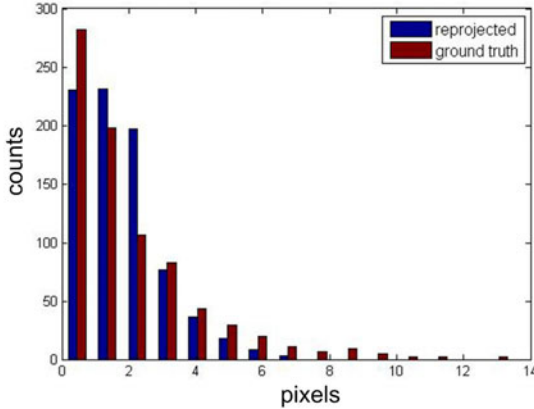


Fig. 4. The histogram comparison of noise distributions: the brown bars indicate the ground truth distribution. While the blue ones indicate the distance of measured points to reprojected curves.

indicating that our algorithm converges to the right solution. Also, from the point to reprojected curve distance shown in Fig. 4, we can see that the reconstructed point-to-curve error is quite close to that of the ground truth distribution of added noise in the simulation.

4.2 Real Image Curves

The proposed algorithm is also checked with real image curves. Two images of printed pictures of an apple and a pear are taken at two different positions, as shown in Fig. 5(a) and (b). Curves on these images are extracted with canny

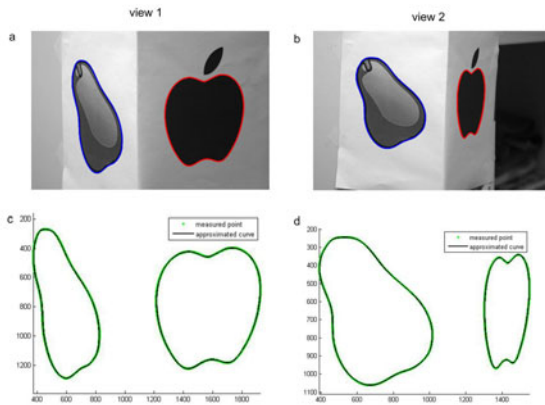


Fig. 5. (a) and (b) Two images of the printed picture taken at two different positions. (c) and (d) Extracted edge points and fitted curves on views one and two.

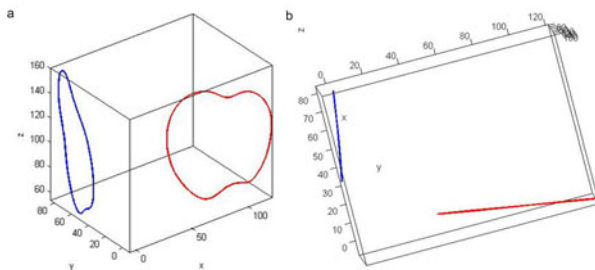


Fig. 6. (a) The reconstructed scene of real images. (b) A top view of the reconstruction result.

edge detector to generate the measured points on the curves. These extracted curves are then processed using our proposed algorithm until it converges, outputting the projection matrix for projective reconstruction. Finally, the results are upgraded to Euclidean space, as shown in Fig. 6.

5 Conclusion

In this paper, a new approach is developed to reconstruct general 3D planar curves from two images taken by uncalibrated cameras. By minimizing the sum of squares of Euclidean point to curve distances, we retrieve the curves on the first view and obtain an optimum homography matrix, thus established one-to-one point correspondences along the curves across the two views. With curves lying on more than one 3D plane, the fundamental matrix are computed, allowing the 3D projective reconstruction to be readily performed.

References

1. Hung, Y.S., Tang, A.W.K.: Projective reconstruction from multiple views with minimization of 2D reprojection error. *International Journal of Computer Vision* 66, 305–317 (2006)
2. Tang, A.W.K., Ng, T.P., Hung, Y.S., Leung, C.H.: Projective reconstruction from line-correspondences in multiple uncalibrated images. *Pattern Recognition* 39, 889–896 (2006)
3. Hartley, R.I.: Projective reconstruction from line correspondences. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1994)
4. Long, Q.: Conic reconstruction and correspondence from two views. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 151–160 (1996)
5. Ma, S.D., Chen, X.: Quadric reconstruction from its occluding contours. In: *Proceedings International Conference of Pattern Recognition* (1994)
6. Ma, S.D., Li, L.: Ellipsoid reconstruction from three perspective views. In: *Proceedings International Conference of Pattern Recognition* (1996)
7. Mai, F., Hung, Y.S., Chesi, G.: Projective reconstruction of ellipses from multiple images. *Pattern Recognition* 43, 545–556 (2010)

8. Kaminski, J., Shashua, A.: On calibration and reconstruction from planar curves. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 678–694. Springer, Heidelberg (2000)
9. Kaminski, J.Y., Shashua, A.: Multiple view geometry of general algebraic curves. *International Journal of Computer Vision* 56, 195–219 (2004)
10. Berthilsson, R., Astrijm, K., Heyden, A.: Reconstruction of general curves, using factorization and bundle adjustment. *International Journal of Computer Vision* 41, 171–182 (2001)
11. Fitzgibbon, A.W.: Robust registration of 2D and 3D point sets. *Image and Vision Computing* 21, 1145–1153 (2003)
12. Luong, Q.T., Faugeras, O.D.: The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision* 17, 43–75 (1996)
13. Hartley, R.I.: In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence* 19, 580–593 (1997)
14. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge (2000)
15. Tang, A.W.K.: A factorization-based approach to 3D reconstruction from multiple uncalibrated images. Ph.D. Dissertation (2004)

A Novel Photometric Method for Real-Time 3D Reconstruction of Fingerprint

Wuyuan Xie^{1,2}, Zhan Song^{1,2}, and Xiaoting Zhang^{1,2}

¹ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

² The Chinese University of Hong Kong, Hong Kong, China

{wy.xie, zhan.song, xt.zhang}@sub.siat.ac.cn

Abstract. 3D fingerprint recognition is an emerging technology in biometrics. However, current 3D fingerprint acquisition systems are usually with complex structure and high-cost and that has become the main obstacle for its popularization. In this work, we present a novel photometric method and an experimental setup for real-time 3D fingerprint reconstruction. The proposed system consists of seven LED lights that mounted around one camera. In the surface reflectance modeling of finger surface, a simplified Hanrahan-Krueger model is introduced. And a neural network approach is used to solve the model for accurate estimation of surface normals. A calibration method is also proposed to determine the lighting directions as well as the correction of the lighting fields. Moreover, to stand out the fingerprint ridge features and get better visual effects, a linear transformation is applied to the recovered normals field. Experiments on live fingerprint and the comparison with traditional photometric stereo algorithm are used to demonstrate its high performance.

1 Introduction

Fingerprint recognition has been a classical topic in computer vision community over last decades. Current fingerprint system usually consists of an image sensor, a light source and a touch panel, and the recognition task is performed over the captured 2D fingerprint image. In practice, the imaged fingerprint is usually degraded caused by improper finger placement, skin deformation, slippage, smearing of finger and touch panel surface [1] etc. Moreover, the touching fingerprint collection method also causes the problem of disease propagation. To overcome these drawbacks, a technique named touch-less fingerprint recognition systems is emerging recently. Such systems can obtain the 3D model of the finger surface by employing a 3D scanning procedure. In comparison with traditional 2D fingerprint system, the 3D system can outperform in both recognition rate and operation efficiency [2-3].

The touch-less live fingerprints acquisition is essentially a problem of 3D surface reconstruction. And methods used for this end are nothing more than those common reconstruction methods in computer vision like Structured Light System (SLS), stereo vision and Shape from Silhouette etc. As a frontier technology, there still few works have been reported in this domain. In [2], an experimental system which consists of five cameras and sixteen green LED lights is proposed. The fingerprint images under

various illuminations are captured by the cameras. Corresponding silhouettes are extracted from these images and then are used for the 3D modelling of the finger surface via Shape from Silhouette method. In [3], a structured light system is proposed for the 3D reconstruction of fingerprint. The system contains a projection device and a synchronized camera. By projecting a sequence of strip patterns onto the finger surface and imaged by the camera, 3D model of the fingerprint can be achieved in less than one second. The two systems have been demonstrated to be effective for quick 3D fingerprint acquisition and outperform current 2D systems in recognition rates. However, the main obstacle for present 3D fingerprint technologies to replace 2D systems comes from their complex structure and high cost. More portable and low-cost devices for quick 3D acquisition of fingerprint are urgently demanded.

In this paper, a novel touch-less 3D fingerprint acquisition system that based upon the principle of Photometric Stereo (PS) is proposed. The system consists of only one camera and seven LED lamps. In the PS methods, its performance highly depends on the surface reflectance modeling as well as the lighting conditions. With a brief analysis and comparison of some present human skin reflectance models, the more reasonable Hanrahan-Krueger (HK) [4] model is used and simplified in our work. A novel method for the calibration of the LED lights including the light direction and correction of the lighting field is proposed. Finally, a surface normal transformation procedure is applied to boost the fingerprints details like the surface ridge features.

The paper is organized as follows. In Section 2, we briefly analyze the rationality of the simplified HK model as well as the solution of the model parameters. In Section 3, system calibration and the surface normal transformation methods are presented. Experimental results on a plastic toy, palm and live fingerprints are offered in Section 4. Conclusion and future work can be found in Section 5.

2 Reflectance Property Modeling of the Finger Surface

Traditional PS methods are usually based on the assumption of Lambert reflection law [5-8], i.e., the target surfaces are supposed with ideal diffuse reflection. Given three or more lights with known directions, surface normal at any image point can be calculated by solving a group of linear Lambertian equations. As for human skin, it is more close to a kind of translucent material whose reflectance model contains certain multiple scattering and specular reflections. And thus makes traditional PS method incapable to get precise 3D reconstruction result. To model the human skin more precisely, Georgiades [9] has introduced a non-Lambertian reflectance model, the Torrance and Sparrow (TS) model, into an uncalibrated PS method to calculate reflectance parameters of human skin and to reduce the negative effects of generalized bas-relief (GBR) [10]. TS model is a physically-based model which assumes that the skin reflectance consists of two components: a) Lambertian lobe at a particular position on the skin and b) purely surface scattering component. In comparison, Hanrahan-Krueger (HK) is such a model for subsurface scattering in layered surfaces based on one dimensional linear transport theory. The basic idea is that the amount of light reflected by a material that exhibits subsurface scattering is calculated by summing

the amount of light reflected by each layer times the percentage of light that actually reaches that layer. In the algorithm, skin is modeled as a two-layer material which consists of epidermis and dermis, and each layer has different reflectance parameters that determine how light reflects from that layer as shown in Fig. 1. Therefore it's a more reasonable model for translucent surfaces like human skin and fingers.

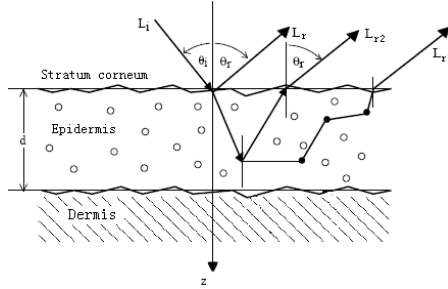


Fig. 1. Reflection and multiple scattering in the two-layer HK skin model

In the original HK model, each layer is parameterized by the absorption cross section σ_a , scattering cross section σ_s , layer thickness d , and the mean cosine g of the phase function. And p determines in which direction the light is likely to scatter as (1), where ϕ is the angle between the light and the view directions:

$$p(\phi, g) = \frac{1}{4\pi} \cdot \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \phi)^{2/3}} \quad (1)$$

The total cross section σ_t , optical depth τ_d and albedo ζ can be expressed as:

$$\sigma_t = \sigma_s + \sigma_a, \quad \tau_d = \sigma_t \cdot d, \quad \zeta = \sigma_s / \sigma_t \quad (2)$$

where σ_t represents the expected number of either absorptions or scatterings per unit length. The optical depth is used to determine how much light is attenuated from the top to the bottom of the layer. ζ indicates the percentage of interactions that result in scattering. More details about HK model discussion can be found in [4].

In this work, to simplify the HK model, a Lambertian term L_m is used to approximate the multiple scattering $L_{r2}, L_{r3}, \dots, L_{rm}$ as:

$$L_m = L_{r2} + L_{r3} + \dots + L_{rm} = \rho \cos \theta_i \quad (3)$$

where ρ is an defined albedo, which determines the amount of diffuse light caused by multiple scattering, and θ_i is the incidence angle between the normal vector and the light direction. Then the revised HK model can be written as:

$$L_r(\theta_r, \varphi_r) = \zeta T^{12} T^{21} p(\phi, g) \frac{\cos \theta_i}{\cos \theta_i + \cos \theta_r} (1 - e^{-\tau_d (\frac{1}{\cos \theta_i} + \frac{1}{\cos \theta_r})}) L_i(\theta_i, \varphi_i) + L_m$$

where T^{12} and T^{21} refer to the Fresnel transmittance terms for lights entering and leaving the surface respectively, and are assumed to be constant over the whole surface as well as g . By replacing $p(\phi, g)$, (ζ, τ_d) and L_m with the right side of Eqn.(1), Eqn.(2), and Eqn.(3) respectively, the simplified HK model can be expressed as:

$$L_r = \frac{\delta_s}{\delta_i} T^{12} T^{21} \frac{(1-g)^2 \cos \theta_i (1 - e^{-(\delta_i d) \left(\frac{\cos \theta_r + \cos \theta_i}{\cos \theta_r \cos \theta_i} \right)}) L_i}{16\pi^2 (1+g^2 - 2g \cos \phi)^{2/3} (\cos \theta_i + \cos \theta_r)} + \rho \cos \theta_i \quad (4)$$

Given any incident light $L_i(\theta_i, \phi_i)$ with incident angle (θ_i, ϕ_i) , the reflection light $L_r(\theta_r, \phi_r)$ can be calculated through Eqn. (4). Eqn. (4) is only calculated for epidermis. Mostly, scattering in the dermis layer will do minor contributions to the final fingerprint image. As a result, in our model, parameters in this layer are ignored and thus also benefit the whole computation cost.

To solve the surface normal vector $\mathbf{n} = (n_x, n_y, n_z)$, $\cos \theta_i$, $\cos \phi$ and $\cos \theta_r$, Eqn. (4) can be rewritten in the form of \mathbf{n} 's inner product between lighting direction $\mathbf{l} = (l_x, l_y, l_z)$ and the view direction $\mathbf{z} = (z_x, z_y, z_z)$ as:

$$\cos \theta_i = \mathbf{l} \cdot \mathbf{n}, \quad \cos \theta_r = \mathbf{z} \cdot \mathbf{n}, \quad \cos \phi = \mathbf{l} \cdot \mathbf{z} \quad (5)$$

There are ten unknowns i.e. $n_x, n_y, n_z, \rho, d, \sigma_s, \sigma_a, g, T^{12}$ and T^{21} in Eqn. (4). Notice that each surface point is associated with a unique quadruple $(\theta_i, \phi_i; \theta_r, \phi_r)$. To clarify the statements, two superscripts are used to mark those point-unique parameters with the first refers to the order of the surface point and the second one indicates the number of light source. For example, $L_r^{j,k}$ represents $L_r(\theta_r, \phi_r)$ of the j^{th} surface point under illumination of the k^{th} light, \mathbf{n}^j refers to normal vector of the j^{th} surface point, while \mathbf{l}^k represents the direction vector of the k^{th} light. Then we can formulate the simultaneous recovery to the following nonlinear optimization problem:

$$\arg \min_{\mathbf{x}} E(\mathbf{x}), \quad \text{where } E(\mathbf{x}) = \sum_{j,k} (L_r^{j,k} - I^{j,k})^2 \quad (6)$$

where $I^{j,k}$ represents the pixel intensity on the k^{th} image of the j^{th} surface point, and \mathbf{x} is a vector containing all the unknown parameters to be estimated, i.e.,

$$\mathbf{x} = (n_x^j, n_y^j, n_z^j, \rho^j, d^j, \sigma_s^j, \sigma_a^j, g, T^{12}, T^{21}), \{\mathbf{x} \in \mathbf{R}^n\}$$

To solve \mathbf{x} from Eqn. (6), a neural network algorithm named ZCNK is used in this work. As described in [11], the object function can be written as:

$$F(\mathbf{x}) = \partial E / \partial \mathbf{x} \quad (7)$$

Since \mathbf{x} is an n -dimensional vector, $F(\mathbf{x})$ consists of n equations. Unlike the initialization method in [11] which takes optional values for all components of \mathbf{x} , we initialize \mathbf{x} as following way:

- (a) For \mathbf{n}^j and ρ^j , their initial values are calculated from Equation (3) through the least square method;
- (b) For $d^j, \sigma_s^j, \sigma_a^j$, and g , we refer to [4] for their initial values;
- (c) T^{12} and T^{21} are initialized randomly;

3 System Calibration and 3D Recovery

3.1 Calibration of Lighting Directions

To determine incident light angles of all the lights, a method proposed in [12] is used. In [12], two cameras and a shiny ball without position and radius information are used to calibrate the lighting directions. Since just one camera is used in our system, the lighting directions are calibrated by the use of one camera and a shiny ball with known radius r . Focal length f and principle point C of the camera can be obtained via method mentioned in [13]. With reference to the camera coordinate frame, each lighting direction can be represented as $\mathbf{l}=(lx,ly,lz)$ as shown in Fig. 2(a). According to [12], the radius d_i can be expressed as:

$$d_i^2 = |CS|^2 - \cos^2 \theta = h_i(s_x, s_y, s_z), \quad \cos \theta = \frac{(CS \cdot CB_i)}{|CS||CB_i|} = \frac{\sqrt{|CS|^2 - d_i^2}}{|CS|} \quad (8)$$

where B_i is a boundary point on the image plane with a checked position (i_1, j_1, f) , then Eqn. (8) contains only three unknowns, i.e. center of the sphere $S=(s_x, s_y, s_z)$. Suppose there are m boundary points $B_i, i \in (1, \dots, m)$ obtained by edge detection algorithm [14], then an error function can be defined as:

$$EOF = \sum_{i=1}^m (h_i - r^2)^2 \quad (9)$$

Minimizing this error function gives us three equations about S , from which the optimal values of the sphere location can be solved. Notice that specular point $P(i_1, j_1)$ on the image plane can be easily detected by finding the brightest image point. Once S is obtained, the corresponding point S_1 of p on the surface and the surface normal vector \mathbf{N} on S_1 , as illustrated in Fig. 2(b), can be solved as:

$$\mathbf{I} = 2(\mathbf{N} \cdot \mathbf{R})\mathbf{N} - \mathbf{R} \quad (10)$$

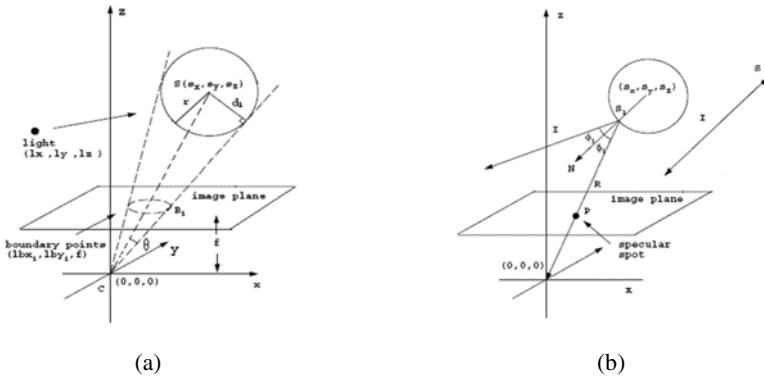


Fig. 2. (a) Illustration of light direction calibration. (b) Relation of N , R , and I .

3.2 Correction of Lighting Field

Since lighting fields of these LED lamps are usually not uniformly distributed as shown in Fig. 3(a). Notice that when the light position is fixed, the brightest spot has a fixed area. This means that under the illumination of a fixed LED, to any image point at (i, j) , the portion $p(i, j)$ of its intensity value $I(i, j)$ to the brightest pixel value I_{max} is constant no matter what kind of the shape of the object is, i.e.,

$$p(i, j) = I(i, j)/I_{max} \tag{11}$$

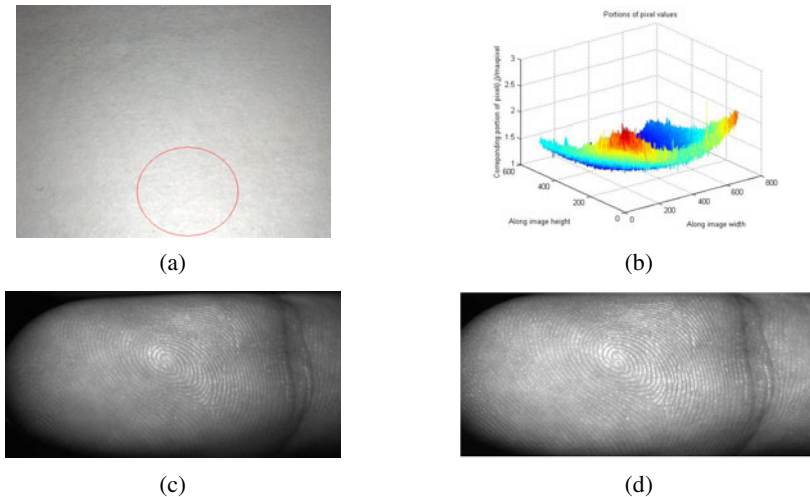


Fig. 3. (a) A white paper is used to correct the LED lighting field so as to make the luminance distribution more uniform. (b) The diagram of values of p that calculated from (a). (c) Original image without calibration. (d) Result after lighting field correction.

Fig. 3(b) gives the diagram of values of p that calculated from Fig. 3(a). With above procedure, the resulting image can have the same illumination at all image points under l^k . Fig. 3(c) and (d) give the comparison between the original image and the corrected image with the proposed method.

3.3 Linear Transformation of the Normal Field

Once surface normal vector \mathbf{n} has been estimated, a linear normal transformation method [7] is adopted consequently. In the algorithm, the average of normal vectors over a local patch w is calculated as a local reference, and the difference between the original surface normal and this reference vector can be amplified as:

$$\mathbf{n}' = \mathbf{n} + k(\mathbf{n} - \text{normalize}(\sum_{j=1}^w \mathbf{n}_j)) \quad (12)$$

This procedure aims to amplify the angle between two neighboring normal vectors and thus improve the visual effect and contrast of the reconstructed 3D model especially to the fingerprint ridges.

4 Experimental Results

Our experimental system consists of a camera with resolution of 659×493 pixels and seven LED lamps mounted around it as shown in Fig. 4. An external I/O board is developed to synchronize the camera and LED lamps. The whole capturing time can be controlled within 0.2 s. Each lamp is connected with a metallic hose so that its illuminant angle can be adjusted freely to fit the size of the target object.

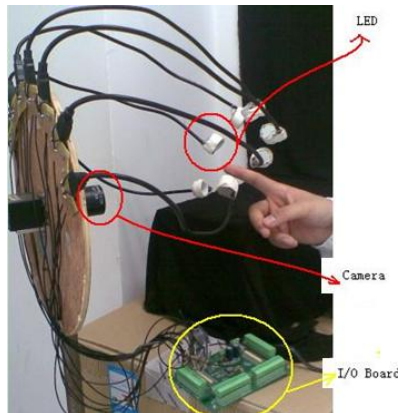


Fig. 4. The experimental system consists of a camera and seven LED lamps and they are synchronized via an external I/O board

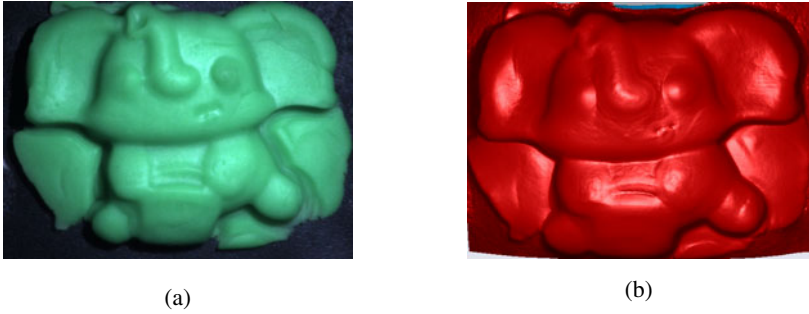


Fig. 5. Reconstruction of a plasticine toy with translucent surface

In the experiment with a plasticine toy as shown in Fig. 5, the parameters d^j , σ_s^j , σ_a^j and g are set to 0.085 mm, 50 mm⁻¹, 3.8 mm⁻¹ and 0.79 respectively.

In the experiment with human palm and fingerprint, the four parameters are set to 0.12 mm, 30 mm⁻¹, 4.5 mm⁻¹ and 0.81 respectively. Fig. 6(a) shows the original image of human palm image under one LED illumination. The reconstruction result without normal transformation is as shown in Fig. 6(b). The result looks smooth with palm print almost invisible. After the transformation (w and k in Eqn. (8) are set to 7

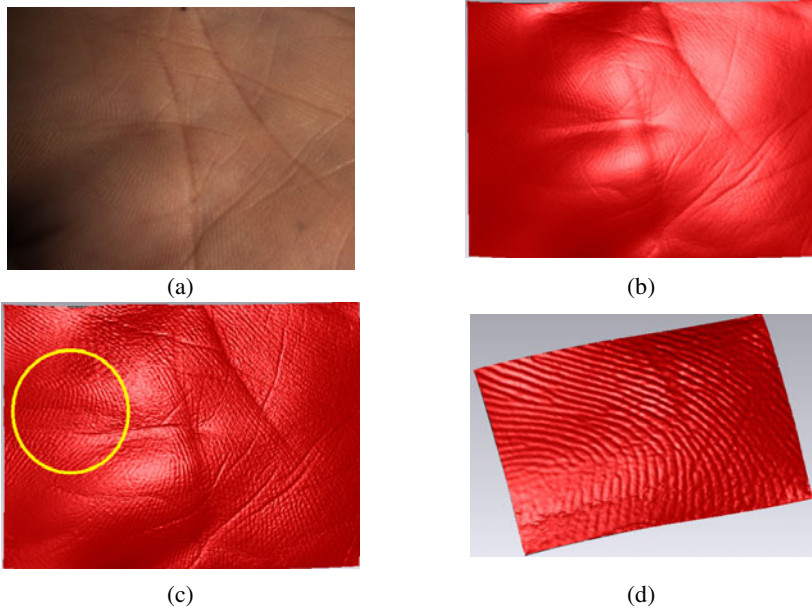


Fig. 6. 3D reconstruction of palm print. (a) Original palm image under one LED illumination. (b) The reconstructed 3D model of palm print without surface normal transformation. (c) 3D model with normal transformation. (d) Cropped 3D image for close observation.

and 2.5 respectively), palm print can be discovered more visually as shown Fig. 6(c). One cropped area is enlarged in Fig. 6(d) for close observation.

The last experiment is conducted on a live finger as shown in Fig. 7(a). The result by traditional Lambert model is also presented for comparison. From Fig. 7(b), we can see that Lambert model fails to obtain skeletons of the fingerprint in the margins. It is mainly caused by that in Lambertian reflection model, pixel brightness is supposed to be independent on viewing direction. But it is not true for finger surface, and therefore makes the recovered surface normal inaccurate as well as 3D shape.

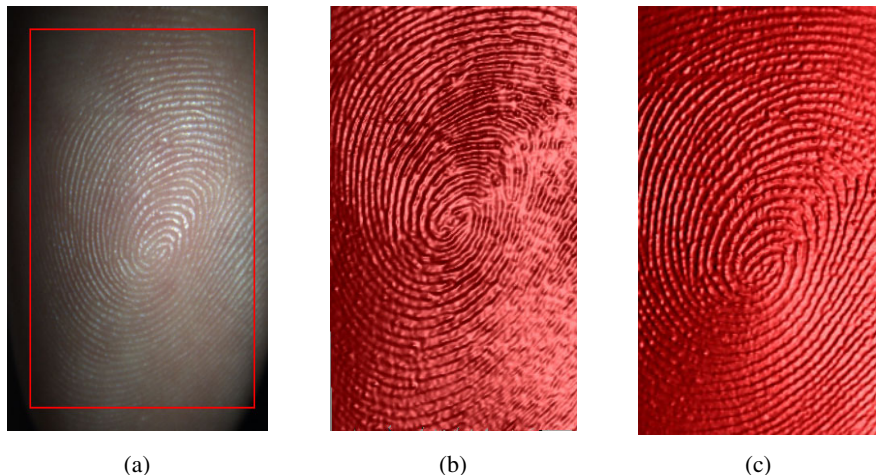


Fig. 7. (a) Finger image under one LED lighting. (b) Reconstruction result via traditional PS method which assumes a Lambert reflectance model. (c) Result by the proposed method.

5 Conclusion and Future Work

In this paper, we have proposed a novel photometric based system for the real-time 3D reconstruction of fingerprint. The system consists of only one camera and seven LED lamps. In the modeling of surface reflectance property, different with previous PS methods, we proposed a simplified HK model by using a Lambertian term to approximate the multiple scattering. A neural network algorithm is used for accurate model parameters estimation. A calibration method is also proposed to determine the lighting directions as well as the correction of the lighting fields. Finally, to improve the visual effect of the reconstructed 3D model, a linear surface normal transformation is introduced. The experiments are conducted with a plasticine toy and human palm and finger to demonstrate its high performance. In comparison with traditional Lambertian based surface model, the proposed method can reconstruct the finger surface in finer details.

Future work can address the minimization of the whole hardware and the use of infrared LED lights so as to make the system more insensitive to ambient lights. We are also working on the 3D fingerprint processing algorithms and finally it will be

integrated with the hardware to accomplish a complete touch-less 3D fingerprint recognition system.

Acknowledgments

The work described in this article was partially supported by NSFC (Project no. 61002040) and Knowledge Innovation Program of the Chinese Academy of Sciences (Grant no. KGCX2-YW-156).

References

1. Delac, K., Grgic, M.: A Survey of Biometric Recognition Methods. In: 46th International Symposium in Electronics in Marine, pp. 184–193 (2004)
2. Chen, Y., Pariziale, G., Eva, D.S., Jain, A.K.: 3D Touchless Fingerprints: Compatibility with Legacy Rolled Images. In: Proceedings of Biometric, pp. 1–6 (2006)
3. Wang, Y.C., Daniel, L., Hassebrook, L.G.: Fit-Sphere Unwrapping and Performance Analysis of 3D Fingerprints. *Applied Optics* 49(4), 592–600 (2010)
4. Hanrahan, P., Krueger, W.: Reflection from Layered Surfaces due to Subsurface Scattering. In: Proceedings of SIGGRAPH, pp. 165–174 (1993)
5. Barsky, S., Petrou, M.: The 4-Source Photometric Stereo Technique for Three-Dimensional Surfaces in the Presence of Highlights and Shadows. *IEEE Trans. on PAMI* 25(10), 1239–1252 (2003)
6. Malzbender, T., Wilburn, B., Gelb, D., Ambrisco, B.: Surface Enhancement Using Real-time Photometric Stereo and Reflectance Transformation. In: Proceedings of the European Symposium on Rendering, pp. 245–250 (2006)
7. Sun, J.A., Smith, M., Smith, L., Midha, S., Bamber, J.: Object Surface Recovery using a Multi-light Photometric Stereo Technique for Non-Lambertian Surfaces Subject to Shadows and Specularities. *Image and Vision Computing* 25(7), 1050–1057 (2007)
8. Osten, W.: A Simple and Efficient Optical 3D-Sensor based on Photometric Stereo. In: The 5th International Workshop on Automatic Processing of Fringe Patterns, pp. 702–706 (2005)
9. Georgiades, A.S.: Incorporating the Torrance and Sparrow Model of Reflectance in Uncalibrated Photometric Stereo. In: ICCV, vol. 2, pp. 591–597 (2003)
10. Belhumeur, P., Kriegman, D., Yuille, A.: The Bas-Relief Ambiguity. In: CVPR, pp. 1040–1046 (1997)
11. Zhao, H., Chen, K.Z.: Neural Network for Solving Systems of Nonlinear Equations. *Acta Electronica Sinica* 30(4) (2002)
12. Zhou, W., Kambhampettu, C.: Estimation of Illuminant Direction and Intensity of Multiple Light Sources. In: Heyden, A., et al. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 206–220. Springer, Heidelberg (2002)
13. Zhang, Z.: A Flexible New Technique for Camera Calibration. *IEEE Trans. on PAMI* 22(11), 1330–1334 (2000)
14. Canny, J.: A Computational Approach to Edge Detection. *IEEE Trans. on PAMI* 8, 679–714 (1986)

3D Camera Pose Estimation Using Line Correspondences and 1D Homographies

Irene Reisner-Kollmann¹, Andreas Reichinger¹, and Werner Purgathofer²

¹ VRVis Research Center

² Vienna University of Technology

Abstract. This paper describes a new method for matching line segments between two images in order to compute the relative camera pose. This approach improves the camera pose for images lacking stable point features but where straight line segments are available. The line matching algorithm is divided into two stages: At first, scale-invariant feature points along the lines are matched incorporating a one-dimensional homography. Then, corresponding line segments are selected based on the quality of the estimated homography and epipolar constraints. Based on two line segment correspondences the relative orientation between two images can be calculated.

1 Introduction

Matching features between two images is an important task in 3D computer vision, e.g. for camera parameter estimation, image retrieval or classification. Local, viewpoint invariant region descriptors are used for these tasks in many applications. They are independently extracted in input images and similar descriptors are selected as putative feature correspondences. Region descriptors are very robust for many scenes, but areas with distinctive textures are required [1]. In this paper we create correspondences between line segments instead of regions. This approach improves the results for scenes which contain few stable 2D features as caused by large homogeneous surfaces, but instead contain straight lines, e.g. urban scenes or interior rooms. The search space for corresponding features is decreased to salient line segments which in turn increases the distinctiveness of the feature descriptors.

Our algorithm matches the intensity profiles along line segments by matching distinctive feature points within these profiles. Corresponding feature points located along straight lines are correlated by a one-dimensional homography. This homography establishes an important constraint on the set of correspondencing points. The similarity between two line segments is computed by comparing intensity values in the line profiles based on the estimated 1D homography. The final line segment correspondences are selected according to reprojection errors from a robust estimation of the camera parameters for the input images.

The advantage of our algorithm is that it doesn't rely on two-dimensional extremal features which may be sparse in some scenes. Splitting the problem

into a two-step approach reduces the dimensionality of the search space for correspondences. This has the advantage that there are fewer possibilities for corresponding points and the RANSAC estimator is more likely to find the best consistent matches.

The remainder of this paper is organized as follows. Section 2 describes the extraction of stable feature points along salient image lines. Section 3 presents how feature points are matched and how point correspondences are used for matching lines and estimating the relative orientation between two input images. Experimental results of our algorithm are presented in Section 4.

1.1 Related Work

Camera Pose Estimation. A widely used approach for computing the relative pose of two cameras is the usage of locally invariant region descriptors [1,2,3]. Such region descriptors are invariant to a varying range of affine transformations in order to compare them in images from different viewpoints. Additionally, many descriptors are invariant to lighting changes. The detection of feature points in this paper is based on the localization of SIFT-features [2], but the generation of the scale space and the detection of extrema is reduced from three to two dimensions.

The relative camera pose is calculated in a similar approach as follows [4]: A set of feature points is extracted in both images and putative correspondences are detected by local descriptor matching. RANSAC [5] is used for a robust calculation of the camera parameters despite wrongly matched feature points. In each iteration of the RANSAC loop a minimum set of points needed for orienting the cameras is selected (5 points for calibrated cameras [6]). All other point correspondences are classified as inliers or outliers according to the estimated fundamental matrix. The camera parameters which returned the highest number of inliers are selected and all inliers are used for a final optimization of the parameters with a least squares solution.

Line Matching. The goal of line matching algorithms is to find corresponding line segments between two or more images. Schmid and Zisserman [7] use known epipolar geometry for establishing point-to-point matches along line segments. Cross-correlation scores are computed for all point correspondences and their average is used as value for the line similarity. Cross correlation is adapted to large camera movements by approximating the areas next to line segments by planes and finding the best fitting homographies. This approach cannot be used for the camera orientation problem because camera parameters would have to be known in advance.

Bay et al. [8] combine color information and topological relations for matching line segments. An initial set of potential line matches is created by comparing the color histograms of stripes next to the line segments. As these line descriptors are not very distinctive, it is necessary to filter wrong matches based on topological relations. The topological filter takes three line segments or a combination of line segments and points and compares their sidedness in both images.

Meltzer and Soatto [9] match arbitrary edges across images instead of straight line segments with a similar approach to ours. They select key points at extremal values in the Laplacian of Gaussian along the edge. The feature descriptors for these points are based on gradient histograms similar to SIFT-points. The feature points are matched with a dynamic programming approach that uses the ordering constraint, i.e. corresponding feature points appear in the same order since projective transformations are order preserving. This constraint is used implicitly in our algorithm because a 1D homography maintains the order of points.

1D Point Correspondences. Scale-space features in 1D signals have been used in other applications. Briggs et al. [10] match scale-space features in one-dimensional panoramic views. The images are taken by an omnidirectional camera used for robot navigation. A simple feature descriptor based on the value and curvature in the DoG-space is used for matching points which is sufficient for small camera baselines. The features are matched by circular dynamic programming, which exploits the fact that corresponding features in the circular images have to appear in the same order.

Xie and Beigi [11] use a similar approach for describing 1D sensor signals measured from human activities. The feature descriptors include the neighboring extremal positions of a key point. Corresponding points are found by nearest-neighbor matching.

2 Feature Extraction

In this section we describe how scale-invariant features along salient line segments are extracted from an image. The first step is to detect lines or line segments in an image, for which standard methods can be used. The next step is to create one-dimensional intensity profiles along the extracted lines. Finally, the scale spaces of these profiles are used to detect stable feature points.

2.1 Line Extraction

Although extracting lines from an image is not the main aspect of this paper, we want to depict some details about the line segments we use for matching. We use images that have been undistorted in a preprocessing step in order to contain straight lines. The parameters for undistortion can be computed together with the intrinsic camera calibration. For uncalibrated cameras, it is possible to use other undistortion methods, e.g. with a line-based approach [12].

Architectural scenes often contain collinear line segments, e.g. along horizontally or vertically aligned windows. Collinear line segments which correspond to the same 3D line in world space induce the same homography on a 1D line camera when they are matched to lines in another image. Therefore it is useful to extract lines across the whole image to get more robust homographies and point matches. On the other hand, parts of the line not corresponding to edge features in the image should not contribute to the matching process, because

it is possible that they are occluded by other objects. For this reason, we use one line segment for all collinear segments and apply weights according to the underlying image edge.

We use a Canny edge detector for producing an edge image and then apply a Hough transform for detecting straight lines. We sample the image along the lines with a fixed step size of one pixel and compute a weight for each sample point. The weights are based on the image gradients and denote the probability that a sample point is part of an image edge. The calculation of weights is defined in Equation 1 where g_i is the gradient at the sample point i and n is the normal of the line. The cosine of the angle between the gradient and the line normal is used to exclude sample points where another edge crosses the current line.

$$w_i = \|g_i\| \cdot \left| \frac{g_i}{\|g_i\|} \cdot n \right| = |g_i \cdot n| \quad (1)$$

For efficiency, low-weighted parts at the ends of a line segment are cut off and not used in subsequent operations. The rest of the line is kept for detecting and matching feature points. The weights are used for decreasing the impact of matched feature points at sample points with no underlying image edge.

2.2 1D Line Profiles

Straight lines can occur at edges within an object or at the border of an object. In the first case both sides of a line are quite stable with respect to each other. However, in the second case the images are only consistent on the side of the object, whereas the background changes with the viewpoint due to parallax. Therefore we investigate the two sides of the line separately during feature detection and matching.

For each side of a line, the image is sampled with the same step size as the weights in Section 2.1. In order to get a more descriptive representation we do not sample a single line but a rectangular extension of the line by a width w into the respective direction. This sampled rectangle is collapsed into a 1D profile by averaging the intensity values for more efficient subsequent computations.

An important parameter is the width w . It has to be large enough to contain distinctive information, i.e. it has to contain image parts next to the edge itself. Otherwise it is not possible to distinguish between noise and image features. If it is too large, on the other hand, multiple features may be collapsed and the one-dimensional profile is very smooth. Furthermore, the corresponding image parts next to a line segment may have differing widths in the case of large projective distortion. We used a profile width of 40 pixels in our experiments, but this parameter clearly depends on image resolution and scene contents.

2.3 Scale Space Analysis

For each line profile a Gaussian scale-space is created in a similar manner as for two-dimensional SIFT-features [10,2]. The one-dimensional signal $I(x)$ is convolved with a Gaussian filter $G(x, \sigma)$ over a range of different scales σ :

$$S(x, \sigma) = G(x, \sigma) * I(x), \text{ with } G(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} \quad (2)$$

Stable feature points are located at extrema in the difference-of-Gaussian (DoG) scale-space $D(x, \sigma)$. The difference-of-Gaussian is an approximation of the Laplacian of Gaussian and can be computed as the difference of two nearby Gaussian scales separated by a constant factor k :

$$D(x, \sigma) = S(x, k\sigma) - S(x, \sigma) \quad (3)$$

The creation of the scale space is implemented efficiently with multiple octaves where the profile is resized to half of its resolution for each subsequent octave. The scale factor σ of neighbored scales is separated by a constant factor $k = 2^{1/s}$ where s is the number of scales per octave. This means that the scale factor σ is doubled within one octave. In each octave $s + 3$ Gaussian scales are created in order to use s DoG scales for extremal value detection. The scale space creation is depicted in Figure 1 and an example can be seen in Figure 2.

Potential feature points are extracted at positions where the value in the DoG-space is higher or lower than its eight surrounding values. The exact position

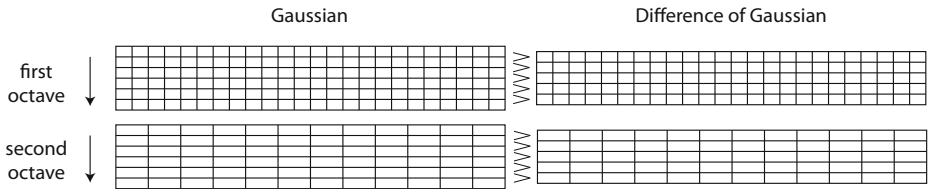


Fig. 1. Generation of Gaussian and DoG scale space for 3 scales per octave



Fig. 2. From top to bottom: sampled rectangle next to a line segment, collapsed profile, Gaussian scale space with 6 octaves and 5 scales per octave, DoG scale space, extrema (white and black dots) in DoG scale space, accepted feature points with refined positions in DoG scale space

of an extremum is determined by fitting a quadratic function using the Taylor expansion of the scale-space $D(x, \sigma)$ [13]. The DoG-value at the extremum is used to reject unstable extrema with low contrast. Extrema with small curvature are also rejected as unstable. Similar to Briggs et al. [10] we calculate the curvature as the geometric mean of the second derivatives $c = \sigma \sqrt{|d_{xx}d_{\sigma\sigma}|}$. The multiplication with σ is necessary to get a scale-invariant curvature value.

3 Feature Matching

Feature matching is split into two parts: The first part matches feature points between two image lines. The second part searches for corresponding lines based on the feature point correspondences and epipolar constraints.

As the orientation of the line segments is unknown, it is necessary to match each side of a line to both sides of the second line. In addition, the feature point descriptors themselves have to be invariant to the orientation of the underlying line. If a large number of line segments have to be matched or if time efficiency is important, it may be better to orientate all line segments such that the brighter profile is on the left side of the line [8]. The number of comparisons is reduced from four to two and the descriptor does not have to be changed. This approach fails if a line segment is located at the boundary of an object and the background contains large intensity changes.

3.1 Feature Point Descriptor

During feature point matching corresponding points between two line profile should be detected. Therefore, we need a matching score that is a good estimate of the probability that two scale space features correspond to the same physical point in the world. Extrema of different types, i.e. minima and maxima, cannot correspond to the same object and a matching score of zero is assigned to these pairs.

Local properties used by Briggs et al. [10] were not discriminative enough in our experiments. Especially in the case of repetitive patterns, e.g. multiple windows of a building along a line, it was not possible to distinguish between correct and wrong correspondences.

We include the neighborhood of a feature point in order to increase the stability of the descriptor. Although the neighborhood can be quite different in case of large projective distortions or occlusions, it is a good description for many cases. The Gaussian scale space is sampled at neighboring points to the left and right of the feature point. The step size between sample points is based on the scale of the feature point in order to get a scale-invariant feature descriptor. The matching score between two features is computed based on the sum of squared distances between corresponding neighboring samples. The matching scores allow to narrow down the set of potential feature matches, but it is still not discriminative enough to extract valid matches directly.

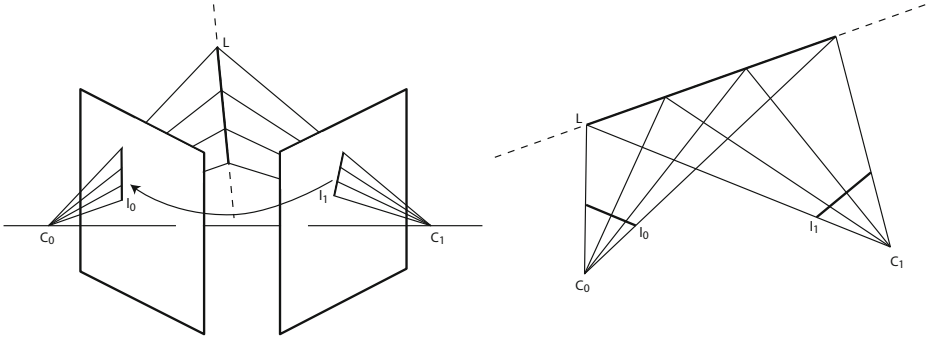


Fig. 3. Transformation from 2D-cameras to 1D-cameras. A 3D line segment L is projected onto two images with camera centers C_0 and C_1 . The cameras are rotated such that L , l_0 and l_1 are located in a plane. In the right image can be seen that a 1D homography maps all points on line l_0 to their corresponding points on the second line l_1 .

3.2 1D Homographies

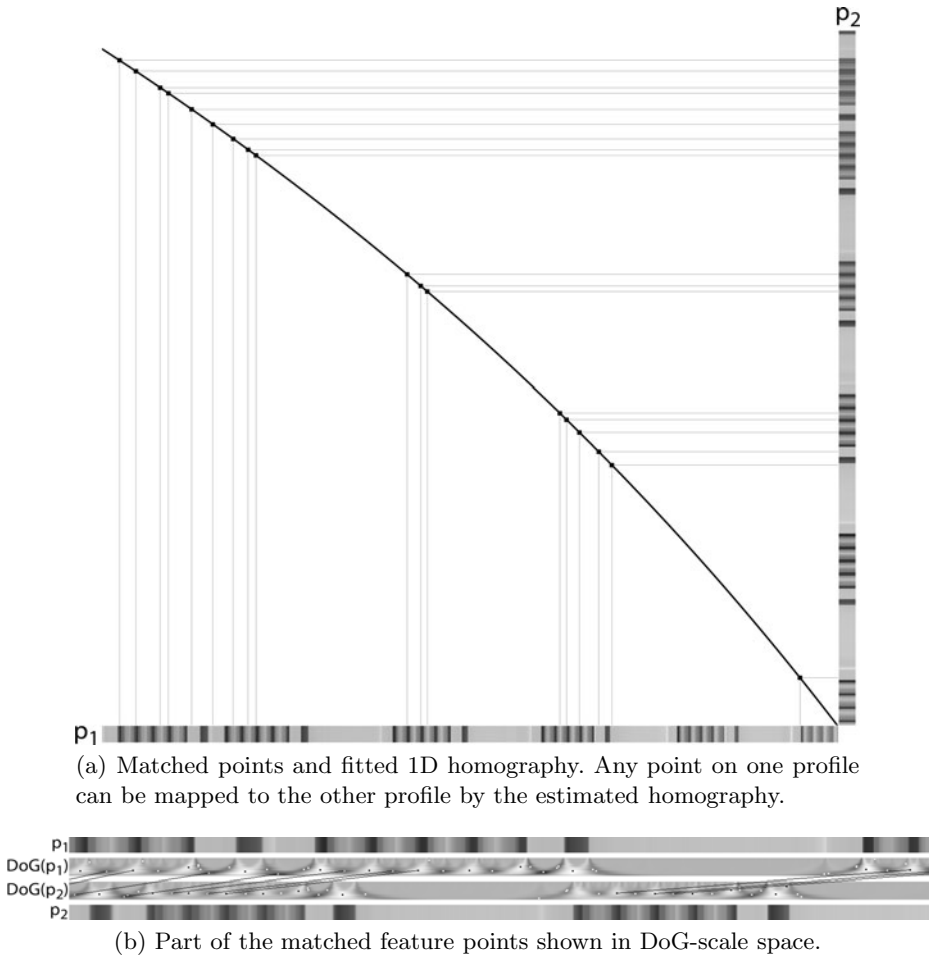
Figure 3 shows the projection of a line in 3D space onto two images. The relation between the corresponding image lines \mathbf{l}_0 and \mathbf{l}_1 is not altered by rotations of the cameras around the 3D line \mathbf{L} . We rotate one camera such that the 3D line \mathbf{L} and the image lines \mathbf{l}_0 and \mathbf{l}_1 are located in one plane. With this transformation the point matching problem is reduced to 1D cameras and it can be easily seen that corresponding points on the image lines are correlated by a one-dimensional homography. The 2×2 -matrix \mathbf{H} maps homogeneous points \mathbf{x}_i on the first image line to the corresponding points \mathbf{x}'_i on the second image line:

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \quad (4)$$

This one-dimensional homography provides an important constraint on the set of corresponding points provided that two line segments belong to the same straight line in 3D space. The constraint that corresponding points appear in the same order is implicitly satisfied by the homography if the line segment is in front of both cameras. A minimum of three points is needed for the calculation of the homography, e.g. with the direct linear transformation (DLT) algorithm [14].

We use RANSAC [5] for the robust estimation of the homography. An initial set of potential point matches is generated by taking the N best correspondences for each feature point based on the feature descriptor presented in Section 3.1. In each iteration of the RANSAC loop three potential point matches are selected for the computation of a 1D homography. All other point correspondences are classified as inliers or outliers according to their symmetric transfer error e based on the absolute difference d (Equation 5).

$$e = d(\mathbf{x}, \mathbf{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathbf{H}\mathbf{x})^2 \quad (5)$$



(a) Matched points and fitted 1D homography. Any point on one profile can be mapped to the other profile by the estimated homography.

(b) Part of the matched feature points shown in DoG-scale space.

Fig. 4. Feature point matches between two line profiles

Finally, the resulting homography is optimized to all inliers and additional inliers from the other point matches are sought. Figure 4 shows an example for point correspondences between two line profiles and the associated homography.

A pair of corresponding lines has to fulfill two constraints in order to be accepted as line match. The first constraint is a minimum number of matched feature points. The second constraint tests how well the line profiles fit to the estimated 1D homography. Densely sampled points on the first line are transformed to their corresponding coordinate in the second line and vice versa. The sum of squared distances between the intensity values in the line profiles at corresponding sample points is used to measure the quality of the homography. The squared distances are multiplied by the weights obtained from the edge response in Section 2.1. The weighting is necessary to avoid contributions from image parts not belonging to the same 3D line, e.g. because the line is occluded by another object.

3.3 Camera Orientation

In the previous section a set of line matches together with point matches along these lines have been extracted. The point matches obey a one-dimensional homography, but the line matches do not fulfill any epipolar constraints yet. It is possible that there are wrongly matched line segments and that one line is matched to multiple lines in the other image.

Consistent line matches are extracted by computing the relative pose between the two images. In the calibrated case, two matched line segments together with their 1D homographies are required to compute the relative pose. We use the five-point-algorithm [6], for which three points are taken from the first line correspondence and two points from the second.

As there is usually only a rather small set of line matches, all potential line matches can be evaluated exhaustively. Of course a RANSAC approach could be used again to speed up computations if necessary. The relative camera orientation is computed for a pair of line matches. For all line matches 3D points are triangulated based on the estimated camera parameters for all point correspondences along the line segments. Line matches are classified as inliers respectively outliers depending on the reprojection errors of these 3D points. Additionally, it is evaluated if the 3D points are located on a three-dimensional line. In order to avoid degenerate cases, a line can only be classified as inlier once, although it might appear in multiple line matches. After evaluating all test cases, the relative orientation that led to most inliers is selected.

4 Experiments

We report results on four different example scenes. All images were taken with a *Canon EOS 5D Mark II* and have a resolution of 5616×3744 pixels. The scale spaces were created with four octaves and four scales per octave. The feature descriptors were created with 80 neighboring sample points.



Fig. 5. Facade scene: 365 point correspondences in four lines. Matching lines are displayed with the same color, unmatched lines are shown in white. The final point correspondences are visualized as black rings. Note that some lines are collapsed because they are located very near to each other.

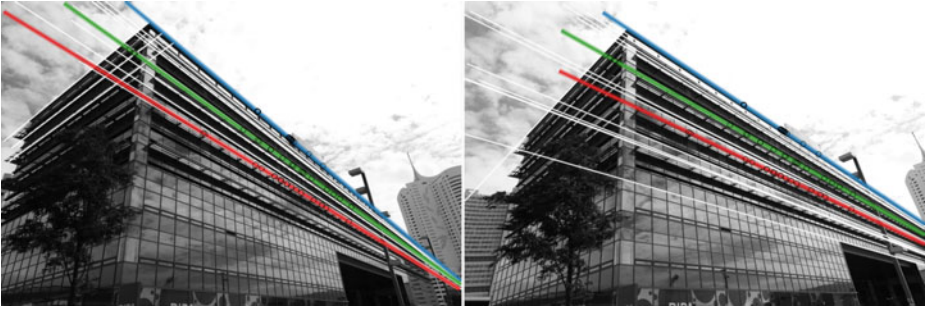


Fig. 6. Building scene: 52 point correspondences in three lines

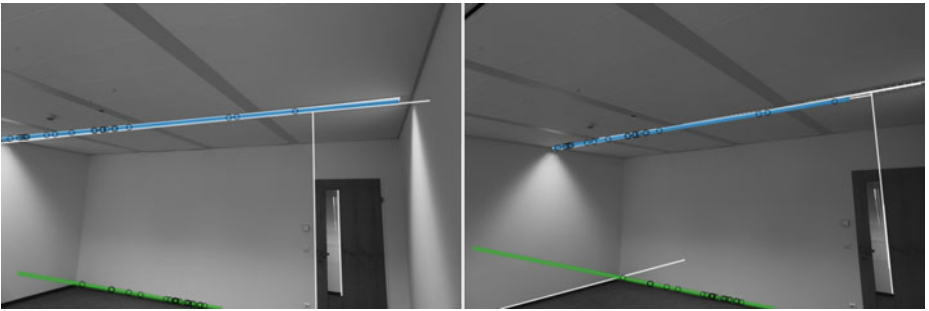


Fig. 7. Interior scene: 41 point correspondences in two lines

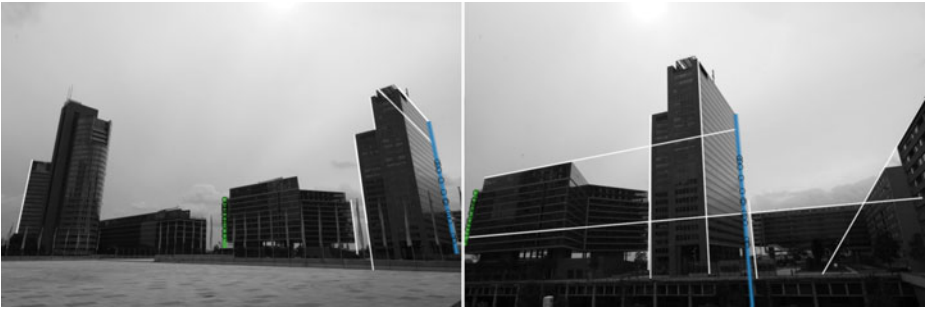


Fig. 8. Urban scene: 28 point correspondences in two lines

The images of the first example show the facade of a house (Figure 5). Ten line segments were extracted in each images, from which correspondences between five line profiles were found initially. Four of these initial matches were the left and right profile of one corresponding line. The relative camera pose approved four line matches and 365 point matches.

Figure 6 shows the pictures of another building. The initial line matching created three line correspondences which were all validated with 52 point

correspondences by camera pose estimation. The problems in this scene are that there are many similar lines and that there are different occlusions from different viewpoints.

Figure 7 shows an interior scene. Ten line segments were extracted in both images. Two line correspondences and 41 point correspondences were validated by camera pose estimation from ten initial line profile matches.

The results for an urban scene can be seen in Figure 8. Ten line segments were extracted in both images, from which two line correspondence and 28 point correspondences were found and validated. This example shows that the algorithm is capable of matching lines at different scales.

5 Conclusion and Outlook

We have presented a new method for matching points located on line segments. The application of a one-dimensional homography allows to compute globally consistent point correspondences along the line segments. The set of corresponding points can be used together with a dense matching score for detecting corresponding line segments between the images. The set of potential line segments is evaluated based on the robust calculation of the relative pose.

We showed that the dimensionality of feature matching can be reduced by splitting it into point matching along line segments and line matching using epipolar constraints. The advantage of our algorithm is that feature points can be found although only a few distinctive 2D features are present in the images, provided that straight lines can be extracted.

For future work, we would like to use the estimated 1D homographies directly for calculating the relative pose between the input images. Two line correspondences and the associated 1D homographies could be used for estimating an initial solution to the camera pose problem. Another improvement of the algorithm will be a pre-selection of potential line matches before corresponding feature points are matched. This initial matching should be based on a simple line descriptor, e.g. a color histogram, and will increase the time efficiency of the algorithm.

References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1615–1630 (2005)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
3. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *British Machine Vision Conference*, pp. 384–393 (2002)
4. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: *SIGGRAPH Conference Proceedings*, pp. 835–846 (2006)
5. 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)

6. Nistér, D.: An efficient solution to the five-point relative pose problem. *IEEE Pattern Analysis and Machine Intelligence* 26, 756–770 (2004)
7. Schmid, C., Zisserman, A.: Automatic line matching across views. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pp. 666–672 (1997)
8. Bay, H., Ferrari, V., Van Gool, L.: Wide-baseline stereo matching with line segments. In: *Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition*, pp. 329–336 (2005)
9. Meltzer, J., Soatto, S.: Edge descriptors for robust wide-baseline correspondence. In: *Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
10. Briggs, A.J., Detweiler, C., Li, Y., Mullen, P.C., Scharstein, D.: Matching scale-space features in 1d panoramas. *Computer Vision and Image Understanding* 103, 184–195 (2006)
11. Xie, J., Beigi, M.S.: A scale-invariant local descriptor for event recognition in 1d sensor signals. In: *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, Piscataway, NJ, USA*, pp. 1226–1229. IEEE Press, Los Alamitos (2009)
12. Thormählen, T., Broszio, H., Wassermann, I.: Robust line-based calibration of lens distortion from a single view. In: *Proceedings of MIRAGE 2003*, pp. 105–112 (2003)
13. Brown, M., Lowe, D.: Invariant features from interest point groups. In: *British Machine Vision Conference*, pp. 656–665 (2002)
14. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518

Near-Optimal Selection of Views and Surface Regions for ICP Pose Estimation

L.H. Mark¹, G. Okouneva¹, P. Saint-Cyr¹, D. Ignakov¹, and C. English²

¹ Ryerson University, Toronto, Canada

² Neptec Design Group Ltd, Ottawa, Canada

Abstract. This paper presents an innovative approach for the selection of well-constrained views and surface regions for efficient ICP pose estimation using LIDAR range scanning. The region selection is performed using the Principal Component Analysis technique with derived predictive indices that can be used to assess a view/region for pose estimation. Localized scanning has been proposed for spacecraft rendezvous operations, particularly in the “last mile” scenario where whole object scanning is not possible. The paper illustrates the PCA approach for selection of optimal scanning views and localized regions using (a) CAD models of several spacecraft structures with supporting simulation results based on large amount of data, and (b) a model of a faceted shape, cuboctahedron, which was scanned using Neptec’s TriDAR laser scanner. The results confirm the hypothesis that the selected views or regions deliver accurate estimates for the pose norm and also for each component of the pose.

1 Introduction

Pose estimation is a fundamental task in computer vision. A range-finding scanner samples an object’s surface to produce a cloud of data points which is then used in an iterative algorithm to compute the position and orientation, pose, of the object. Such algorithms generally seek to minimize a cost function that quantifies the registration error between a model and the corresponding data points. Our interest lies in a particular class of pose estimation where the exact locations of the LIDAR-sampled points are unspecified, and are gathered evenly across a visible area of the object from a single viewpoint. A continuous surface model of the object, generally assumed to be a triangulated surface mesh model, is registered against the data point set to produce a pose estimate for the object. Our focus is on the potential terminal accuracy of the Iterative Closest Point (ICP) algorithm in the context of imperfect input data, and on the sensitivity of the pose solution in the vicinity of the true pose to the inevitable presence of noise-like error. The practical application of the present work supports the use of LIDAR based computer vision for spacecraft rendezvous operations [1], [2] and [3]. For the situation depicted in Figure 1, the approaching spacecraft scans predetermined “feature areas” of the International Space Station in order to determine its relative position for docking procedures.

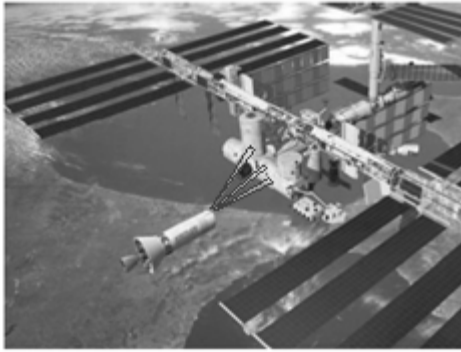


Fig. 1. LIDAR-assisted spacecraft rendezvous

Geometric constraint analysis is an application of Principal Component Analysis. It examines the sensitivity of shape registration error to variation in the model's pose, providing a powerful way of assessing the expected accuracy of iterative registration algorithms. In this paper, we use the term registration to refer to the cost computed when comparing the alignment of model and data. Pose estimation, then, is the process of minimizing the registration cost to determine the best pose. Both ICP [4] and constraint analysis share the same attractive feature of being based on the bulk calculation of data, avoiding feature (primitives and local shape invariants) detection tasks. Simon [5] introduced the application of constraint analysis to the ICP process of pose estimation in computer vision. In his work, constraint analysis was used to optimize the selection of target points on human bones for scanning during radiation therapy. The optimization was based on the use of Noise Amplification Index (NAI) [6] where larger NAI values were shown to correspond to smaller values of the norm of the pose error. While Simon uses a sparse set of key points for the bone problem, Shahid & Okouneva [7] used the same form of discrete-point constraint analysis applied to point-clouds collected from the uniform projection of points onto "windowed" areas of spacecraft objects in order to identify optimal local scan areas for pose estimation. Another related paper by Gelfand et al. [8] considers sampling strategies over an area to produce an effective set of points for use by ICP. A more recent paper by McTavish & Okouneva [9] generalizes the concept of discrete-point self-registration to a surface integral-based self-registration referring to its use for pose estimation assessment and view selection as Continuum-Shape Constraint Analysis (CSCA). The authors account for the directional nature of a single scan by incorporating a view factor into the CSCA cost matrix calculation. The result is a cost matrix that is a well-defined property of the shape geometry and also dependent on view direction. In this case, constraint analysis was used to establish whole-object views for accurate pose estimation.

The goal of this paper is to further demonstrate the use of the constraint analysis approach for intelligent selection of views and localized surface regions of objects with complex geometry for LIDAR scanning and consequent ICP pose estimation. In particular, it is shown that constraint analysis indices such as the Noise Amplification Index and inverse condition number can be used for view/region selection, and the

accuracy of all pose components increases as the values of these indices increase. These indices were chosen because they provide upper boundaries on the absolute norm and relative norm of the pose error. Other proposed indices are mostly related to the shape and volume of the error ellipsoid (see [6] for details). The presented constraint analysis approach is supported by a large amount of simulated ICP runs performed using models of spacecraft structures such as orbiter Shuttle, module Airlock of the International Space Station and telescope Hubble. The approach is also supported by ICP results for the model of a cuboctahedron which was scanned with Neptec's TriDAR laser scanner.

2 Constraint Analysis Indices and ICP Pose Estimation

This section summarizes the basic concepts of pose estimation and constraint analysis indices as indicators of ICP performance. Comparing to [7], we derive the relations between norm the pose error and constraint analysis indices in a more generalized form. We start with two sets of N points: $\{\mathbf{d}_i = (d_{ix}, d_{iy}, d_{iz})^T\}$ are data points, and $\{\mathbf{m}_i = (m_{ix}, m_{iy}, m_{iz})^T\}$ are model points with the normals $\{\mathbf{n}_i^m\}$, $i=1, \dots, N$. Data points can be thought as points measured by a rangefinder scanner, and model points as points of a known CAD model. The goal of pose estimation is to find the rotation and translation which align the data points with the model points as shown in Figure 2. The most commonly used pose estimation algorithm is the Iterative Closest Point (ICP) algorithm introduced in [4]. ICP iteratively seeks a rigid body transformation, a rotation matrix \mathbf{R} and a translation vector $\mathbf{t} = (t_x, t_y, t_z)^T$, which minimizes the sum of squared distances from \mathbf{d}_i to the plane tangent to the model at the point \mathbf{m}_i :

$$E = \sum_{i=1}^N \left((\mathbf{R}\mathbf{d}_i + \mathbf{t} - \mathbf{m}_i) \mathbf{n}_i^m \right)^2 \rightarrow \min \quad (1)$$

At the final stage of ICP, when rotations are small, the displacement of the point \mathbf{d}_i can be represented as $(\boldsymbol{\omega} \times \mathbf{d}_i + \mathbf{t})^T$, where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ is a vector of small rotations around X , Y and Z axes. The minimization problem (1) then becomes:

$$E = \sum_1^N \left((\mathbf{d}_i - \mathbf{m}_i) \mathbf{n}_i^m + \boldsymbol{\omega} (\mathbf{d}_i \times \mathbf{n}_i^m) + \mathbf{t} \mathbf{n}_i^m \right) \quad (2)$$

To solve for the differential transformation $\mathbf{x} = (\omega_x, \omega_y, \omega_z, t_x, t_y, t_z)^T$, equate the partial derivatives of (2) with respect to all components of \mathbf{x} to zero. The resulting linear (6 x 6) system has the form:

$$\mathbf{J}\mathbf{J}^T \mathbf{x} = \mathbf{J}\mathbf{f} \quad (3)$$

where $\mathbf{J} = \begin{pmatrix} \mathbf{d}_1 \times \mathbf{n}_1^m & \dots & \mathbf{d}_N \times \mathbf{n}_N^m \\ \mathbf{n}_1^m & \dots & \mathbf{n}_N^m \end{pmatrix}$ is a (3N x 6) matrix and $\mathbf{J}\mathbf{f}$ is the 6-vector

representing the residuals. The covariance matrix $Cov = \mathbf{J}\mathbf{J}^T$ can be considered a sensitivity matrix for small displacements of data points. The system (3) represents the system of normal equations $\mathbf{J}^T \mathbf{x} = \mathbf{f}$. The solution of (3) minimizes the norm

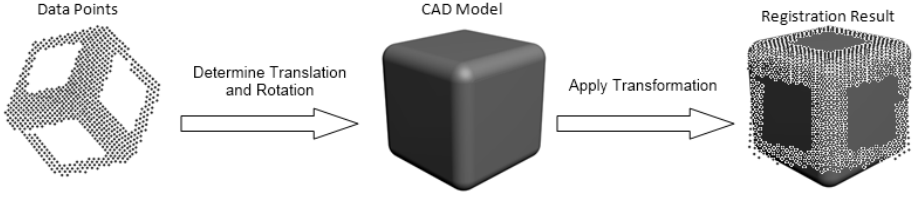


Fig. 2. Pose estimation process registering a point cloud to a CAD model

$\|\mathbf{J}^T \mathbf{x} = \mathbf{f}\|$, or the norm $\|\mathbf{J}^T \mathbf{x}\|$, as \mathbf{f} is small at the final stage of alignment. The vector $\mathbf{J}^T \mathbf{x}$ represents the total amount by which the point-to-plane distances for all points will change if the points are moved by the differential transformation \mathbf{x} .

The following inequalities are valid for the norms:

$$\sigma_6 \leq \frac{\|\mathbf{f}\|}{\|\mathbf{x}\|} \leq \sigma_1 \quad \text{and} \quad \frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\sigma_1}{\sigma_6} \frac{\|\delta \mathbf{f}\|}{\|\mathbf{f}\|} \quad (4)$$

where $\delta \mathbf{x}$ and $\delta \mathbf{f}$ represent the noise on the vector \mathbf{x} as the result of the noise on the vector of residuals \mathbf{f} , and $\sigma_1 \geq \dots \geq \sigma_6$ are ordered singular values of \mathbf{J}^T . The inequalities (4) imply

$$\|\delta \mathbf{x}\| \leq \frac{\sigma_1}{\sigma_6} \|\delta \mathbf{f}\| \quad (5)$$

The inequality (5) states that in order to minimize the pose error norm $\|\delta \mathbf{x}\|$, we have to minimize the ratio σ_1/σ_6^2 or maximize the inverse ratio σ_6^2/σ_1 which is called the *Noise Amplification Index (NAI)*:

$$\|\delta \mathbf{x}\| \leq \frac{1}{NAI} \|\delta \mathbf{f}\|, \quad \text{and} \quad NAI = \frac{\sigma_6^2}{\sigma_1} \quad (6)$$

The ratio σ_1/σ_6 is called the condition number *Cond*. The inequality (4) states that to minimize the relative pose error, the condition number must be minimized. We will also consider the inverse condition number $InvCond = 1/Cond$ which, according to the inequality (4) $\frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{1}{InvCond} \frac{\|\delta \mathbf{f}\|}{\|\mathbf{f}\|}$, must be maximized to achieve minimal relative pose error. The squared norm $\|\mathbf{f}\|^2$ is the misalignment error \mathbf{E} which can be represented as $\mathbf{E} = \|\mathbf{f}\|^2 = \mathbf{x}^T \mathbf{J} \mathbf{J}^T \mathbf{x}$.

Constraint analysis of *Cov* can be used as a mathematical tool as it reflects the property of a object's surface to be constrained or unconstrained in a certain direction. PCA performs the decomposition of *Cov* as:

$$\mathbf{J} \mathbf{J}^T = (\mathbf{q}_1 \dots \mathbf{q}_6) \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_6 \end{pmatrix} \begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_6^T \end{pmatrix} = \mathbf{Q} \mathbf{S} \mathbf{Q}^T \quad (7)$$

Then, the misalignment error is represented as $E = \mathbf{x}^T \mathbf{Q} \mathbf{S} \mathbf{Q}^T \mathbf{x} = \sum_{i=1}^6 \lambda_i (\mathbf{x}^T \mathbf{q}_i)^2$,

where \mathbf{q}_i are the eigenvectors of Cov corresponding to the eigenvalues λ_i . Note, that the singular values σ_i^J and $\sigma_i^{J^T}$ of the matrices \mathbf{J} and \mathbf{J}^T are related to the eigenvalues of Cov as: $\sigma_i^J = \sigma_i^{J^T} = \sqrt{\lambda_i} = \sqrt{\sigma_i^{JJ^T}}$.

The geometry of the surface plays an important role in the convergence of ICP to the global solution. For example, two planes can slide with respect to each other without changing the error (2), or, two spheres can rotate with respect to each other with no effect on the error if the reference frame is at the center of the spheres. In these situations, we say that the surface is unconstrained in certain directions: planes are unconstrained in translation directions, and spheres are unconstrained in rotations.

The eigenvector \mathbf{q}_1 corresponding to the eigenvalue λ_1 represents “the transformation of the maximum constraint”. Perturbing the points by \mathbf{q}_1 will result in the largest possible change in \mathbf{E} from among all possible transformations. Similarly, the differential transformation in the direction of \mathbf{q}_6 corresponds to the transformation of the maximum freedom and will result in the smallest possible error.

As the number of points increases, the error \mathbf{E} also increases. Therefore, it is natural to consider a normalized error $\mathbf{E}_{norm} = 1/N \mathbf{E}$ and normalized $NAI_{norm} = 1/\sqrt{N} NAI$. These indices can be used for selection of object’s views for efficient pose estimation. Views with high values of the indices deliver statistically better pose estimates than views with low values.

3 Experimental Results

In this Section, we present:

- a) Simulated results on statistical relations between ICP accuracy and NAI_{norm} and $InvCond$ values. The inequalities (4) and (6) establish the upper boundaries on the norm of the pose error. Therefore, high values of the indices correspond to lower values of the norm and relative norm of the pose error. This relation is demonstrated using large amount of data gathered for the models of the orbiter Shuttle, module Airlock and telescope Hubble.
- b) Results on the relations between individual pose components and indices NAI_{norm} and $InvCond$. For autonomous rendezvous, it is more important to be able to assess the accuracy of each pose component (degree of freedom) rather than the overall norm.
- c) Results on the relation “pose error norm - NAI_{norm} ” from a) were obtained for the cuboctahedron model which was scanned by Neptec’s TriDAR laser scanner.

3.1 Simulated Results for Spacecraft Structures

The models for the orbiter Shuttle, module Airlock and telescope Hubble are presented in Figures 4, 5 and 6.

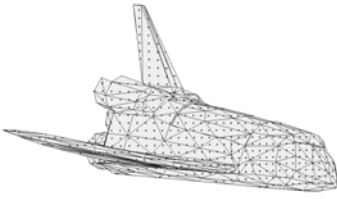


Fig. 4. Orbiter Shuttle

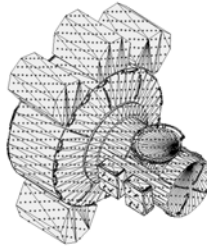


Fig. 5. Airlock

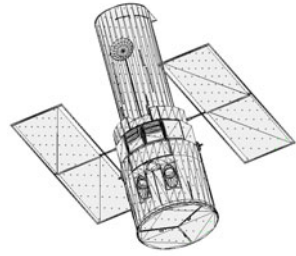


Fig. 6. Telescope Hubble

Figures 7 to 12 present graphs “ NAI_{norm} and $InvCond$ vs. pose error norm”. A thousand randomized views of the models were scanned by a simulated LIDAR. Views contained from 400 to 850 scan points. Each point on the graphs represents an average of 100 scans, each with a random Gaussian noise added to locations of the points on the surface.

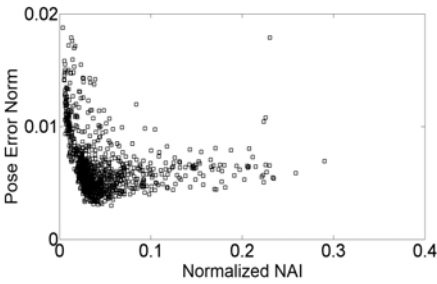


Fig. 7. NAI_{norm} vs. pose error norm for Shuttle

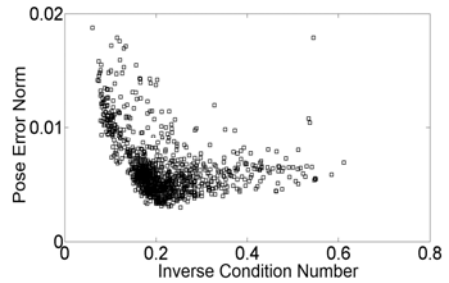


Fig. 8. $InvCond$ vs. pose error norm for Shuttle

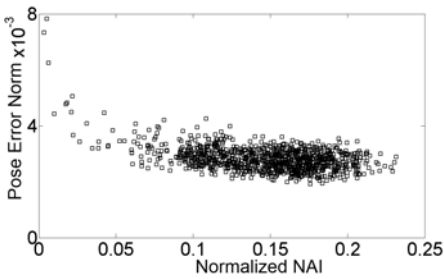


Fig. 9. NAI_{norm} vs. pose error norm for Airlock

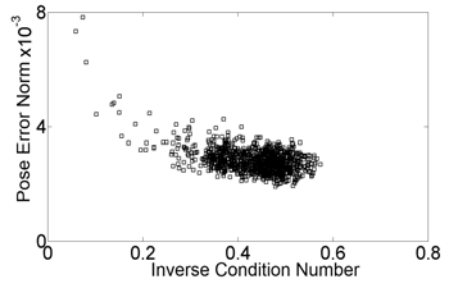


Fig. 10. $InvCond$ vs. pose error norm for Airlock

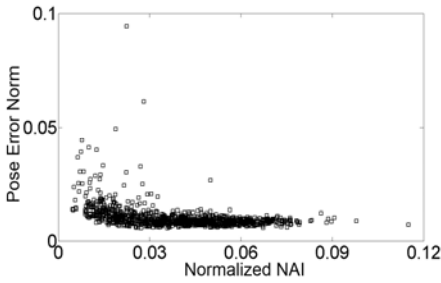


Fig. 11. NAI_{norm} vs. pose error norm for Hubble

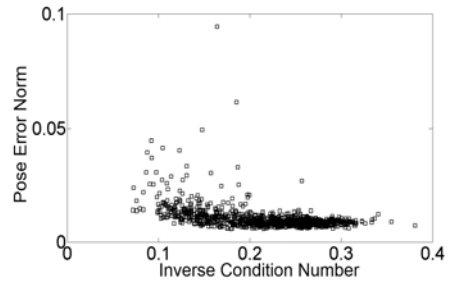


Fig. 12. $InvCond$ vs. pose error norm for Hubble

The above graphs show that both NAI_{norm} and $InvCond$ can be used to select views which have the accurate pose. Both indices show a similar behavior. The increase of both indices happens simultaneously due to the increase of the minimal eigenvalue λ_6 . The maximal eigenvalue λ_1 showed a relative stability across all 1000 views. The example can be seen in Figures 13 and 14.

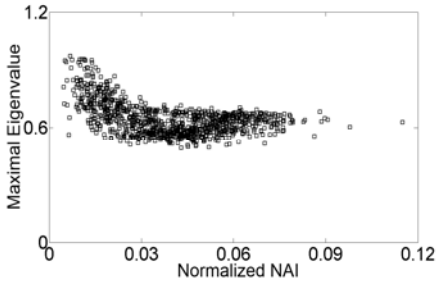


Fig. 13. NAI_{norm} vs. λ_1 for Hubble

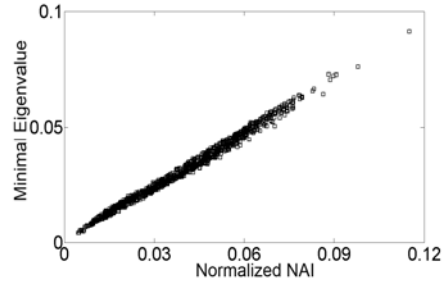


Fig. 14. NAI_{norm} vs. λ_6 for Hubble

Graphs 15 to 20 were constructed using the Hubble telescope and show that NAI_{norm} can also be used to assess individual pose components.

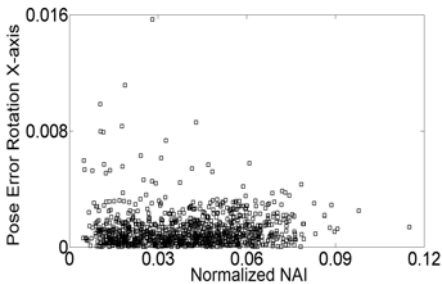


Fig. 15. NAI_{norm} vs. X-axis error for Hubble

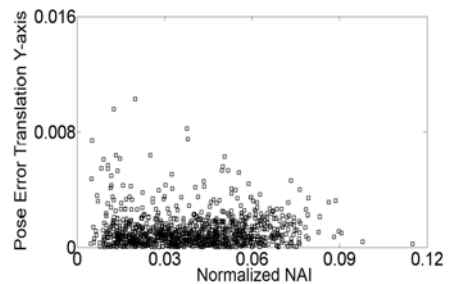


Fig. 16. NAI_{norm} vs. Y-axis error for Hubble

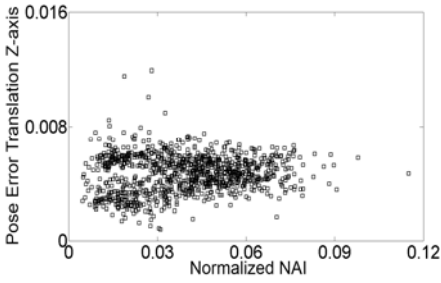


Fig. 17. NAI_{norm} vs. Z-axis error for Hubble

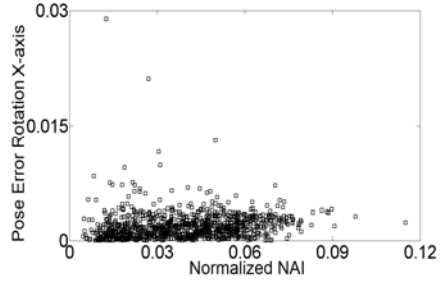


Fig. 18. NAI_{norm} vs. rotation about X-axis for Hubble

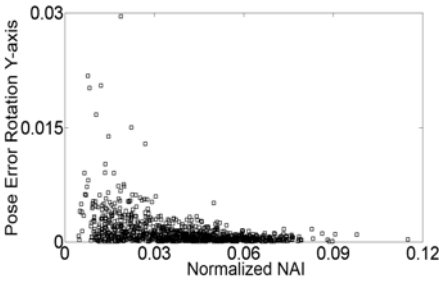


Fig. 19. NAI_{norm} vs. rotation about Y-axis for Hubble

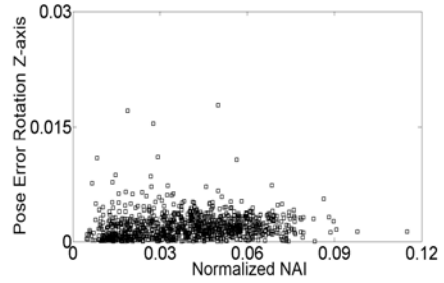


Fig. 20. NAI_{norm} vs. rotation about Z-axis for Hubble

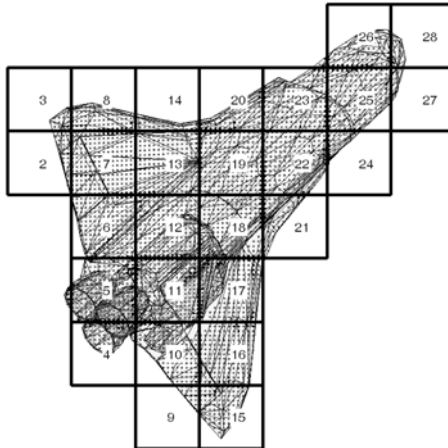


Fig. 21. Window areas for Shuttle model

NAI_{norm} and $InvCond$ can also be used to assess individual areas on the model's surface. Figure 21 shows the areas, windows, for the Shuttle. Figure 22 represents a combined graph which, for each window, and simultaneously shows NAI_{norm} and the Pose Error Norm. It can be seen that if a window has an increased value of NAI_{norm} , then it has a lower pose error and vice-versa. See the case for Window 11 in Figure 23 and Window 20, in Figure 24 is which has a low value of NAI_{norm} and a high pose error.

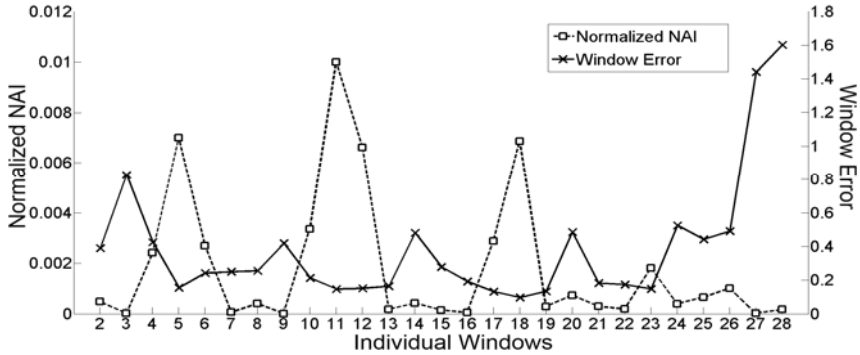


Fig. 22. NAI_{norm} and pose error norm for Shuttle windows

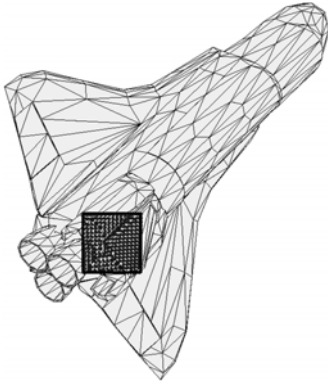


Fig. 23. Shuttle well-constrained window #11

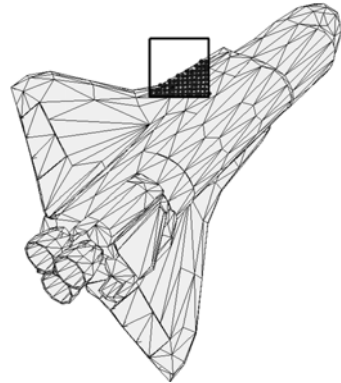


Fig. 24. Shuttle poorly-constrained window #20

3.2 Results for Cuboctahedron Scanned with Neptec's TriDAR Scanner

A model of a cuboctahedron (Figure 25a) was scanned with Neptec's TriDAR scanner. Figure 26 presents a combined graph of NAI_{norm} /Pose Error Norm for each window from Figure 25b. Computation of NAI_{norm} values for real data requires normal vectors. For the point cloud obtained from the TriDAR scanner, the normal for each point based on a plane described by the nearest neighbouring points. The graph shows that NAI_{norm} calculated from real point cloud can also be used to assess regions for pose estimation.

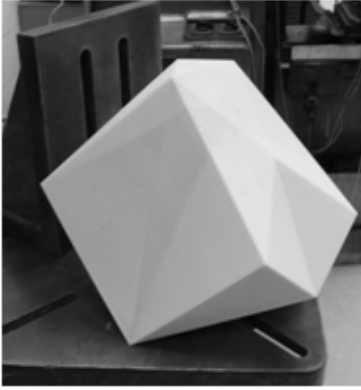


Fig. 25a. Cuboctahedron

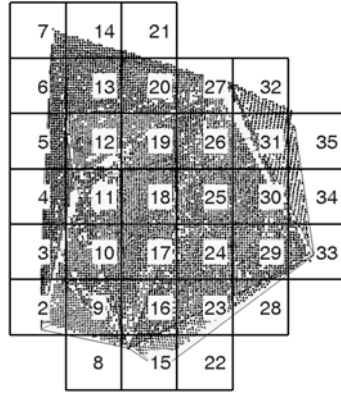


Fig. 25b. Windows for Cuboctahedron

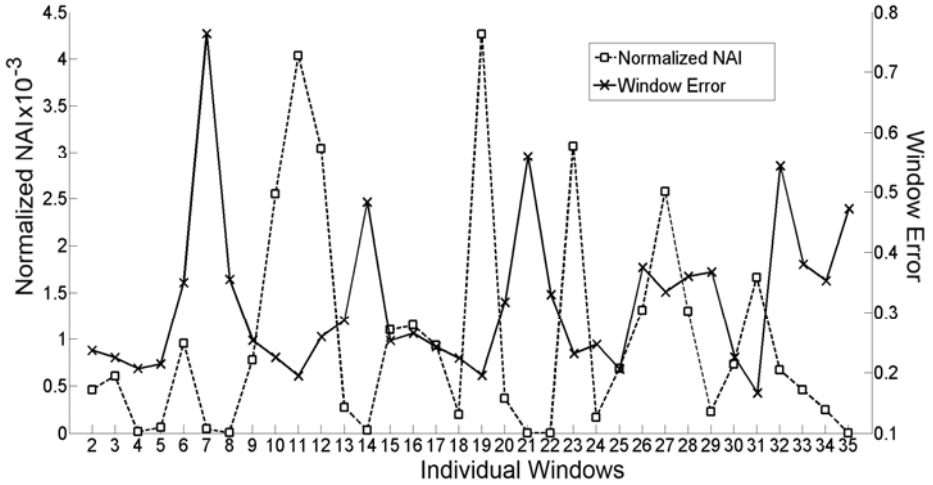


Fig. 26. NAI_{norm} and pose error norm for cuboctahedron windows

4 Conclusion

This paper focuses on application of constraint analysis indices for assessment of object’s views or regions on its surface for pose estimation. The indices considered were the Noise Amplification Index and the inverse condition number. It was shown that both indices can be used to select views or surface regions for efficient pose estimation. Larger values of the indices statistically correspond to lower errors in pose components. The capability to intelligently select views/regions is an important feature of future space vision systems which will be used for autonomous docking and inspections. Such selection can be made on the ground during the mission planning or it can be done in space during the operation using data generated by the scanner point

clouds. The results presented in the paper are supported by large amount of simulated data and data collected using Neptec's TriDAR laser scanner.

An immediate future work related to the topic of the present paper can include research on how to combine several surface's regions to achieve best possible pose estimates.

Acknowledgements

The work presented in this paper was supported by the National Sciences and Engineering Research Council of Canada, Neptec Design Group of Ottawa and the Canadian Space Agency. The authors gratefully acknowledge Mrs. G. Bouchette of Neptec for her tremendous support of testing held at Neptec.

References

1. Samson, C., English, C., Deslauriers, A., Christie, I., Blais, F., Ferrie, F.: Neptec 3D Laser Camera System: From Space Mission STS-105 to Terrestrial Applications. NRC Report 46565, Canadian Aeronautics and Space Journal 50(2) (2004)
2. Allen, A.C.M., Langley, C., Mukherji, R., Taylor, A.B., Umasuthan, M., Barfoot, T.D.: Rendezvous Lidar Sensor System for Terminal Rendezvous, Capture, and Berthing to the International Space Station. Sensors and Systems for Space Applications II. In: SPIE Proceedings, Vol. 6598 (2008)
3. Jasiobedzki, P., Abraham, M., Newhook, P., Talbot, J.: Model Based Pose Estimation for Autonomous Operations in Space. In: Proceedings of the IEEE International Conference on Intelligence, Information and Systems (1999)
4. Besl, P., McKay, N.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
5. Simon, D.A.: Fast and accurate shape-based registration. Carnegie Mellon University, Pittsburgh (1996)
6. Nahvi, A., Hollerbach, J.M.: The Noise Amplification Index for Optimal Pose Selection in Robot Calibration. In: Proc. IEEE Intl. Conf. Robotics and Automation, Minneapolis MN, pp. 22–28 (1996)
7. Shahid, K., Okouneva, G.: Intelligent LIDAR Scanning Region Selection for Satellite Pose Estimation. Computer Vision and Image Understanding 107, 203–209 (2007)
8. Gelfand, N., Rusinkiewicz, S.: Geometrically Stable Sampling for the ICP Algorithm. In: Proc. International Conference on 3D Digital Imaging and Modeling, Stanford, CA, pp. 260–267 (2003)
9. McTavish, D., Okouneva, G., Choudhuri, A.: CSCA-Based Expectivity Indices for LIDAR-based Computer Vision. In: Mathematical Methods and Applied Computing, vol. 1, pp. 54–62. WSEAS Press, Dublin (2009)

Region and Edge-Adaptive Sampling and Boundary Completion for Segmentation

Scott E. Dillard¹, Lakshman Prasad², and Jacopo Grazzini²

¹ Pacific Northwest National Laboratory

² Los Alamos National Laboratory

Abstract. Edge detection produces a set of points that are likely to lie on discontinuities between objects within an image. We consider faces of the Gabriel graph of these points, a sub-graph of the Delaunay triangulation. Features are extracted by merging these faces using size, shape and color cues. We measure regional properties of faces using a novel shape-adaptive sampling method that overcomes undesirable sampling bias of the Delaunay triangles. Instead, sampling is biased so as to smooth regional statistics within the detected object boundaries, and this smoothing adapts to local geometric features of the shape such as curvature, thickness and straightness. We further identify within the Gabriel graph regions having uniform thickness and orientation which are grouped into directional features for subsequent hierarchical region merging.

1 Introduction

Perceptual organization (*aka* grouping and segmentation) is a process that computes regions of the image that come from different objects with little detailed knowledge of the particular objects present in the image [9]. One possible solution to the image segmentation problem, which requires to construct a partition of the image into perceptually meaningful parts, is to perform it subsequently to edge detection [27,16,26,11]. Early works in computer vision have emphasized the role of edge detection and discontinuities in segmentation and recognition [19,18,29]. This line of research stresses that edge detection should be done at an early stage on a brightness, colour, and/or texture representation of the image and segmentation (likewise other early vision modules) should operate later on [17].

An edge detector [30,6] yields a set of pixels which are likely to lie on object boundaries, but the union of these pixels may not form complete boundaries that partition the image [5]. The goal of the segmentation is to then complete the object boundaries by linking pixels together into closed loops [16,25,2]. A well-known solution to this problem is to consider the output of the edge detector as a real-valued function (e.g., magnitude of the image gradient) and then perform the watershed transform [28,12,11]. This produces a so-called *over-segmentation*, a partition of the image that contains too many regions, but from which the desired segmentation can hopefully be obtained by region merging.

A number of methods have been presented which work with a binary edge detector, one which indicates that a pixel is or is not to be treated as part of

an object boundary. One strategy is to employ the Delaunay triangulation of the edge pixels and then select some subset of the triangle edges to complete the object boundaries. Criteria for selecting this subset include region properties such as triangle size and average of color [14,15], or contour properties such as continuity of direction [23,22].

In this paper we improve upon existing methods in two ways. First, we propose a method for shape-adaptive sampling of regional properties (e.g. color) that simultaneously smoothes properties within objects and sharpens them across object boundaries. Second, we incorporate directionality information into the regional properties, not just contour properties, to extract long, straight features of nearly-constant thickness. These two components are incorporated into a pre-segmentation method that parses the image into visually significant regions which are then combined into a hierarchy by merging.

2 Background

We begin with an edge detector, a method for identifying those edge pixels which are likely to coincide with the boundaries of objects depicted in an image. Edge detection is a well-studied problem and an active area of research [20]. We use the Canny detector [3] although we do not rely on any particular properties of this detector and the general strategy presented here is applicable to any detector. For clarity we call the set of points produced by the edge detector *edgels*, whereas a line segment between two such points is an *edge*.

Proximity is paramount among the cues responsible for grouping edgels into salient object contours [7], and so proximity graphs such as the Delaunay triangulation and Gabriel graph provide good candidate edges for completing object boundaries. The *Delaunay triangulation* of a point set $P \subset \mathbb{R}^2$ is a triangulation of the convex hull of P such that the interior of every triangle's circumcircle is disjoint from P . The *Gabriel graph* is a sub-graph of the Delaunay triangulation containing every edge that is a diameter of a circle whose interior is disjoint from P . For uniformity we also refer to this circle as the edge's circumcircle, with circumradius equal to half the edge's length. More complete definitions of these constructions can be found in the text by Goodman & O'Rourke [10].

Let $D_P(x) = \min_{p \in P} \|x - p\|$, the distance from a given point to the closest member point in the set P . Taken as an elevation map, the faces and edges of the Gabriel graph are uniquely identified with peaks and saddle points of $D_P(x)$, respectively [8]. A Gabriel face f contains a point m which is a local maximum of D_P , maximally far from all the nearby points P and hence considered to lie in the middle of the shape defined by P . The value of D_P gives the local width of the shape, which is the radius of the largest circumcircle of any triangle in the Gabriel face. We define the *thickness* of a Gabriel face to be twice this radius, equal to $2D_P(m)$. Figure 1 illustrates a Delaunay triangulation (dashed lines) and the Gabriel graph embedded within it (solid lines.) The circles labeled f and g are the largest that can fit inside their respective Gabriel faces and so define the thickness of those faces. If we consider both faces as parts of the whole shape,

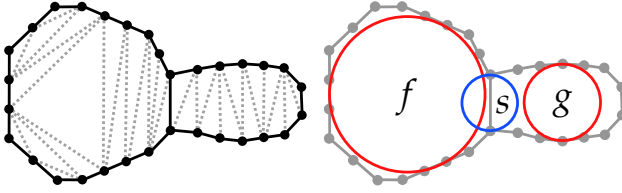


Fig. 1. Left: A Delaunay triangulation (dashed lines) and Gabriel graph (solid lines.) Right: Circles f and g are the circles of greatest radius centered within their respective faces. The distance between the radii of f and s gives the total variation in thickness of the shape.

then the Gabriel edge between them, labeled s , identifies the thinnest part of the shape. We measure the variation in a shape’s thickness by the ratio between it’s maximum and minimum local thickness, in this case f and s respectively.

3 Related Work

Prasad & Skourikhine [22] proposed a method to link edgels into boundaries that begins by grouping them into chains that are adjacent according to an 8-pixel neighborhood connectivity. A system of filters for the Delaunay triangulation of edge chains was formulated, for example, “Do all three triangle vertices belong to different chains?” or “Does a triangle edge connect to chain endpoints?”

The methods proposed by Köthe *et al.* [14] and Letscher & Fritts [15] are similar in that they both were inspired by α -shapes [4], so-called because all Delaunay triangles and edges of circumradius greater than α are removed. Additionally, both groups of authors proposed to use a pair of thresholds, with segmentation regions being “seeded” by triangles having circumradius greater than the larger threshold and region boundaries delineated by edges and triangles with circumradius below the smaller threshold. Their methods differ in how they deal with connected components of triangles with radii between α and β . Köthe *et al.* proved that, under certain assumptions, a topological thinning of the remaining triangles produces an accurate boundary reconstruction. Letscher & Fritts proposed to merge the remaining regions according to color information, with mergers being subject to topological constraints. Both sets of authors advocate using color information to merge triangles. In this paper we explore the effect of the sampling method used to assign a color to triangles.

4 Region and Edge-Adaptive Sampling

Multiple authors [15, 14] have suggested the use of circumradius to discern large triangles in the middle of objects from small triangles on their boundary, with these small triangles being subsequently merged with others based on regional properties such as color. It then becomes important how one measures the pixel color distribution under a triangle. Monte Carlo sampling has been suggested [22]

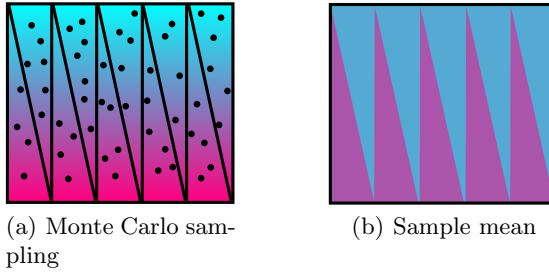


Fig. 2. (a) Triangle colors are estimated by uniform sampling over their area, either by a Monte Carlo method or rasterization. (b) The shape of triangles biases the sampling.

to obtain triangle color information, and enumerating the pixels under a triangle by rasterization is an obvious choice. However, uniform sampling of pixel intensity over the triangle interior inherits a bias from the shape of the triangle. If the triangle has one edge much shorter than the other two then the barycenter shifts toward that edge and biases the sampling in that direction, as shown in Figure 2. The desired behavior of an edge detector is to yield a high density of points along the boundary of an object and few in its interior, so we expect that many such skinny triangles will occur.

To counter these triangle sampling artifacts we bias the sampling in a different direction. Rather than sample uniformly within the triangle, we propose to sample within its circumcircle. We observe that a Gabriel face contains the circumcenters of all the Delaunay triangles it comprises. The amount which a circumcircle extends outside of the Gabriel face depends on the local properties of the shape at the Gabriel edge. In Figure 3(a) the Gabriel edge is long compared to the radii of the maximal circumcircles in the adjacent faces and does not obstruct the circumcircles from penetrating across the edge. In Figure 3(b) the Gabriel edge is shorter, forming a narrowing of the shape which restricts the amount which the neighboring circumcircles overlap. In Figure 3(c) the Gabriel edge obviously bounds the shape and so the circumcircle is contained well within the interior of the face. By sampling regional properties, such as average pixel color, within the circumcircles of Delaunay triangles we not only alleviate the undesirable sampling bias of skinny triangles but introduce desirable bias by sampling away from Gabriel edges which are likely to bound shapes and smoothing the sampling across Gabriel edges which are likely to lie in the middle of shapes.

We can exaggerate the influence of boundary shape on the sampling bias by shrinking the circumcircles. Samples are then shifted away from the edgels, which are those pixels indicated by the edge detector to be regions of significant image variability, e.g. high gradient or fluctuating image statistics. The effect is similar to anisotropic diffusion [21] but is decoupled from the image gradient by way of the edge detector. Figure 4 demonstrates the circumcircle-based sampling method and contrasts it with triangle-based sampling.

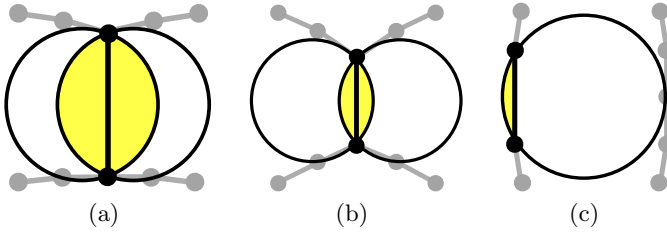


Fig. 3. (a) Circumcircles overlap across Gabriel edges in the interior of shapes, creating a smoothing effect when color is sampled within these circles. (b) Circumcircles do not overlap as much in regions where the shape is varying, and hardly at all (c) across edges on the shape boundary.

To evaluate the effect of triangle color sampling on region merging we use the binary scale-climbing region merging heuristic proposed by Guigues *et al.* using the Mumford-Shah two-term energy function [11]. Figure 5(a) shows the Delaunay triangles that remain after removing edges that are longer than 5 pixels. We arbitrarily chose to merge regions until 40% of edgels were removed. Figure 5(b) shows the merged regions resulting from uniform sampling of triangle colors, and in Figure 5(c) the triangles are given the colors sampled under their circumcircles. Notice in Figure 5(b) how some long, thin triangles poke out of the regions they have been merged with. The region boundaries in Figure 5(c) correspond more closely with the Gabriel graph, as we expect from the deliberate sampling bias. Also note that the dynamic range of colors is preserved better with the proposed sampling method, as color is not sampled along region boundaries.

It should be noted that the sum of the areas of the circumcircles is not linearly related to the sum of the areas of their triangles. Consider the Delaunay triangulation of points placed along two parallel lines at unit intervals. Move the lines apart and the area of each triangle grows in proportion to the distance moved, as do the radii of the circumcircles and thus their areas grow quadratically. To mitigate what is potentially a source of quadratic computational complexity, we use a constant-time sampling procedure for each circumcircle. We generate a pyramid of downsampled images, where each pixel at level i is the average of the four pixels below it at level $i + 1$. A circumcircle centered at c with radius r is given the color obtained from the image at level $\lceil \log_2 r \rceil$ using bilinear interpolation to weight the four downsampled pixels in the neighborhood of c .

5 Directional Features

Large, significant features can readily be identified by eliminating long Gabriel edges, which are unlikely to bound objects because they have very little corroboration from the edge detector. It does not suffice, however, to consider edge length alone when identifying smaller features, which may still be visually and semantically significant. Rather than fall back to color-based region merging in

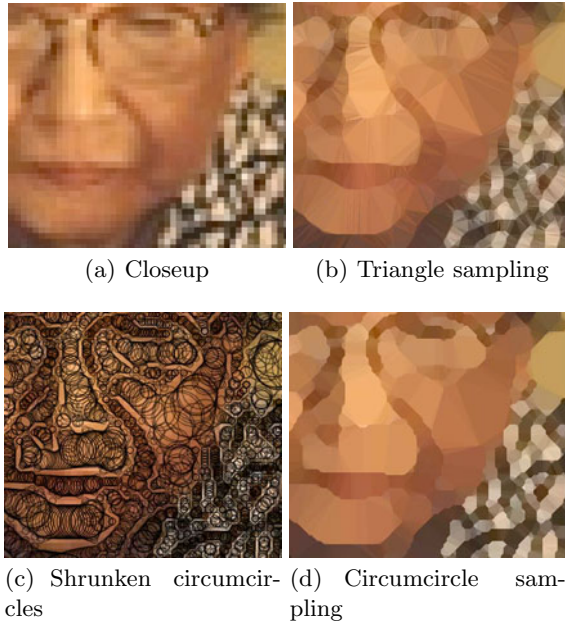


Fig. 4. Closeup on the face of the woman in Figure 8(a), left. (a) Original detail. In (b), Delaunay triangles are given the average color of pixels underneath them using Monte Carlo sampling and bilinear interpolation. Circumscribing circles of Delaunay triangles are shown in (c), with 75% of their original radius. In (d) triangles are given the average color of pixels under their shrunken circumcircles.

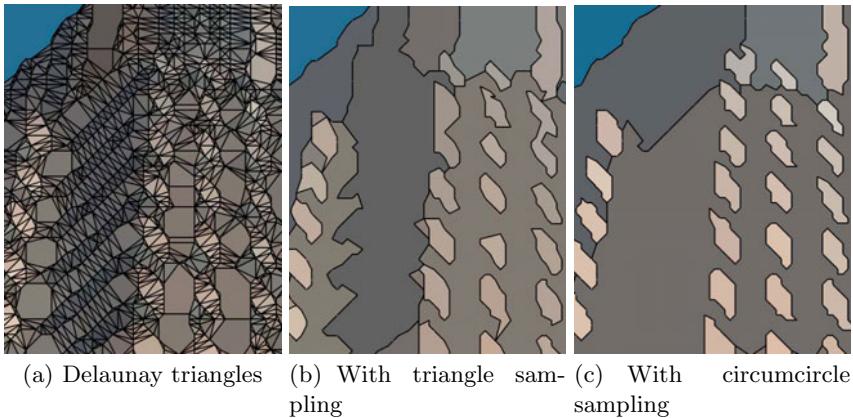


Fig. 5. Closeup on the building in Figure 8(d), right. (a) Delaunay triangles remaining after removing edges longer than 5 pixels. Small triangles are merged using the heuristic of Guigues *et al.* [11] until 40% of edgels have been removed. In (b) triangles colors were obtained by uniform sampling over their interiors, and in (c) by the proposed method of sampling in their circumcircles.

the small-radius regime, we propose to extract long, thin features using directional properties. A *directional feature* is a connected set of Gabriel faces with minimal variation in thickness and orientation.

To identify these features we first classify all Gabriel edges as *contour* or *chord*. A contour edge is definitely part of the shape boundary, whereas a chord edge may or may not be contained in the shape interior. An edge e is classified as contour if its length is not greater than $\sqrt{2}$ pixels, or a chord otherwise. This corresponds to the usual notion of 8-neighbor pixel connectivity, although different thresholds may be used with sub-pixel edge detection [14]. Additionally, Gabriel edges may be classified as contour if shape thickness varies too much across the edge. This variation is measured by the ratio of edge length to the maximum thickness of either adjacent Gabriel face. This additional criterion for discriminating between contour and chord edges will be justified shortly. The *degree* of a Gabriel face is then number of Gabriel chords on its boundary. Candidate groups of degree-2 faces are then identified. A *sequence* of Gabriel faces is one such that faces f_i and f_{i+1} share a common Gabriel edge. We find all sequences of degree-2 faces and then recursively split them until there is no significant variation of thickness or directionality within each sequence.

The variation of thickness is measured by the ratio T_{\min}/T_{\max} , where T_{\min} is the length of the shortest Gabriel edge between subsequent faces of the sequence, and T_{\max} is the greatest thickness of any face in the sequence. If this ratio smaller than some fixed threshold (we use 0.5 in the results presented below) then the sequence is split at the shortest Gabriel edge. Figure 6(a) shows a pair of Gabriel faces that fail this condition. Because we only consider sequences of Gabriel faces with variation in thickness above this threshold, it is justified to classify as contour those Gabriel edges across which thickness varies too much, as they cannot possibly be part of a sequence. This results in more faces having degree of 2 and the formation of longer candidate sequences.

The variation of directionality of the sequence is measured by considering how well the sequence is approximated by a straight line segment between the circumcenters of the first and last Gabriel faces (centers of their maximal circumcircles.) Call these centers p_0 and p_1 . Let e be a Gabriel edge between subsequent Gabriel faces, with endpoints q_0 and q_1 . Let t be the value for which $q_0 + t(q_1 - q_0)$ intersects line p_0p_1 , and let $C_e = |0.5 - t|$. When $C_e = 0$ the line p_0p_1 cuts e directly in half. If $C_e < 0.5$ then p_0p_1 intersects e somewhere in its interior. $C_e > 0.5$ means that p_0p_1 strays outside the ‘‘channel’’ defined by e . We bound the directional variation of Gabriel face sequences by placing a threshold on C_e for each edge e encountered in the sequence. If some C_e value is greater than this threshold, we split the sequence at the edge with the largest value of C_e . Figure 6(b) shows a sequence of Gabriel faces with small values of C_e for the intervening edges, whereas Figure 6(c) exhibits larger values of C_e .

Figure 7 shows the effect of varying the threshold on C_e from 0.125 to 2. To illustrate the directional features we show in white their *centerlines*, which are the line segments connecting circumcenters of Delaunay triangles encountered

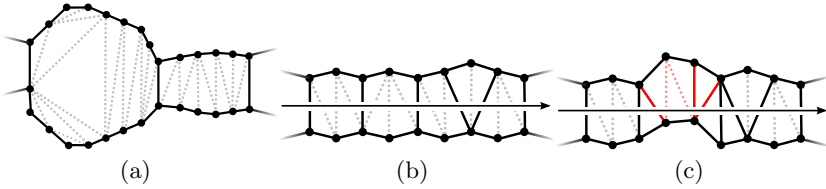


Fig. 6. (a) Thickness varies too much between the larger Gabriel face and the shared Gabriel edge, and so these two faces are not a candidate directional feature. (b) The line from the first Gabriel edge midpoint to the last cuts all intervening edges close to their middles, so these faces constitute a directional feature. (c) The line strays outside of the middle of some of the intervening edges (colored) and prevents the faces from forming a directional feature.

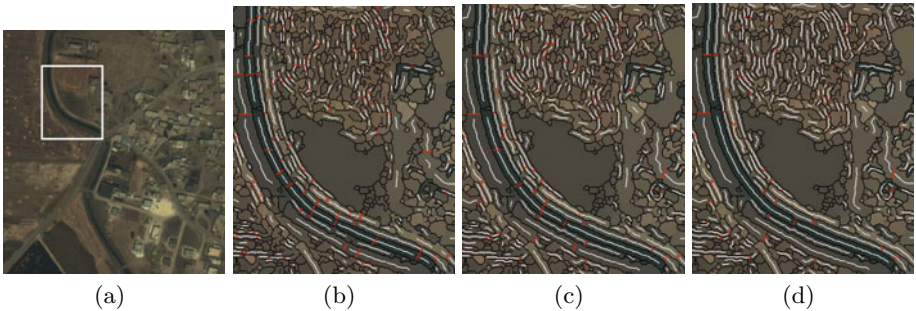


Fig. 7. (a) Satellite image. Directional features are found using varying thresholds: (b) $1/8$, (c) $1/2$ and (d) 2 . Contour edges are drawn in black and the centerline approximations of directional features are drawn in white. Gabriel edges which cut directional features are drawn in red. Note how larger thresholds cause fewer cuts.

on a walk from the first Gabriel edge to the last. For low thresholds of C_e these centerlines will be nearly straight while larger thresholds allow them to curve.

6 Results

Figure 8 shows the proposed method applied to four images from the Berkeley segmentation dataset [20]. A 2x-upsampled Canny edge detector was used with Gaussian smoothing $\sigma = 1.5$, high and low hysteresis thresholds set to 80th and 70th percentile of image gradient magnitude, respectively. Delaunay edges longer than 5 pixels were removed. Directional features were identified with $C_e \leq 1$ and $T_{\min}/T_{\min} \geq 1/2$. Images in the middle row of Figure 8 show the result of the proposed pre-segmentation methods. Initial regions with Gabriel faces having thickness not less than 2.5 pixels are shown with their average color. Smaller regions are left white, except for directional features whose centerlines are drawn in black. Region colors were obtained with the proposed circumcircle-sampling



Fig. 8. Top row: original images. Middle row: Large segmentation regions, those containing Gabriel faces of thickness greater than 2.5 pixels, are drawn with their color. Small regions are left white, except for the centerlines of directional features, which are drawn in black. Bottom row: Regions remaining after merging until 60% of edgels are removed.

method. The bottom row of Figure 8 shows regions that remain after removing 60% of edgels using the binary scale-climbing heuristic [11].

7 Conclusion

Completion of edge detection boundaries by Delaunay triangulation allows shape information to be explicitly incorporated into the pre-segmentation process.

Shape-adaptive color sampling tangibly improves region merging results. Early identification of salient image features based on color, size, directionality and uniformity of thickness allows for these features to be captured as discrete perceptual units within the region merging hierarchy, rather than relying on the region merging heuristic to hopefully coalesce them at some point in the merging process. The effect is to increase the density of information attributed to regions contained in the hierarchy, which in turn increases the value of hierarchical region merging as a tool for higher-level machine vision tasks such as parts-based object recognition and scene description [24,13].

References

1. Arbelaez, P., Cohen, L.: Constrained image segmentation from hierarchical boundaries. In: Proc. IEEE Computer Vision and Pattern Recognition (2008)
2. Arbelaez, P., Maire, M., Fowlkes, C.C., Malik, J.: From contours to regions: An empirical evaluation. In: Proc. IEEE Computer Vision and Pattern Recognition, pp. 2294–2301 (2009)
3. Canny, J.: A computational approach to edge detection. Readings in computer vision: issues, problems, principles, and paradigms, 184 (1987)
4. Edelsbrunner, H., Mücke, E.: Three-dimensional alpha shapes. In: Proceedings of the 1992 workshop on Volume visualization, p. 82. ACM, New York (1992)
5. Elder, J., Zucker, S.: Computing contour closure. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 399–412. Springer, Heidelberg (1996)
6. Elder, J.: Are edges incomplete? International Journal of Computer Vision 34(2/3), 97–122 (1999)
7. Elder, J., Goldberg, R.: Ecological statistics of Gestalt laws for the perceptual organization of contours. Journal of Vision 2(4), 5 (2002)
8. Giesen, J., John, M.: The flow complex: A data structure for geometric modeling. Computational Geometry 39(3), 178–190 (2008)
9. Gonzalez, R., Woods, R.: Digital Image Processing, 2nd edn. Prentice-Hall, Upper Saddle River (2002)
10. Goodman, J., O'Rourke, J.: Handbook of Discrete and Computational Geometry. Chapman & Hall, Boca Raton (2004)
11. Guigues, L., Cocquerez, J., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision 68(3), 289–317 (2006)
12. Haris, K., Efstratiadis, S., Maglaveras, N., Katsaggelos, A.: Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on Image Processing 7(12), 1684–1699 (1998)
13. Hoiem, D., Efros, A., Hebert, M.: Closing the loop in scene interpretation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2008)
14. Köthe, P., Stelldinger, H.M.: Provably correct edgel linking and subpixel boundary reconstruction. In: Franke, K., Müller, K.R., Nikolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 81–90. Springer, Heidelberg (2006)
15. Letscher, D., Fritts, J.: Image segmentation using topological persistence. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 587–595. Springer, Heidelberg (2007)
16. Leung, T., Malik, J.: Contour continuity in region based image segmentation. In: Burkhhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 544–559. Springer, Heidelberg (1998)

17. Manjunath, B., Chellappa, R.: A unified approach to boundary perception: edges, textures and illusory contours. *IEEE Transactions on Neural Networks* 4, 96–108 (1993)
18. Marr, D.: *Vision*. W.H. Freeman & Co, New York (1982)
19. Marr, D., Hildreth, E.: Theory of edge detection. *Proceedings of the Royal Society London B* 207, 187–217 (1980)
20. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(5), 530–549 (2004)
21. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence* 12(7), 629–639 (1990)
22. Prasad, L., Skourikhine, A.: Vectorized image segmentation via trixel agglomeration. *Pattern Recognition* 39(4), 501–514 (2006)
23. Ren, X., Fowlkes, C., Malik, J.: Learning probabilistic models for contour completion in natural images. *International Journal of Computer Vision* 77, 47–63 (2008)
24. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(7), 1270–1281 (2008)
25. Stelldinger, P., Köthe, U., Meine, H.: Topologically correct image segmentation using alpha shapes. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) *DGCI 2006*. LNCS, vol. 4245, pp. 542–554. Springer, Heidelberg (2006)
26. Sumengen, B., Manjunath, B.: Multi-scale edge detection and image segmentation. In: *Proc. of European Signal Processing Conference*, pp. 4–7 (2005)
27. Tabb, M., Ahuja, N.: Multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Image Processing* 6(5), 642–655 (1997)
28. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6), 583–598 (1991)
29. Walters, D.: Selection of image primitives for general-purpose visual processing. *Computer Vision, Graphics, and Image Processing* 37, 261–298 (1987)
30. Ziou, D., Tabbone, S.: Edge detection techniques: an overview. *International Journal on Pattern Recognition and Image Analysis* 8(4), 537–559 (1998)

Universal Seed Skin Segmentation

Rehanullah Khan¹, Allan Hanbury³, and Julian Stöttinger^{1,2}

¹ Computer Vision Lab, Vienna University of Technology

² CogVis Ltd., Vienna, Austria

³ Information Retrieval Facility, Vienna, Austria

Abstract. We present a principled approach for general skin segmentation using graph cuts. We present the idea of a highly adaptive universal seed thereby exploiting the positive training data only. We model the skin segmentation as a min-cut problem on a graph defined by the image color characteristics. The prior graph cuts based approaches for skin segmentation do not provide general skin detection when the information of foreground or background seeds is not available. We propose a concept for processing arbitrary images; using a universal seed to overcome the potential lack of successful seed detections thereby providing basis for general skin segmentation. The advantage of the proposed approach is that it is based on skin sampled training data only making it robust to unseen backgrounds. It exploits the spatial relationship among the neighboring skin pixels providing more accurate and stable skin blobs. Extensive evaluation on a dataset of 8991 images with annotated pixel-level ground truth show that the universal seed approach outperforms other state of the art approaches.

1 Introduction

Skin detection has a wide range of applications both in human computer interaction and content based analysis. Applications such as: detecting and tracking of human body parts [1], face detection [2], naked people detection, people retrieval in multimedia databases [3] and blocking objectionable content [4], all benefit from skin detection. The most attractive properties of color based skin detection are the potentially high processing speed and invariance against rotation, partial occlusion and pose change. However, standard skin color detection techniques are negatively affected by changing lighting conditions, complex backgrounds and surfaces having skin-like colors.

We present the idea of skin segmentation based on a global seed which we represent as the universal seed. With the universal seed we successfully remove the need for local foreground seeds from an image. No time consuming training is required and the universal seed can easily be updated with new skin examples under different lighting conditions. The skin segmentation problem is modeled as a min-cut problem on a graph defined by the image color characteristics. We use an efficient algorithm [5] for finding min-cut/max-flow in a graph. Experiments are performed following recent evaluation [6,7,8,9,10]. We select the best performing approaches, namely AdaBoost, BayesNet, NaiveBayes, YCbCr static model, RBF network and J48. The results show that our universal seed approach outperforms other approaches.

Related work regarding skin detection and segmentation is presented in Section 2. Section 3 presents a framework for seed based segmentation, the graph building process,

weights assignment and the universal seed for skin segmentation. Experimental details are given in Section 4 and the results are discussed in Section 5. Section 6 concludes.

2 Related Work

In computer vision, skin detection is used as a first step in face detection, e.g. [11], and for localization in the first stages of gesture tracking systems, e.g. [1]. It has also been used in the detection of naked people [12,13] and for blocking objectionable content [4]. The latter application has been developed for videos.

The approaches to classify skin in images can be grouped into three types of skin modeling: parametric, non-parametric and explicit skin cluster definition methods. The parametric models use a Gaussian color distribution since they assume that skin can be modeled by a Gaussian probability density function [14]. Non-parametric methods estimate the skin-color from the histogram that is generated by the training data used [15].

An efficient and widely used method is the definition of classifiers that build upon the approach of skin clustering. This thresholding of different color space coordinates is used in many approaches, e.g. [16] and explicitly defines the boundaries of the skin clusters in a given color space, generally termed as static skin filters. The static filters used in YCbCr and RGB color spaces for skin detection are reported in [7] and [17]. The main drawback of static filters is a comparably high number of false detections [6]. Khan et al [18] addressed this problem by opting for a multiple model approach, which makes it possible to filter out skin for multiple people with different skin tones and reduce false positives.

The choice of a color space is important for many computer vision algorithms because it induces the equivalence classes to the detection algorithms [19]. Color spaces like the HS* family transform the RGB cube into a cylindrical coordinates representation. They have been widely used in skin detection scenarios, such as [20,21]. Perceptually uniform color spaces like the CIELAB, CIELUV are used for skin detection e.g. in [2]. Orthogonal color spaces like YCbCr, YCgCr, YIQ, YUV, YES try to form as independent components as possible. YCbCr is one of the most successful color spaces for skin detection and used in e.g. [22].

Neural networks [23], Bayesian Networks e.g. [8], Gaussian classifiers e.g. [15], and self organizing maps [20] have been used to try to increase the classification accuracy.

In the literature of segmentation, Graph-cuts provide a globally optimal solution for N -dimensional segmentation when the cost function has specific properties as defined in [24]. A semi-automatic method for general image segmentation was created by Boykov et al. [24]. A user puts marks on the image, acting as a cue for being counted as segments and updating the marks without graph reconstruction. The method of Li et al. [25] consists of two steps: an object marking task as in [24] and the pre-segmentation, followed by a simple boundary editing process. The work of Shi & Malik [26] segments the image into many non-overlapping regions. They introduced normalized graph cuts and the method has often been used in combination with computing pixel neighborhood relations using brightness, color and texture cues [27].

Micusik et al. [28] make an assumption that each textured or colored region can be represented by a small template, called the seed and positioning of the seed across the

input image gives many possible sub-segmentations of the image. A probability map assigns each pixel to just one most probable region and produces the final pyramid representing various detailed segmentations. Each sub-segmentation is obtained as the min-cut/max-flow in the graph built from the image and the seed. Graph cuts is used for skin segmentation in [29] using both the foreground and background seeds. The foreground seeds are obtained through the face detection algorithm assuming detected faces in the image for skin segmentation, thereby lacking in generic skin detection. We introduce the universal seed concept thereby providing a basis for general skin segmentation when no seeds are available from local images.

3 Universal Seed Skin Segmentation

Using the universal seed based skin segmentation, we exploit the spatial relationship among the skin pixels thereby achieving a performance boost for segmentation. For skin segmentation, a graph is constructed whose nodes represent image pixels and whose edges represent the weights. The min-cut/max-flow algorithm presented in [5] is used for the graph cut.

3.1 Graph Representing the Skin Image

We use a skin segmentation technique based on an interactive graph cut method as used in [28] and [24]. Before segmentation we construct a graph. The graph is shown in Figure 1 for a 9 pixel image and 8-point neighborhood N with representative pixels q and r . For an image, the number of graph nodes equals the pixel count plus two extra nodes labeled as F, B representing foreground and background respectively. There are two types of weights to be set, the foreground/background (skin/non-skin) weights and the neighborhood weights. The foreground weights are computed based on the universal seed. The background weights are calculated from all of the pixels in the image. For min-cut/max-flow a very efficient algorithm [5] is used.

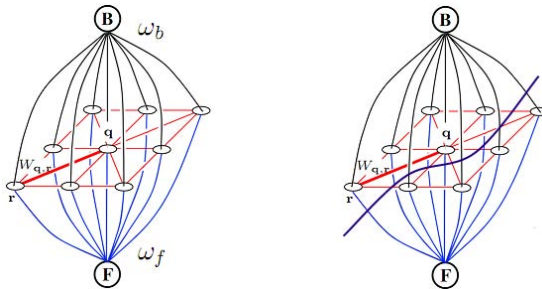


Fig. 1. Left: Graph representation for 9 pixel image. Right: A cut on the graph.

Neighborhood Weights. For a greyscale image the neighborhood weights (weight matrix) $W_{q,r}$, as reported by [24] are

$$W_{q,r} \propto e^{-\frac{\|I_q - I_r\|^2}{2\sigma^2}} \cdot \frac{1}{\|q - r\|} \quad (1)$$

where I_q and I_r are the intensities at point q and point r , $\|q-r\|$ is the distance between these points and σ is a parameter. For color images we modify the above function to take color into account,

$$W_{q,r} = e^{-\frac{\|c_q - c_r\|^2}{\sigma_1}} \cdot \frac{1}{\|q-r\|} \quad (2)$$

where c_q and c_r are the YCbCr vectors of points at the position q and r . $\|q-r\|$ is the distance between these points and σ_1 is a parameter. For skin detection purposes a value of $\sigma_1 = 0.02$ is used, which is the optimized value for segmentation in [30] and is obtained experimentally for giving the best performance on a large database of images. We use a neighborhood window of size 21×21 . For skin segmentation we use a sampling rate of 0.3. This means that we only select at random a sample of 30% of all the pixels in the window. There are two reasons: Firstly by using only a fraction of pixels we reduce the computational demands and secondly only a fraction of pixels allows the use of larger windows and at the same time preserves the spatial relationship between the neighboring pixels.

Foreground/Background Weights. For foreground/background weights, the regional penalty of a point as being “skin” (foreground) \mathcal{F} or “non-skin” (background) \mathcal{B} [30] is

$$R_{\mathcal{F}|q} = -\ln p(\mathcal{B}|c_q) \quad \text{and} \quad R_{\mathcal{B}|q} = -\ln p(\mathcal{F}|c_q) \quad (3)$$

where $c_q = (c_Y, c_{Cb}, c_{Cr})^T$ stands for a vector in \mathbb{R}^3 of YCbCr values at the pixel q . The posterior probabilities in Equation 3 are computed as follows,

$$p(\mathcal{B}|c_q) = \frac{p(c_q|\mathcal{B})p(\mathcal{B})}{p(\mathcal{B})p(c_q|\mathcal{B}) + p(\mathcal{F})p(c_q|\mathcal{F})} \quad (4)$$

For the skin segmentation problem we first demonstrate it on $p(\mathcal{B}|c_q)$, for $p(\mathcal{F}|c_q)$ the steps are analogous. Initially we fix $p(\mathcal{F}) = p(\mathcal{B}) = 1/2$ and thus,

$$p(\mathcal{B}|c_q) = \frac{p(c_q|\mathcal{B})}{p(c_q|\mathcal{B}) + p(c_q|\mathcal{F})} \quad (5)$$

where the “skin” and “non-skin” prior probabilities are

$$p(c_q|\mathcal{F}) = f_{c_Y}^Y \cdot f_{c_{Cb}}^{Cb} \cdot f_{c_{Cr}}^{Cr} \quad \text{and} \quad p(c_q|\mathcal{B}) = b_{c_Y}^Y \cdot b_{c_{Cb}}^{Cb} \cdot b_{c_{Cr}}^{Cr} \quad (6)$$

and $f_i^{\{Y,Cb,Cr\}}$, resp. $b_i^{\{Y,Cb,Cr\}}$, represents the foreground, resp. the background histogram of each color channel separately at the i th bin. $\omega_f = \lambda R_{\mathcal{F}|q}$, where ω_f is the foreground weight, λ is set to 1000 and controls the importance of penalties for foreground and background against the neighborhood weights. Similarly the background weight ω_b is given by $\omega_b = \lambda R_{\mathcal{B}|q}$.

3.2 Universal Seed

A local skin patch from the image can be used as a seed to detect skin in an image. One solution for obtaining local seeds from an image is to use a face detector. A face

detector is normally followed by post filtering steps for the removal of non-skin portion from the face for using it as the local skin seed. In case of failure of face detection the local seed based skin segmentation will fail. We propose a concept for processing arbitrary images; using a universal seed to overcome the potential lack of successful seed detections thereby providing basis for using static foreground weights based skin segmentation. With the universal seed, the objective is producing a seed/template that is as general as possible and can be used as skin filter. We base the segmentation process on positive training data samples only, exploiting the spatial relationship between the neighborhood skin pixels. For the universal seed, different skin tones are collected, see Figure 2. These positive skin samples cover different ethnicities in different lighting conditions. For the universal seed we do not use the negative (non-skin) portion of the image. Since there could be infinite background/negative training data, the objective is taking a skin/non-skin decision based on representative skin samples. For the skin scenario this makes sense as the skin covers a well defined region in a color space. We denote these positive representative skin samples as the universal seed. The universal seed is highly adaptive. For adding a new skin patch under different lighting conditions we just have to merge it with skin patches and recalculate the foreground histogram. The foreground histogram in Equation 6 for a new image is calculated based on this seed. The background histogram is calculated from the whole image. Since $\sum_{i=1}^N \bar{b}_i = 1$, the probability $p(c_q|\mathcal{B})$ gives smaller values than $p(c_q|\mathcal{F})$ for the “skin” colors present in the universal seed therefore we compute the background histogram from all the image pixels.



(a)

Fig. 2. Skin samples used for universal seed

4 Experiments

Universal seed based skin segmentation is compared to skin segmentation using Adaboost, BayesNet, NaiveBayes, YCbCr static model, RBF network and J48 on the basis of F-measure and specificity. The dataset used is available on-line¹. A total of 8991 images with annotated pixel-level ground truth are used as test images for evaluation.

Following Kakumanu [6], we select explicit thresholding (YCbCr static filter) because it is a fast and simple rule based filter, successfully used in [7] for skin detection. We select the Bayesian network [8] and neural network [9] based classifiers based on the reported best performance in [6]. Based on the independent feature model we select the Naive Bayesian. J48 is selected based on the superior performance in [31] for tree based classifier. Following [10] boosting is the optimal detection method for skin color and faces.

¹ <http://www.feeval.org>

In experiments we aim to combine these evaluations in one experimental setup and evaluate our proposed approach. It is measured precisely per pixel for the dataset.

The evaluation is based on F-measure and specificity for 8991 test images. The F-measure is calculated by evenly weighting precision and recall. The specificity is defined as the true negative rate. For all the experiments, the color space used is YCbCr. The three components of YCbCr are used as feature vectors for AdaBoost, BayesNet, NaiveBayes, RBF network and J48.

We aim to evaluate the approaches on unseen data. For training data, we choose 118 arbitrary images from the Internet with a great variety of skin colors and lighting conditions. Therefore, the training data is non-overlapping and from different sources with respect to the test dataset. The images are annotated per pixel and provide positive and negative training data. The total number of features for skin pixels is 2113703, the number of negative training samples is 6948697. For the universal seed we do not use the negative (non-skin) pixels. This positive set of skin pixels is denoted as the universal seed. Example positive training data can be found in Figure 2. In this context, the proposed approach uses significantly less training data than other approaches. In the following, the experimental results are presented and discussed.

5 Results

Example results of the proposed approach using the universal seed approach are visualized in Figure 3. Figure 4 reports cases where skin is missed or false detections are considered as skin.

Specificity calculated on a per pixel basis is reported in Figure 5. Figure 6 reports mean and standard deviation for specificity and F-measure calculated on a per image basis. Figure 7 shows the F-measure calculated on a per pixel basis. In Figure 5 it can be seen that the specificity of universal seed approach is higher than YCbCr static approach and less than AdaBoost, BayesNet, NaiveBayes, RBF network and J48. Similarly, mean and standard deviation for specificity in Figure 6 shows that the specificity mean (0.54) of Universal seed approach is lower than that of AdaBoost (0.77), BayesNet (0.63), NaiveBayes (0.68), RBF network (0.85) and J48 (0.83) with standard deviations of 0.35, 0.21, 0.32, 0.30, 0.11, 0.11 for universal seed, AdaBoost, BayesNet, NaiveBayes, RBF network and J48 respectively. This is because the universal seed approach is based on training on positive data only and therefore lower true negative rate. The specificity mean of universal seed is higher than that of YCbCr static approach (0.41).

In terms of increasing/decreasing trend in Figure 6 for specificity, the universal seed approach is similar to BayesNet and NaiveBayes approaches. The mean and standard

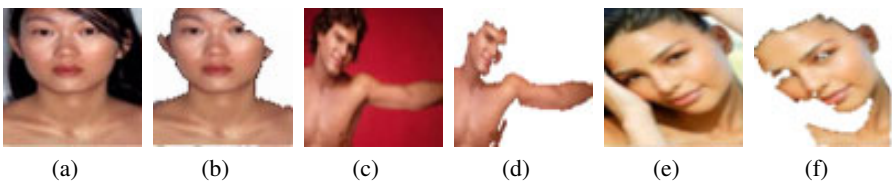


Fig. 3. Universal seed skin segmentation: successful skin detection



Fig. 4. Universal seed skin segmentation: cases where skin is not properly segmented

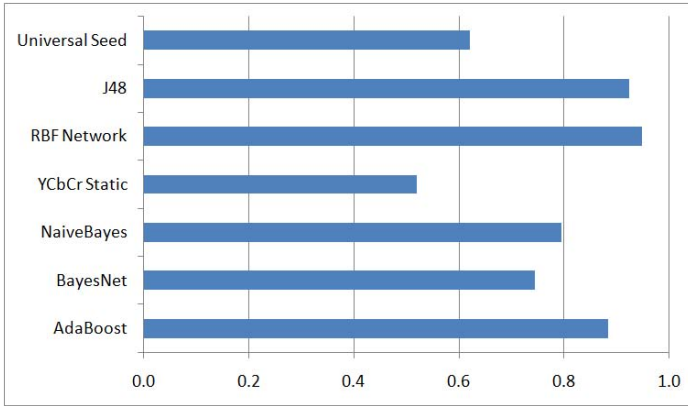


Fig. 5. Specificity for 8991 images. Since the universal seed approach is based on positive data only, the true negative rate is not as high as the F-measure given in Figure 7. The values reported are on a per pixel basis.

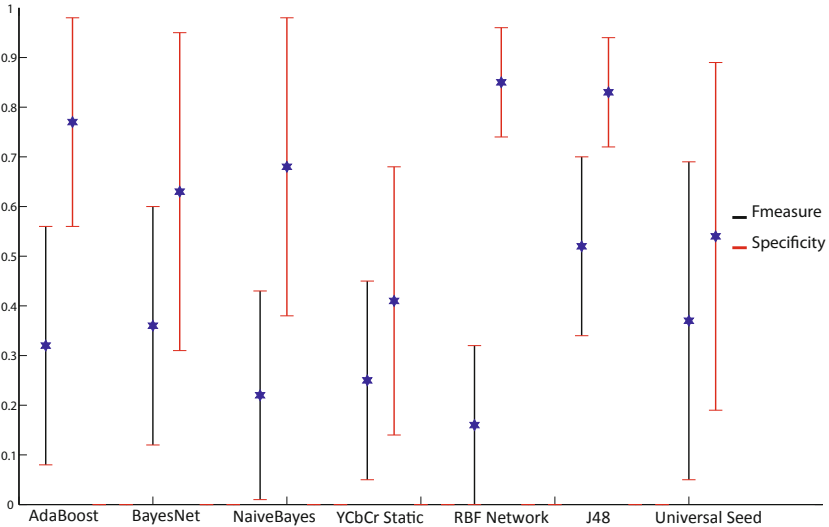


Fig. 6. Mean and standard deviations for F-measure and specificity for 8991 test images. The values reported are on a per image basis.

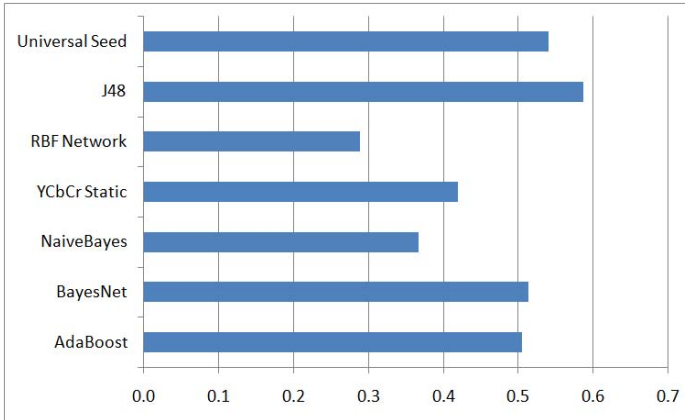


Fig. 7. F-measure for 8991 images. The universal seed approach outperforms other approaches in terms of precision and recall with the exception of tree based classifier (J48). The values are calculated on a per pixel basis.

deviation for F-measure in Figure 6 show that the universal seed mean (0.37) is lower than J48 (0.52) and higher than AdaBoost (0.32), BayesNet (0.36), NaiveBayes (0.22), YCbCr static (0.25) and RBF network (0.16) with standard deviations of 0.32, 0.18, 0.24, 0.24, 0.21, 0.20, and 0.16 for universal seed, J48, AdaBoost, BayesNet, NaiveBayes, YCbCr static and RBF network respectively.

Regarding precision and recall, Figure 7 shows that the universal seed approach has higher F-measure (0.54) compared to other approaches with the exception of the tree based classifier J48 with F-measure of 0.59. The universal seed approach provides increased classification performance of almost 4% to AdaBoost, 3% to Bayesian network, 18% to Naive Bayesian, 12% to YCbCr static, 26% to RBF network and decreased performance of almost 5% compared to J48.

For the test data set the universal seed approach outperforms other approaches in terms of precision and recall with the exception of J48. For the skin detection scenario, the J48 simple rule based decision classification generalizes well for simple feature based classification with high F-score outperforming the universal seed approach.

6 Conclusion

We provide a basis for skin segmentation based on positive training data only using a universal seed to overcome the potential lack of successful seed detections from within an image. The universal seed for skin detection is based on learning a skin model for foreground weights of the graph using skin samples. Experiments on a database of 8991 images with annotated pixel-level ground truth show that using positive training data only, the universal seed approach is well suited to stable and robust skin detection, outperforming other approaches which require negative and positive training data with the exception of J48.

Acknowledgment

This work was partly supported by the Austrian Research Promotion Agency (FFG), project OMOR 815994, the Higher Education Commission of Pakistan under Overseas scholarships (Phase-II-Batch-I) and CogVis² Ltd. However, this paper reflects only the authors views; the FFG, HEC or CogVis Ltd. are not liable for any use that may be made of the information contained herein.

References

1. Argyros, A.A., Lourakis, M.I.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 368–379. Springer, Heidelberg (2004)
2. Cai, J., Goshtasby, A.: Detecting human faces in color images. *Image and Vision Computing* 18, 63–75 (1999)
3. Cao, L.L., Li, X.L., Yu, N.H., Liu, Z.K.: Naked people retrieval based on adaboost learning. In: MLC, pp. 1133–1138 (2002)
4. Stöttinger, J., Hanbury, A., Liensberger, C., Khan, R.: Skin paths for contextual flagging adult videos. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5876, pp. 303–314. Springer, Heidelberg (2009)
5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI* 26, 1124–1137 (2004)
6. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. *PR* 40, 1106–1122 (2007)
7. Chai, D., Ngan, K.: Locating facial region of a head-and-shoulders color image. In: *Int. Conf. Automatic Face and Gesture Recognition*, pp. 124–129 (1998)
8. Sebe, N., Cohen, I., Huang, T.S., Gevers, T.: Skin detection: A Bayesian network approach. In: *ICPR*, pp. 903–906 (2004)
9. Phung, S.L., Chai, D., Bouzerdoum, A.: A universal and robust human skin color model using neural networks. In: *IJCNN*, pp. 2844–2849 (2001)
10. Pavlovic, V.: Boosted detection of objects and attributes. In: *CVPR* (2001)
11. Hsu, R., Abdel-Mottaleb, M., Jain, A.: Face detection in color images. *PAMI* 24, 696–706 (2002)
12. Fleck, M.M., Forsyth, D.A., Bregler, C.: Finding naked people. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 593–602. Springer, Heidelberg (1996)
13. Lee, J.S., Kuo, Y.M., Chung, P.C., Chen, E.L.: Naked image detection based on adaptive and extensible skin color model. *PR* 40, 2261–2270 (2007)
14. Yang, M., Ahuja, N.: Gaussian mixture model for human skin color and its application in image and video databases. In: *SPIE*, pp. 458–466 (1999)
15. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. *IJCV* 46, 81–96 (2002)
16. Phung, S.L., Bouzerdoum, A., Chai, D.: Skin segmentation using color pixel classification: Analysis and comparison. *PAMI* 27, 148–154 (2005)
17. Peer, P., Kovac, J., Solina, F.: Human skin colour clustering for face detection. In: *EUROCON*, vol. 2, pp. 144–148 (2003)

² www.cogvis.at

18. Khan, R., Stöttinger, J., Kampel, M.: An adaptive multiple model approach for fast content-based skin detection in on-line videos. In: ACM MM, AREA workshop (2008)
19. Stokman, H., Gevers, T.: Selection and fusion of color models for image feature detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 371–381 (2007)
20. Brown, D., Craw, I., Lewthwaite, J.: A SOM based approach to skin detection with application in real time systems. In: *BMVC 2001*, pp. 491–500 (2001)
21. Fu, Z., Yang, J., Hu, W., Tan, T.: Mixture clustering using multidimensional histograms for skin detection. In: *ICPR, Washington, DC, USA*, pp. 549–552 (2004)
22. Wong, K., Lam, K., Siu, W.: A robust scheme for live detection of human faces in color images. *SPIC 18*, 103–114 (2003)
23. Lee, J.Y., Suk-in, Y.: An elliptical boundary model for skin color detection. In: *ISST*, pp. 579–584 (2002)
24. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: *ICCV-WS 1999*, vol. 1, pp. 105–112 (2001)
25. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. In: *SIGGRAPH, New York, NY, USA*, pp. 303–308 (2004)
26. Shi, J., Malik, J.: Normalized cuts and image segmentation. *PAMI* 22, 888–905 (2000)
27. Malik, J., Belongie, S., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. *International Journal of Computer Vision* 43, 7–27 (2001)
28. Micsúf, B., Hanbury, A.: Automatic image segmentation by positioning a seed. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 468–480. Springer, Heidelberg (2006)
29. Hu, Z., Wang, G., Lin, X., Yan, H.: Skin segmentation based on graph cuts. *Science and Technology* 14, 478–486 (2009)
30. Micsúf, B., Hanbury, A.: Steerable semi-automatic segmentation of textured images. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) *SCIA 2005*. LNCS, vol. 3540, pp. 35–44. Springer, Heidelberg (2005)
31. Khan, R., Hanbury, A., Stoettinger, J.: Skin detection: A random forest approach. In: *ICIP (to appear, 2010)*

A Sharp Concentration-Based Adaptive Segmentation Algorithm

Christophe Fiorio¹ and Andre Mas²

¹ LIRMM, UMR 5506 CNRS
Universite Montpellier 2

F-34095 Montpellier Cedex 5, France

² I3M, UMR 5149 CNRS
Universite Montpellier 2

F-34095 Montpellier, France

Abstract. We propose an adaptive procedure for segmenting images by merging of homogeneous regions. The algorithm is based on sharp concentration inequalities and is tailored to avoid over- and under-merging by controlling simultaneously the type I and II errors in the associated statistical testing problem.

1 Introduction

In this paper we focus on a strategy which belongs to region merging segmentation [1,2]. It consists in starting with pixel or very small groups of connected pixels as primary regions and then grouping them two by two when adjacent regions are considered to be the same. This decision is generally made through a statistical test to decide the merging of regions [3]. The segmentation process uses a merging predicate based on this test to decide, essentially locally, to merge or not regions. Difficulties of these approaches come from essentially two points. First, this locality in decisions has to preserve global properties such as those responsible for the perceptual units of the image. Second, since based on local decision, algorithms are generally greedy and so order-dependent. But conversely, these approaches can yield very efficient algorithms [4,5].

Our goal in this paper is to propose a local criterion based on a statistical test which ensures preserving global properties and stay efficient. The problem of order dependence will be mentioned in section 8 at the end of this document. To this aim we use concentration inequalities as first proposed by C. Fiorio and R. Nock [6,7] and then continued by R. Nock and F. Nielsen [8] to propose a statistical region merging algorithm. We propose to use another concentration inequality in order to try to improve quality of the result but also to improve control on the segmentation by defining more significant parameters and proposing to point out an indifference zone where we know that decision is not sure.

The paper is organized as follow. First in section 2 we present the basic theory we use to define our merging criterion. Then our framework is presented

at section 3. Section 4 shows how to approximate our formula to be able to use them in practice. At last, section 5 presents our main result for region segmentation. Then we compare our results with previous works at section 6 and give some experimental results at section 7. At last we conclude this article at section 8.

2 Basics

Image segmentation by merging of regions or by any technique involving a decision rule (based for instance on a threshold) may be viewed as a traditional two (or more) samples testing problem. Usually some numerical features of two regions must be compared to decide whether they look alike or not. Suppose you are given two regions, say R and R' , made of pixels. Roughly speaking the general test could be written : $\begin{cases} H_0 : \text{The regions } R \text{ and } R' \text{ must be merged} \\ H_1 : \text{The regions must not be merged} \end{cases}$.

We remind the reader that two types of error may appear when implementing a test: rejecting H_0 wrongly called type I error (we shall denote α the probability of the type I error), and rejecting H_1 wrongly called type II error (we shall denote β the probability of the type II error). It is a well known fact that α and β cannot be made arbitrarily small together 9.

Usually authors propose merging predicates based on the control of the type I error (see e.g. 10,11) without controlling the probability of the type II error. The main drawback of this procedure is clearly overmerging (one may not accept H_1 often enough).

In this article we propose an adaptive procedure to control both error types. This procedure is adaptive in the sense that for each test the threshold depends on second order characteristics of the regions to be tested (the variance associated to the greylevels in fact). Besides the decision rule even depends on the size of the regions. In the following \mathbb{E} , \mathbb{V} and \mathbb{P} denote expectation variance and probability respectively. Now we focus on the mathematical setting of our study. We consider that the region R is made of pixels viewed as independent discrete random variables denoted $(X_i)_{i \in |R|}$ with values in $\{0, \dots, 256\}$. We assume that for all $X_i \in R$, $\mathbb{E}X_i = m$ and $\mathbb{V}X_i = \sigma^2$.

Inspired by Nock and Fiorio's theory 6,12,8 of underlying perfect images we claim that two regions should be merged whenever the common expectation of pixels from both images are the same. In fact, $H_0 : m = m'$ and $H_1 : m \neq m'$. Clearly m' is the expectation of the greylevel for region R' . This classical problem could be tackled through ANOVA techniques. But the usual assumption (gaussian distributions and common variance amongst the dataset) are unrealistic. Fortunately our method is based on concentration inequalities and makes independence on the pixels the only serious assumption in the model.

Now focusing on our segmentation problem and denoting X_i (resp. X'_j) the value of the greylevel of pixel i (resp. j) in region R (resp R'), we set the empirical greylevel mean in R by $S = \frac{1}{|R|} \sum_{i \in R} X_i$ and $S' = \frac{1}{|R'|} \sum_{i \in R} X'_i$ in region R' .

Let us denote by σ^2 the common variance of the distributions of pixels within region R and σ'^2 its counterpart for R' . If we deal with the type I error, the decision rule will be based on the difference $|S - S'|$ and on a threshold t_α . The latter being defined by

$$\mathbb{P}_{H_0} (|S - S'| > t_\alpha) \leq \alpha \tag{1}$$

The type II error will be treated below in the paper similarly to the type I error. The next section is devoted to introducing the probabilistic tools needed to handle inequalities such as (1).

3 Random Framework and Tools

3.1 Concentration Inequalities Based on Variance

Concentration inequality roughly claim that the probability for functional of independent random variables to deviate from their means rapidly decrease : these functionals are “concentrated” around a non random number (the expectation) with a high probability.

For instance the well-known Mc Diarmid’s inequality [13] used by C. Fiorio and R. Nock to derive concentration bounds and thresholds for the segmentation algorithm in [6,12,8] is extremely general and powerful in many cases. Yet insofar as the procedure relies on a comparison of means (see (1)), it could be fruitful to use concentration inequality more specifically designed for sums of independent random variables instead of general functionals. Amongst these are the famous Hoeffding’s [14], Bennett’s [15] and Bernstein’s [16] inequalities... It is easily seen that Mc Diarmid’s inequality generalizes Hoeffding’s one since it involves a supremum bound for the variation of each coordinate of the function. On the other hand Bennett’s inequality takes into account the variance of the random variables and we can expect it to be more precise in many situations. Here is this inequality in its general formulation (the notations are those of Ledoux-Talagrand (1991) [17]):

$$\mathbb{P} \left(\left| \sum_i V_i \right| > t \right) \leq 2 \exp \left(\frac{t}{a} - \left(\frac{t}{a} + \frac{b^2}{a^2} \right) \log \left(1 + \frac{at}{b^2} \right) \right) \tag{2}$$

Where the independent and **centered** real random variables V_i ’s are such that $|V_i| \leq a$ almost surely and $b^2 = \sum_i \mathbb{E}V_i^2$.

The preceding bound may be rearranged in a more practical way. Indeed if $\mathbb{E}V_i^2 = \sigma^2$ for all i . Then $b^2 = n\sigma^2$ and setting $x = at/\sigma^2$

$$\mathbb{P} \left(\left| \sum_i V_i \right| > nt \right) \leq 2 \exp \left[-\frac{n\sigma^2}{a^2} ((1+x) \log(1+x) - x) \right] \tag{3}$$

We have already seen above and will make more precise below that one of our goals would be to determine the smallest threshold t (or equivalently x) such

that the term of the right in the above display is less than α (α is the type I error). The solution is

$$x_{th}^* = \min \left\{ x > 0 : 2 \exp \left[-\frac{n\sigma^2}{a^2} ((1+x) \log(1+x) - x) \right] \leq \alpha \right\} \quad (4)$$

Let us denote φ the positive non-decreasing, convex, one to one function defined on the set of positive real numbers by $\varphi(x) = (1+x) \log(1+x) - x$. This function will be referred to as the “concentration function”, even if this term does not match the usual probabilistic meaning. If the inverse of function φ was explicitly known an obvious solution would be $x_{th}^* = \varphi^{-1} \left(\frac{a^2}{n\sigma^2} \log \left(\frac{2}{\alpha} \right) \right)$. But φ^{-1} cannot be explicitly computed. We will propose an explicitly known function φ^\dagger such that $\varphi^\dagger \geq \varphi^{-1}$ and we will choose $x^* = \varphi^\dagger \left(\frac{a^2}{n\sigma^2} \log \left(\frac{2}{\alpha} \right) \right)$. Obviously the closer function φ^\dagger is from φ^{-1} , the closer x^* is from x_{th}^* ($x^* \geq x_{th}^*$) and the better our threshold. Before tailoring φ^\dagger we underline important features of the bound (3).

3.2 From Gaussian to Poisson Bounds

It is often noted in the literature on concentration inequalities (or on large deviation theory) that, depending on the value of t , exponential bounds such as (2) may be approximated the following way:

- when t is small, namely if $at/b^2 \leq 1/2$

$$\mathbb{P} \left(\left| \sum_i V_i \right| > t \right) \leq 2 \exp(-t^2/4b^2) \quad (5)$$

which illustrates the gaussian behavior of the random sum.

- when t is large the following inequality is sharper

$$\mathbb{P} \left(\left| \sum_i V_i \right| > t \right) \leq 2 \exp \left(-\frac{t}{a} \left(\log \left(1 + \frac{at}{b^2} \right) - 1 \right) \right) \quad (6)$$

and is referred to as a Poisson type behavior.

The main problem relies on the missing link between the two displays above: what should we do when t is neither large nor small? What does even mean “large” and “small” with respect to our problem. We answer these questions in the next section. However we notice that Bernstein [16] proved that, uniformly

in $x \in \mathbb{R}$, $\varphi(x) \geq \frac{x^2}{2(1+x/3)}$. This bound is sharp for small x -and would lead

to an explicit φ^\dagger -but not for large x . The function φ^\dagger we propose below strictly reflects both “domains”: Poisson and Gaussian and even define an intermediate area.

4 Approximating the Inverse Concentration Function

4.1 Tailoring the Function

We set

$$\varphi^\dagger(y) = \begin{cases} \varphi_1^\dagger(y) = \frac{y}{3} + \sqrt{\frac{y^2}{9} + 2y} & \text{if } 0 \leq y \leq 3(\log 3) - 2 \\ \varphi_2^\dagger(y) = \frac{y + 2.3}{\log 3.3} - 1 & \text{if } 3(\log 3) - 2 \leq y \leq 4(\log 4) - 3 \\ \varphi_3^\dagger(y) = \frac{y}{\log\left(\frac{y}{\log y} + 1\right)} \left(1 + \frac{1}{\log y}\right) - 1 & \text{if } 4(\log 4) - 3 \leq y \end{cases} \quad (7)$$

The functions φ_1^\dagger (resp. φ_3^\dagger) lies in the Gaussian (resp Poissonian) domain and φ_2^\dagger features an intermediate domain as announced sooner in the paper.

Now we quickly explain how function φ^\dagger was derived :

- Bernstein bound directly provides φ_1^\dagger : Indeed if $\varphi(x) \geq \frac{x^2}{2(1+x/3)}$,
 $\varphi^{-1}(y) \leq \varphi_1^\dagger(y)$
- Function φ_3^\dagger is obtained as a by product of Newton algorithm applied to [\(4\)](#). Indeed we seek $\min\{x > 0 : ((1+x)\log(1+x) - x) \geq M\}$, where M is known and [\(4\)](#) is rewrited a simpler way. Straightforward calculations prove that $x_0(M) = M/\log(M)$ is a good estimate for large M . Then iterating once Newton's algorithm from $x_0(M)$, we seek the intersection of the tangent at $x_0(M)$ to function φ and the line $y = M$. Denote this point $(\kappa(M), M)$. We just set $\varphi_3^\dagger(y) = \kappa(y)$. The good behavior of this estimate is due to the slow rate of increase of $\varphi'(x) = \log(1+x)$.
- Function φ_2^\dagger linearly interpolates between φ_1^\dagger and φ_3^\dagger .

4.2 Measuring the Error

Many procedures could be proposed to measure the error between the theoretical threshold x_{th}^* and our estimate x^* . We propose the following which is quite intuitive and global.

Take an x , whose image by φ is $y = \varphi(x)$. The approximation error by φ^\dagger is clearly $|x - \varphi^\dagger(\varphi(x))|$.

Definition 1. *As a criterion for measuring the goodness of our approximating function we take $\mathcal{C} = \sup_{x \in \mathbb{R}} \frac{|x - \varphi^\dagger(\varphi(x))|}{x}$.*

The next Proposition enlightens the choice of the numerical constants in [\(7\)](#)

Proposition 1. *If φ^\dagger is chosen as in [\(7\)](#), then $\mathcal{C} \leq 0.055$*

In other words, our method provides a threshold which approximates the optimal one just inducing a 5% error (about 5% in fact...).

Remark 1. Obviously the bound on \mathcal{C} given above may be improved (and made smaller than the 0.05 threshold) by adding a fourth function, say $\varphi_{2.5}^\dagger$, between φ_2^\dagger and φ_3^\dagger .

5 Main Result

5.1 Type I&II Errors

Obviously the greylevel of any region may be viewed as a discrete random variables almost surely bounded by $a = 255$. Assume first that the variances of both regions (denoted σ^2 and σ'^2) are non null. Here is the first result related to error of type I : Set $M = \frac{a^2}{|R|\sigma^2} \log\left(\frac{4}{\alpha}\right)$, $M' = \frac{a^2}{|R'|\sigma'^2} \log\left(\frac{4}{\alpha}\right)$, and $t_I(\alpha) = \frac{\sigma^2}{a}\varphi^\dagger(M) + \frac{\sigma'^2}{a}\varphi^\dagger(M')$.

Theorem 1. *When $m = m'$, $\mathbb{P}(|S - S'| > t_I(\alpha)) \leq \alpha$.*

In other words if $|S - S'| \leq t_I(\alpha)$ we accept the merging of both regions

Remark 2. M and M' both depend on σ^2 and σ'^2 which are unknown and will be approximated by $\frac{1}{|R|} \sum_{i=1}^{|R|} (X_i - \bar{X})^2$ and $\frac{1}{|R'|} \sum_{i=1}^{|R'|} (X'_i - \bar{X}')^2$ respectively.

Now we focus on the error of type II whose probability was denoted β : accepting H_0 wrongly. At this point we need to address a crucial issue. We try to avoid overmerging and we should first define what we mean by “distinct” regions. We guess that if we take two perfectly homogeneous regions says R with all pixels taking values v and R' with all pixels taking values $v + 1$, the human eye may not consider them as “different”. Consequently before going further we should define precisely what we mean by distinct regions. We introduce the parameter $\Delta \in \mathbb{N}$, as the minimum greylevel difference between two perfect regions beyond which the segmentation procedure should always “refuse” a merging.

In other words and with mathematical symbol : if $|\mathbb{E}X - \mathbb{E}X'| > \Delta$ we assume that the average greylevel is too different for both regions to be merged. Obviously Δ is a tuning parameter that will first of all depend on the “universal human eye” but also on the type of images that have to be analyzed.

We are ready to state the next Theorem providing the threshold for type II error.

Theorem 2. *When $|\mathbb{E}X - \mathbb{E}X'| \geq \Delta$ (i.e. when R and R' should be considered as distinct regions),*

$$\mathbb{P}(|S - S'| < \Delta - t_I(\beta)) < \beta$$

and if $|S - S'| \geq \Delta - t_I(\beta)$ we reject the merging.

Remark 3. In fact the previous Theorem bounds $\sup_{|\mathbb{E}X - \mathbb{E}X'| \geq \Delta} \mathbb{P}(|S - S'| \leq t)$ by β . This probability is nothing but the maximal probability of error for merging two distinct regions whereas they should not.

We assumed that $\sigma \neq 0$ and $\sigma' \neq 0$. In practice it may happen that either one or both variances are null. This situation is encountered when all pixels of one (or both) of the regions have the same greylevel (say m and m'). If both variance are equal we propose to merge them iff $|m - m'| \leq \Delta$. Now if only one of the variances is null (say $\sigma' = 0$ and $\sigma \neq 0$) then the region R' will be seen as a perfect region. Our concentration inequalities still hold but in a one sample instead of the two sample framework above. A quick inspection of the proof of theorem [11](#) shows that the bound that appears there must be replaced by

$$\tilde{t}_I(\alpha) = \frac{\sigma^2}{a} \varphi^\dagger(M) \text{ and } \tilde{t}'_I(\alpha) = \frac{\sigma'^2}{a} \varphi^\dagger(M') \tag{8}$$

with M like for theorem [11](#)

5.2 The Segmentation Process

From what was done above the segmentation process decision could be the following :

- If $|S - S'| \leq t_I(\alpha)$ merge R and R'
- If $|S - S'| \geq \Delta - t_I(\beta)$ do not merge R and R'

We know that in the first case (resp. the second), the probability of error is less than α (resp. β). But for both conditions not to be contradictory the following must hold :

$$\Delta \geq t_I(\alpha) + t_I(\beta) \tag{9}$$

An ultimate question arises : what should we do if $t_I(\alpha) < |S - S'| < \Delta - t_I(\beta)$? The interval $[t_I(\alpha), \Delta - t_I(\beta)]$ is sometimes referred to as the indifference zone in statistical testing theory.

We propose to base this last step on an homogeneity test for the second moment (we could also have investigated a randomized test but this approach is less intuitive). Since we cannot make a decision based on means we look at the second order moments : if they are close enough we accept the merging of both regions. Set $V = \frac{1}{|R|} \sum_{i=1}^{|R|} X_i^2$ and $V' = \frac{1}{|R'|} \sum_{i=1}^{|R'|} (X'_i)^2$. Applying Mc Diarmid's (see [18,13](#)) inequality to V and V' we get when $\mathbb{E}X^2 = \mathbb{E}X'^2$:

$$\mathbb{P}(|V - V'| > s) \leq \exp\left(-\frac{2s^2}{\frac{a^4}{|R|} + \frac{a^4}{|R'|}}\right)$$

This time the threshold on the second moment is

$$t_I(\gamma) = \sqrt{\frac{1}{2} \left(\frac{a^4}{|R|} + \frac{a^4}{|R'|} \right) \log\left(\frac{1}{\gamma}\right)} \tag{10}$$

where γ is a prescribed probability.

To summarize the test procedure, we have four parameters (α, β, Δ) and γ where

- Δ is the value of intensity discrepancy that makes two regions being considered as belonging to two different objects in the image ; this threshold would be adapted to the dynamics of the image.
- α is the probability not to merge the two regions whereas we should.
- β is the probability to merge the two regions whereas we should not.
- γ is used in the indifference zone and is the probability that the difference of second order moments of the two regions is too large so that the regions should not be merge.

In order to achieve an optimal time segmentation, we base our algorithm on algorithms described in [19] and used by [6][2][8]. These algorithms are based on the following tests:

1. If $|S - S'| \leq t_I(\alpha)$ merge R and R' where $t_I(\alpha)$ is given in theorem [1].
2. If $|S - S'| \geq \Delta - t_I(\beta)$ do not merge R and R' where $t_I(\beta)$ is computed using same formula as $t_I(\alpha)$ given in theorem [1].
3. If $t_I(\alpha) < |S - S'| < \Delta - t_I(\beta)$ compute $t_I(\gamma)$, V and V' and if $|V - V'| > t_I(\gamma)$, do not merge R and R' otherwise merge R and R' , where $t_I(\gamma)$ is given by equation [10].

For the sake of efficiency we use algorithms proposed in [19] based on Union-Find data structure [20]. To ensure linear complexity, [19] claims that computing the criteria has to be done in constant time. Moreover, updating parameters of regions should also be done in constant time. If we look at formula of theorem [1] we can easily check that, whenever σ and $|R|$ are known, computing $t_I(\alpha)$ can be considered done in constant time (we suppose that in practice arithmetic operations, even log are done in constant time, which is not true in theory but almost true in practice with regard to other operations).

Knowing $|R|$ is easy, it suffices to keep this number as parameter of each region. When a merging is done, we just add the two corresponding values to obtain $|R|$ of the newly created region.

As noted in remark [2], σ^2 is approximated by $\frac{1}{|R|} \sum_{i=1}^{|R|} (X_i - \bar{X})^2$ that can be set to $\sigma^2 = \frac{\sum_{i=1}^{|R|} X_i^2}{|R|} - \bar{X}^2$ and so we only need to keep updated $\sum_{i=1}^{|R|} X_i^2$ and $\sum_{i=1}^{|R|} X_i$ which is as easy to maintain as $|R|$ by the same process.

When we are located in the indifference zone, we must be able to compute in constant time V , V' of and equation [10]. It is simple to check, according to previous remarks, that these equations can also be computed in practical constant time. So we can define a function `Oracle`, based on the 3 tests described at end of section [5.2], which decides if two region R and R' should be merge or not. According to this `Oracle` (see algorithm [Oracle]), `Scanline` and `MergeSquare` algorithms of [19] perform in linear times and so are very fast.

Function Oracle(R, R')

Input: R and R' two regions to be checked

Result: **true** if they should be merged, else **false**

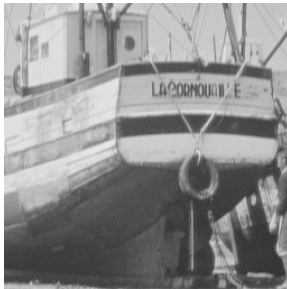
```

1 if ( $\sigma^2 = 0$ ) and ( $\sigma'^2 = 0$ ) then return  $|m - m'| \leq \Delta$ ;
2 else if  $\sigma^2 = 0$  then return  $|S' - m| \leq \tilde{t}'_I(\alpha)$ ;
3 else if  $\sigma'^2 = 0$  then return  $|S - m'| \leq \tilde{t}_I(\alpha)$ ;
4 else if  $|S - S'| \leq t_I(\alpha)$  then return true;
5 else if  $|S - S'| \geq \Delta - t_I(\beta)$  then return false;
6 else /* indifference zone:  $t_I(\alpha) < |S - S'| < \Delta - t_I(\beta)$  */
7   return  $|V - V'| \leq t_I(\gamma)$ ;

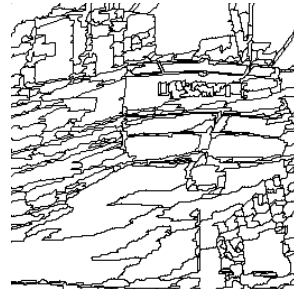
```

6 Discussion and Comparison with Previous Works

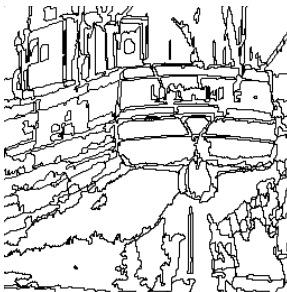
Our threshold cannot always be compared with the “theoretical” one proposed by Nock and Nielsen (2005) [10]. Since it is based on a different concentration inequality. However for the sake of completeness it may be of interest to compare both threshold. The analytic comparison is really uneasy, so we plotted this function for $|R| = |R'| = r$, ranging from 1 to 40 (there was no need to go beyond since $\theta(|r|)$ decreases quickly) and for different values of σ^2 ranging



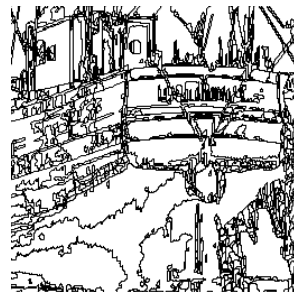
(a) original image



(b) $\alpha = .8$ $\beta = .5$ $\gamma = .1$ $\Delta = 8$



(c) with pre-segmentation



(d) original scanline

Fig. 1. Comparison with original scanline and without pre-segmentation

from 1000 to 16000. It turns out that our threshold is less sharp than the one by Nock and Nielsen only for very small regions and works better for small σ . We check that introducing the variance in the concentration inequality provides adaptivity and an extra amount of sharpness.

7 Experimental Results and Practical Improvements

Our main test is $t_I(\alpha)$ but it requires that $\sigma \neq 0$ and $\sigma' \neq 0$. If we look an image with low dynamic such as *cornouaille* image (see figure 1a)

and perform our segmentation process directly we obtain image at figure 1b which clearly shows effect of scanning order. In fact this effect is amplified by the fact that dynamic is low and so a lot of regions have initially null variance. To avoid this bias, we propose to pre-segment image by grouping very similar pixels into small regions by using same algorithm but with an Oracle looking only at greyvalue of pixels.

Result can be seen at figure 1c. It can be compare to result with original scanline process of 19 with same value of Δ presented figure 1d.

We have also done some comparisons with work of 6 which first introduce concentration inequality for segmentation and from which are derived numerous papers of R. Nock 12,8. For that, we use a very colorful image and perform

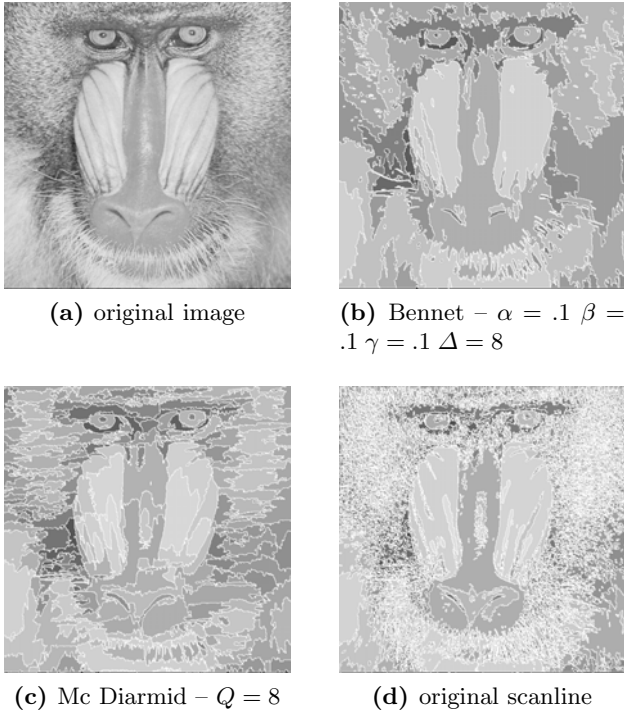


Fig. 2. Comparison with previous works

segmentation by applying our algorithm to each channel and comparing them as R. Nock in [5]. As algorithm is basically designed for greylevel images, we need to relax condition on α to have consistent results on the three RGB channels. So we set $\alpha = \beta = \gamma = 0.10$ and compare the result (see figure 2b with Mc Diarmid segmentation criterion in [6,12,8] with Q parameter set to 8 (see figure 2c).

8 Conclusion and Further Work

We proposed in this paper a merging criterion based on a statistical test using local properties but preserving global properties. Moreover an efficient region segmentation algorithm, based on this test, has been given. The originality of this work lies in the fact that we take into account the two classical types of error for statistical tests. This leads to an adaptive multi-test which decides if regions should be merged, should not, or if we are in indifference zone where both decisions could be accepted. For this last case, we have proposed a way to finally decide by introducing another parameter and looking at second order moments.

Thus the algorithm we propose requires the setting of four parameters $\alpha, \beta, \gamma, \Delta$. In fact, the two first parameters need not to be tuned precisely. They even can be fixed whatever is the image since they are related to the probability of making error and so denote the trust you want in your test. Classical value in statistical applications is 5%. γ and Δ are more sensitive parameters but are far more intuitive than the Q parameter of [6,12,8] since they describe the perceptual separability property of grey levels. Then, it has to be related to the image analyzed. We have proposed a first improvement by doing a pre-segmentation in order to decrease impact of Δ parameter and scanning order. Another way to improve results is to sort regions before merging them in order to reduce the order dependence as done in [12,8]. Instead of sorting, we could also to carry out seeded region merging. According to [19], the algorithm will loose its linear time complexity but will stay near linear as proved by R.E. Tarjan in [20] and so keep almost all of its efficiency. In the present paper, criteria are based on first and second order moment of the regions, but as `Scanline` algorithm scans regions starting from pixels, we could also take into account local pixel criteria to improve the test in the case of indifference zone.

References

1. Pavlidis, T.: Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Process* 1, 360–372 (1972)
2. Zucker, S.W.: Survey: Region growing: Childhood and adolescence. *Computer Vision, Graphics, and Image Processing* 5, 382–399 (1976)
3. Forsyth, D.A., Ponce, J.: *Computer Vision: A Modern Approach*. Prentice-Hall, Englewood Cliffs (2003)
4. Monga, O.: An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence* 1, 351–375 (1987)

5. Nock, R.: Fast and reliable color region merging inspired by decision tree pruning. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), vol. 1, pp. I-271-I-276 (2001)
6. Fiorio, C., Nock, R.: Image segmentation using a generic, fast and non-parametric approach. In: 10th IEEE International Conference on Tools with Artificial Intelligence, Taipei, Taiwan, R.O.C, pp. 450-458. IEEE Computer Society, Los Alamitos (1998)
7. Fiorio, C., Nock, R.: A concentration-based adaptive approach to region merging of optimal time and space complexities. In: British Machine Vision Conference, Bristol, England, vol. 2, pp. 775-784 (2000)
8. Nock, R., Nielsen, F.: Statistical region merging. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 1452-1458 (2004)
9. Lehmann, E.L.: Testing statistical hypotheses, 2nd edn. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, New York (1986)
10. Nock, R., Nielsen, F.: Semi-supervised statistical region refinement for color image segmentation. *Pattern Recognition: Image Understanding for Photographs* 38, 835-846 (2005)
11. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs; approximation and online algorithms. *Theoretical Computer Science* 361, 172-187 (2006)
12. Fiorio, C., Nock, R.: Sorted region merging to maximize test reliability. In: International Conference on Image Processing, Vancouver, Canada, vol. 01, pp. 808-811. IEEE, Los Alamitos (2000)
13. Mc Diarmid, C.: Concentration for independent permutations. *Comb. Probab. Comput.* 11, 163-178 (2002)
14. Hoeffding, W.: Probability inequalities for sums of bounded random variables. *Journal of American Statistical Association* 58, 13-30 (1963)
15. Bennett, G.: Probability inequalities for the sum of independent random variables. *Journal of American Statistical Association* 57, 33-45 (1962)
16. Bernstein, S.: On a modification of chebyshev's inequality and of the error formula of laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math.* 1 (1924)
17. Ledoux, M., Talagrand, M.: Probability in banach spaces. isoperimetry and processes. *Ergebnisse der Mathematik und ihrer Grenzgebiete* 3, xii+480 (1991)
18. Mc Diarmid, C.: Concentration. In: Habib, D.M., Ramirez-Alfonsin, R. (eds.) *Probabilistic Methods for Algorithmic Discrete Mathematics*, New-York, pp. 195-248. Springer, Heidelberg (1998)
19. Fiorio, C., Gustedt, J.: Two linear time Union-Find strategies for image processing. *Theoretical Computer Science* 154, 165-181 (1996)
20. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. *J. of the Association for Computing Machinery* 22, 215-225 (1975)

Segmentation for Hyperspectral Images with Priors

Jian Ye, Todd Wittman, Xavier Bresson, and Stanley Osher

Department of Mathematics, University of California, Los Angeles, CA 90095

Abstract. In this paper, we extend the Chan-Vese model for image segmentation in [1] to hyperspectral image segmentation with shape and signal priors. The use of the Split Bregman algorithm makes our method very efficient compared to other existing segmentation methods incorporating priors. We demonstrate our results on aerial hyperspectral images.

1 Introduction

A *hyperspectral image* is a high-dimensional image set that typically consists of 100-200 image channels. Each channel is a grayscale image that indicates the spectral response to a particular frequency in the electromagnetic spectrum. These frequencies usually include the visible spectrum of light, but most of the channels are focused in the infrared range. This allows a hyperspectral image to reveal features that are not visible in a standard color image. Each pixel in the image will have a spectral response vector that is the high-dimensional equivalent of the pixel's "color". Certain materials have a characteristic spectral signature that can be used to identify pixels containing that material. In an aerial hyperspectral scene an analyst could, for example, locate manmade materials or distinguish healthy vs. dead vegetation. For this reason, there is great interest in developing fast detection methods in hyperspectral imaging for applications such as aerial surveillance, mineral and agricultural surveys, chemical analysis, and medical imaging.

Unfortunately, due to the high dimensional complexity of the data, it is difficult to create accurate image segmentation algorithms for hyperspectral imagery. To improve the segmentation results, prior knowledge about the target objects can be incorporated into the segmentation model. In the spectral domain, a *spectral prior* is a vector specifying a target spectral signature that the pixels in the segmented object should contain. For example, one could specify the spectral signature of a particular mineral or type of biological tissue. This signature could be obtained from a spectral library or selecting a pixel from the image that is known to contain the desired material. In the spatial domain, a *shape prior* is a template binary image describing the outline of the desired targets. For example, in an aerial image one might use an airplane silhouette for automatically locating airplanes or in medical imaging one might enforce circular shapes to locate blood cells. An illustration of these two types of priors is shown in Fig. 1.

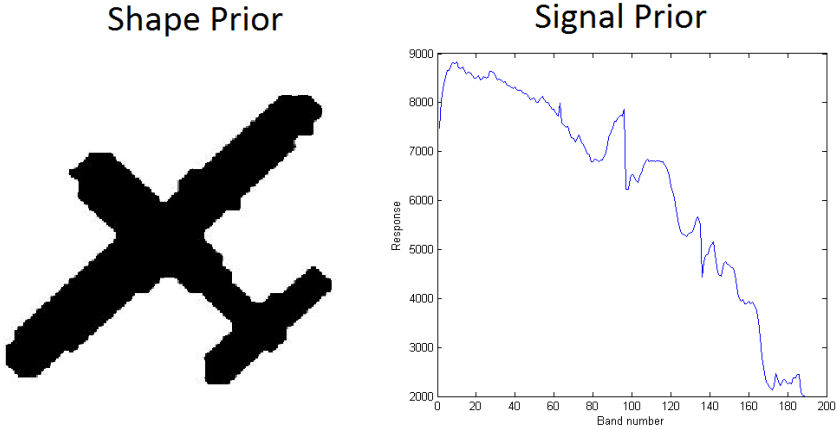


Fig. 1. Example priors. Left: Shape prior for outline of an airplane. Right: Signal prior for the hyperspectral signature of metal found in used in aircraft.

Image segmentation using shape priors has had great developments in recent years. Cremers, Osher and Soatto [2] incorporated statistical shape priors into image segmentation with the help of the level set representation. Their prior is based on an extension of classical kernel density estimators to the level set domain. They also propose an intrinsic registration of the evolving level set function which induces an invariance of the proposed shape energy with respect to translation. Using the level set representation, several other methods [3,4,5] have also been developed in recent years.

Bresson *et.al.* [4] extended the work of Chen *et. al.* [6] by integrating the statistical shape model of Leventon *et. al.* [7]. They propose the following energy functional for a level set function ϕ and grayscale image I :

$$F(\phi, \mathbf{x}_T, \mathbf{x}_{pca}) = \int_{\Omega} \{g_{\epsilon}(|\nabla I(x)|) + \beta \hat{\phi}^2(g_{\mathbf{x}_T}(x), \mathbf{x}_{pca})\} \delta(\phi) |\nabla \phi| d\Omega \quad (1)$$

where g_{ϵ} is a decreasing function vanishing at infinity. The first term is the geodesic active contours classical functional which detects boundaries with the edge detector g_{ϵ} . The second term measures the similarity of the shape to the zero level set of $\hat{\phi}(\mathbf{x}_T, \mathbf{x}_{pca})$.

Later on, Bresson *et. al.* [5] used the boundary information and shape prior driven by the Mumford-Shah functional to perform the segmentation. They propose the following functional:

$$F = \beta_s F_{shape}(C, \mathbf{x}_{pca}, \mathbf{x}_T) + \beta_b F_{boundary}(C) + \beta_r F_{region}(\mathbf{x}_{pca}, \mathbf{x}_T, u_{in}, u_{out}). \quad (2)$$

with

$$F_{shape} = \oint_0^1 \hat{\phi}^2(\mathbf{x}_{pca}, h_{\mathbf{x}_T}(C(q))) |C'(q)| dq, \quad (3)$$

$$F_{boundary} = \oint_0^1 g(|\nabla I(C(q))|) |C'(q)| dq, \quad (4)$$

$$F_{region} = \int_{\Omega_{in}(\mathbf{x}_{pca}, \mathbf{x}_T)} (|I - \mu_{in}|^2 + \mu |\nabla u_{in}|^2) d\Omega \\ + \int_{\Omega_{out}(\mathbf{x}_{pca}, \mathbf{x}_T)} (|I - \mu_{out}|^2 + \mu |\nabla u_{out}|^2) d\Omega \quad (5)$$

In the above functional, the first term is based on a shape model which constrains the active contour to retain a shape of interest. The second term detects object boundaries from image gradients. The third term globally drives the shape prior and the active contour towards a homogeneous intensity region.

Recently, Cremers *et al.* [8] used a binary representation of the shapes and formulated the problem as a convex functional with respect to deformations, under mild regularity assumptions. They proposed the following energy functional

$$E_i(q) = \int f(x)q(x)dx + \int g(x)(1 - q(x))dx + \int h(x)|\nabla q(x)|dx \quad (6)$$

The above first two terms are the integrals of f and g over the inside and outside of the shape, while the last term is the weighted Total Variation norm [9]. Ketut *et al.* [10] applied the technique of graph cuts to improve the algorithm runtime.

In this paper, we further improve the speed of the segmentation model with shape priors by using the Split Bregman method [10][11][12][13], a recent optimization technique which has its roots in works such as [14][15][16]. Also, we adapt the model to the hyperspectral images by the use of spectral angle distance and a signal prior.

2 Image Segmentation with Shape Priors

Variational methods have been widely used for the image segmentation problem. One of the most successful segmentation models is the Active Contour Without Edge (ACWE) model proposed by Chan and Vese [1]. To segment a grayscale image u_0 with a curve C , the authors proposed the following energy functional:

$$F(c^+, c^-, C) = \mu \cdot \text{Length}(C) + \lambda^+ \int_{\text{inside}(C)} |u_0(x, y) - c^+|^2 dx dy \quad (7) \\ + \lambda^- \int_{\text{outside}(C)} |u_0(x, y) - c^-|^2 dx dy$$

where $\text{Length}(C)$ is the length of the curve C and c^+ and c^- denote the average intensity value inside and outside the curve, respectively. This is a two-phase version of the Mumford-Shah model. The idea is that C will be a smooth minimal length curve that divides the image into two regions that are as close as possible to being homogeneous. Later on, Chan and Vese extended the model to vector valued images [17] as

$$F(\bar{c}^+, \bar{c}^-, C) = \mu \cdot \text{Length}(C) + \frac{1}{N} \sum_{i=1}^N \lambda_i^+ \int_{\text{inside}(C)} |u_0(x, y) - c^+|^2 dx dy \quad (8)$$

$$+ \frac{1}{N} \sum_{i=1}^N \lambda_i^- \int_{\text{outside}(C)} |u_0(x, y) - c^-|^2 dx dy$$

where λ_i^+ and λ_i^- are parameters for each channel, $\bar{c}^+ = (c_1^+, \dots, c_N^+)$ and $\bar{c}^- = (c_1^-, \dots, c_N^-)$ are two unknown constant vectors.

Chan *et al.* proposed a convexification of Chan-Vese model in [18]. Analogously, we can convexify the vectorial version of Chan-Vese model in this way:

$$E(u) = \min_{0 \leq u \leq 1} \int g |\nabla u| + \mu \langle u, r \rangle \quad (9)$$

where $r = \frac{1}{N} \sum_{i=1}^N [(c_1 - f_i)^2 - (c_2 - f_i)^2]$ and f_i is i -th band of the hyperspectral image with a total of N bands.

In order to constrain the geometry shape of the resulting object, we want to minimize the area difference of shape prior and resulting object up to an affine transformation. The proposed energy functional is as follows:

$$E(u) = \min_{0 \leq u \leq 1, \mathbf{x}_T} \int g |\nabla u| + \mu \langle u, r \rangle + \alpha |u - w| \quad (10)$$

where $w = h_{\mathbf{x}_T}(w_0)$, w_0 is the shape prior, and $h_{\mathbf{x}_T}$ is a geometric transformation parameterized by \mathbf{x}_T .

The above equation can be solved in an iterative way. It consists of following two steps.

Step 1: Update u .

Fix the prior w and its associated pose parameters \mathbf{x}_T and update u by using the fast Split Bregman algorithm.

To apply the Split Bregman algorithm, we make the substitutions $d_1 = \nabla u = (u_x, u_y)^\tau$, $d_2 = u - w$, $\mathbf{d} = (d_1^\tau, d_2)^\tau$, $F(u) = (u_x, u_y, u - w)^\tau$. To approximately enforce these equality constraints, we add two quadratic penalty functions. This gives rise to the unconstrained problem

$$(u^*, \mathbf{d}^*) = \arg \min_{0 \leq u \leq 1, \mathbf{d}} |d_1|_g + \alpha |d_2| + \mu \langle u, r \rangle + \frac{\lambda_1}{2} \|d_1 - \nabla u\|_2^2$$

$$+ \frac{\lambda_2}{2} \|d_2 - (u - w)\|_2^2 \quad (11)$$

Then we apply Bregman iteration to the unconstrained problem (11). This results in the following sequence of optimization problems:

$$\begin{aligned}
(u^{k+1}, d^{k+1}) &= \arg \min_{0 \leq u \leq 1, d} |d_1|_g + \alpha |d_2| + \mu \langle u, r \rangle & (12) \\
&+ \frac{\lambda_1}{2} \|d_1 - \nabla u - b_1^k\|_2^2 + \frac{\lambda_2}{2} \|d_2 - (u - w) - b_2^k\|_2^2 \\
b_1^{k+1} &= b_1^k + (\nabla u^k - d_1^k) \\
b_2^{k+1} &= b_2^k + (u^k - w - d_2^k) & (13)
\end{aligned}$$

where u can be solved by Gauss-Seidel iteration and d can be solved by shrinkage.

The whole algorithm for solving for u is as follows:

- 1: **while** $\|u^{k+1} - u^k\| > \epsilon$ **do**
- 2: Define $r^k = (c_1^k - f)^2 - (c_2^k - f)^2$
- 3: $u^{k+1} = GS_{GCS}(r^k, \mathbf{d}^k, \mathbf{b}^k)$
- 4: $\mathbf{d}^{k+1} = shrink_g(\nabla u^{k+1} + \mathbf{b}^k, \lambda)$
- 5: $\mathbf{b}^{k+1} = \mathbf{b}^k + \nabla u^{k+1} - \mathbf{d}^{k+1}$
- 6: Find $\Omega_k = \{x : u^k(x) > \mu\}$
- 7: Update $c_1^{k+1} = \frac{\int_{\Omega_k} f dx}{\int_{\Omega_k} dx}$, and $c_2^{k+1} = \frac{\int_{\Omega_k^c} f dx}{\int_{\Omega_k^c} dx}$
- 8: **end while**

Step 2: , fix u and update x_T .

The parameters we consider for affine transformations are rotation θ , translation T and scaling s . For affine transformations, we can express $x_{old} = x - D = sA(x - c) + T$. Here D is the displacement vector, A is a rotation matrix, T is the translation vector and s is the scaling factor. A is a function of the rotation angle θ .

$$A(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (14)$$

The derivative of the matrix A is

$$A_\theta(\theta) = \begin{bmatrix} -\sin \theta & -\cos \theta \\ \cos \theta & -\sin \theta \end{bmatrix} \quad (15)$$

If we fix u , then the original optimization problem (10) becomes

$$\begin{aligned}
E(x_T) &= \alpha |u - w| \\
&= \alpha |u - w_0(sA(x - c) + T)| & (16)
\end{aligned}$$

By the calculus of variations, we have

$$\begin{aligned}
T_t &= -\delta E / \delta T \\
&= -2\alpha \cdot \text{sign}(w_0(A(x - c) + T) - u(x)) (\nabla w_0(A(x - c) + T)) & (17)
\end{aligned}$$

$$\begin{aligned}
c_t &= -\delta E / \delta c \\
&= 2\alpha \cdot \text{sign}(w_0(A(x-c) + T) - u(x))A(\theta)^\tau * (\nabla w_0(A(x-c) + T))
\end{aligned} \tag{18}$$

$$\begin{aligned}
\theta_t &= -\delta E / \delta \theta \\
&= -2\alpha \cdot \text{sign}(w_0(A(x-c) + T) - u(x))\nabla w_0(A(x-c) + T) \cdot (A_\theta(\theta)(x-c))
\end{aligned} \tag{19}$$

$$\begin{aligned}
s_t &= -\delta E / \delta s \\
&= -2\alpha \cdot \text{sign}(w_0(A(x-c) + T) - u(x))\nabla w_0(A(x-c) + T) \cdot (A(\theta)(x-c))
\end{aligned} \tag{20}$$

The initialization of the above affine transformation parameters are: $c = \text{center}(u)$, $T = \text{center}(w)$, $s = 1$, where $\text{center}(u)$ denotes the center of the mass of u . The above procedure is repeated until convergence. For the pose parameter θ , since the energy functional is not convex in this parameter, in order to avoid the local minimum we usually try four different initial values and choose the one which leads to smallest minimum energy.

The alignment of shape prior and segmentation result u can be accelerated by adding an additional attraction term:

$$\min_{0 \leq u \leq 1, x_T} \int g|\nabla u| + \mu \langle u, r \rangle + \alpha|u - w| + \beta|w - f|^2 \tag{21}$$

The optimization is the same for finding u and the optimization for the affine transformation parameters will be similar to model (10).

3 Image Segmentation with Spectral Priors

One of the interesting properties of hyperspectral images is that for different materials, we have different spectral signatures. By combining both the spectral information and shape prior, we can segment some very challenging hyperspectral images.

The natural measure for distinguishing two different hyperspectral signatures v_1 and v_2 is the spectral angle:

$$\theta = \arccos\left(\frac{\|v_1 \cdot v_2\|_2}{\|v_1\|_2 \|v_2\|_2}\right). \tag{22}$$

By using the spectral angle, the original optimization problem (10) can be rewritten as

$$E(u) = \min_{0 \leq u \leq 1, \mathbf{x}_T} \int g|\nabla u| + \mu \langle u, r \rangle \tag{23}$$

where $r(i, j) = \theta(\mathbf{c}_1(i, j), \mathbf{f}(i, j)) - \theta(\mathbf{c}_1(i, j), \mathbf{f}(i, j))$ for $i = 1 : Nx, j = 1 : Ny$.

A signal prior is a hyperspectral signature that we want our segmented object to contain. If we want to use the signal prior c_p , then we will have $r(i, j) = \theta(c_p(i, j), \mathbf{f}(i, j)) - \theta(c_b(i, j), \mathbf{f}(i, j))$ for $i = 1 : Nx, j = 1 : Ny$. The signal prior can be obtained from a known spectral library or by selecting a pixel from the image with the desired signature.

If both a signal and a shape prior are used, we can use the segmentation from spectral prior as an initialization for u , and apply the shape prior model (21) to do the final segmentation.

4 Results

Fig. 2 shows a synthetic 100-band hyperspectral image of two overlapping ellipses with different spectral signatures. Without using any priors, the multi-dimensional Chan-Vese model will segment the entire shape from the background. Incorporating priors, we can force the segmentation of a specified material or a given template shape. Note that when we use both a signal and shape prior, the segmentation finds the shape in the image that contains the maximum amount of the specified material signature.

Fig. 3 demonstrates more clearly the advantage of using priors. While algorithms using only spectral information can pick up most of the airplane, it separates the segmentation into two pieces. Thus, an algorithm can be reasonably certain to pick up the material of an airplane, it can not make the conclusion that an actual airplane has been detected. Using shape priors, however, will actually make the judgement that an airplane has been found. Fig. 3 shows that the algorithm can still operate with some mismatches in the shape of the airplane. However, because the shape prior is a different type of airplane than the one under consideration, the segmentation contour does not match the airplane outline as well as the segmentation using only the signal prior. This is meant to illustrate that shape priors need to be used carefully, as the resulting contour may fit the prior well but not the data.

Fig. 4 shows the segmentation result for detecting multiple airplanes in a 224-band hyperspectral image of Santa Monica Airport. From the initial

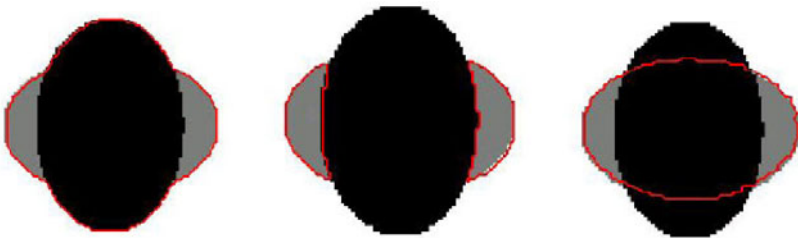


Fig. 2. Segmentation of a synthetic hyperspectral image. Left: Segmentation without priors. Center: Segmentation using signal prior of gray material. Right: Segmentation incorporating signal prior and an ellipse shape prior.

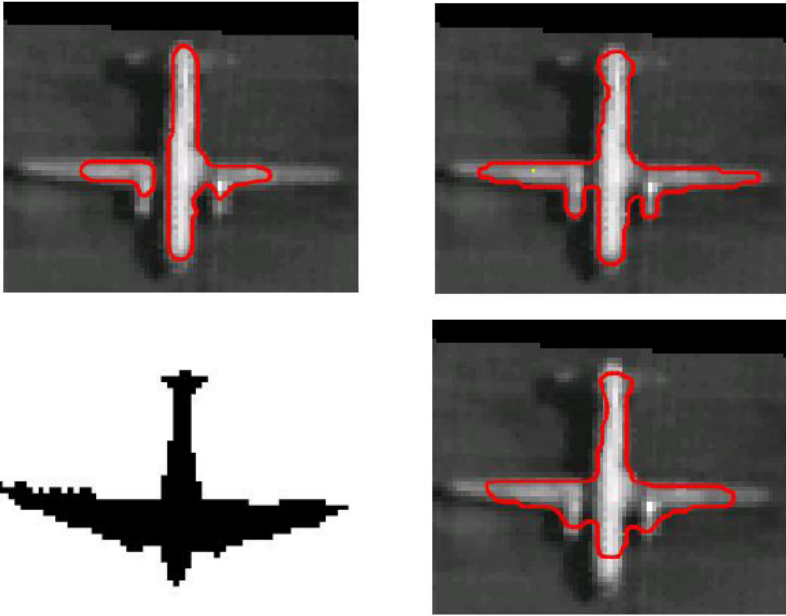


Fig. 3. Segmentation of a single object. Top left: Segmentation without priors. Top right: Segmentation using a metal signal prior. Bottom left: Airplane shape prior. Bottom right: Segmentation using both signal and shape priors.



Fig. 4. Segmentation of multiple objects in hyperspectral image of Santa Monica Airport. Left: Segmentation without priors. Right: Segmentation using a metal signal prior and the airplane shape prior.

Chan-Vese segmentation result, we take out the planes that we are interested in one by one. For each plane, we choose a rectangular box to enclose the plane. And then we do the segmentation with the shape prior for each plane. At the end, we combine all the segmentation results together to get the final result for the whole image.

5 Conclusion

We have demonstrated the segmentation results for both shape and signal priors for synthetic images and hyperspectral images. With the introduction of the Split Bregman method, we can solve the optimization problem more quickly than other segmentation methods incorporating priors. Our algorithm is efficient and also robust to different kinds of images. Further research could involve applications to mapping and remote sensing, learning the priors from the data, and extending the results to handle multiple priors such as shape or spectral libraries.

Acknowledgements

The authors would like to thank Andrea Bertozzi and Tom Goldstein for their support and advice. This work was supported by the US Department of Defense.

References

1. Chan, T., Vese, L.: Active contours without edges. *IEEE Trans. on Image Processing* 10, 266–277 (2001)
2. Cremers, D., Osher, S., Soatto, S.: Kernel density estimation and intrinsic alignment for knowledge-driven segmentation. In: Rasmussen, C.E., Bühlhoff, H.H., Schölkopf, B., Giese, M.A. (eds.) *DAGM 2004*. LNCS, vol. 3175, pp. 415–438. Springer, Heidelberg (2004)
3. Chan, T., Zhu, W.: Level set based shape prior segmentation. In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 1164–1170 (2005)
4. Bresson, X., Vandergheynst, P., Thiran, J.P.: A priori information in image segmentation: energy functional based on shape statistical model and image information. In: *IEEE Int. Conf. Image Processing*, vol. 2, pp. 425–428 (2003)
5. Bresson, X., Vandergheynst, P., Thiran, J.P.: A variational model for object segmentation using boundary information and shape prior driven by the Mumford-Shah functional. *Int. J. Computer Vision* 68, 145–162 (2006)
6. Chen, Y., Tagare, H., Thiruvankadam, S., Huang, F., Wilson, D., Gopinath, K., Briggsand, R., Geiser, E.: Using prior shapes in geometric active contours in a variational framework. *Int. J. Computer Vision* 50, 315–328 (2002)
7. Leventon, M., Grimson, W., Faugeras, O.: Statistical shape influence in geodesic active contours. In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 316–323 (2000)
8. Cremers, D., Schmidt, F., Barthel, F.: Shape priors in variational image segmentation: Convexity, Lipschitz continuity and globally optimal solutions. In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2008)

9. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60, 259–268 (1992)
10. Fundana, K., Heyden, A., Gosch, C., Schnorr, C.: Continuous graph cuts for prior-based object segmentation. In: *Int. Conf. on Pattern Recognition*, pp. 1–4 (2008)
11. Bresson, A., Esedoglu, S., Vanderghynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. *J. Mathematical Imaging and Vision* 28, 151–167 (2007)
12. Burger, M., Osher, S., Xu, J., Gilboa, G.: Nonlinear inverse scale space methods for image restoration. *Communications in Mathematical Sciences* 3725, 25–36 (2005)
13. Goldstein, T., Osher, S.: The split Bregman method for l1 regularized problems. *SIAM J. Imaging Science* 2, 323–343 (2009)
14. Peaceman, D., Rachford, J.: The numerical solution of parabolic and elliptic differential equations. *J. of Soc. for Ind. and Applied Math.* 3, 28–41 (1955)
15. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Math. with Applications* 2, 17–40 (1976)
16. Glowinski, R., Le Tallec, P.: *Augmented Lagrangian methods for the solution of variational problems*
17. Chan, T., Sandberg, B., Vese, L.: Active contours without edges for vector-valued images. *J. Visual Communication and Image Representation* 11, 130–141 (2000)
18. Chan, T., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Applied Mathematics* 66, 1632–1648 (2006)

The Curve Filter Transform – A Robust Method for Curve Enhancement

Kristian Sandberg

Computational Solutions, LLC
1800 30th St, Suite 210B
Boulder, CO 80301

<http://www.computationalsolutions.com>

Abstract. In this paper we introduce the Curve Filter Transform, a powerful tool for enhancing curve-like structures in images. The method extends earlier works on orientation fields and the Orientation Field Transform. The result is a robust method that is less sensitive to noise and produce sharper images than the Orientation Field Transform. We describe the method and demonstrate its performance on several examples where we compare the result to the Canny edge detector and the Orientation Field Transform. The examples include a tomogram from a biological cell and we also demonstrate how the method can be used to enhance handwritten text.

1 Introduction

In this paper we improve upon previous work on using orientation fields to enhance curve-like objects in images. The method is based on the Orientation Field Transform (OFT, see [5], [7], [8]), but introduces a new type of transform that significantly improves the sharpness of the enhanced features while being less sensitive to noise.

The OFT was originally developed for enhancing membranes in tomograms of biological cells, but the method is general and can be applied to any image where curve-like objects need to be enhanced. The OFT has the property of not only enhancing curve-like structures, but also suppress point-like structures.

Other methods for enhancing curve-like structures includes the Hough transform (see [1] and references therein), but whereas the Hough transform works best to detect global objects, the OFT is local in nature. The Canny edge detector [3] can be used to detect general discontinuities, including edges of curve-like structures. However, it does not distinguish between point-like and curve-like structures, and requires significant blurring in noisy environments, which leads to poor localization of the edges.

Anisotropic diffusion methods (see e.g., [9]) are well suited for de-noising curve-like objects without blurring the edges. However, as noted in [6] and [2], these methods does not distinguish between point-like and curve-like objects. Anisotropic diffusion methods typically involves solving computationally expensive non-linear PDEs, and also tend to rely on tuning several parameters to achieve good results.

The introduction of one or more user controlled parameters can often improve the processed image. However, parameters can also offset the benefit of automated image enhancement. A user needs to have a good understanding of the effect of the parameters, and too many non-intuitive parameters is a common obstacle for wide-spread adaptation of automated image processing algorithms in laboratory settings.

The method in this paper only requires one parameter, which corresponds to the typical width of the targeted structure. Hence, the same method can easily be adapted to a broad variety of situations, and is easy for a new user to learn.

In this paper we introduce the Curve Field Transform (CFT), which uses the idea of orientation fields [4] and the OFT to generate what we will refer to as a curve field. A curve field assigns a weighted curve segment to each location in the image. Each such curve segment has a support that typically extends over many pixels in the image such that each pixel is covered by multiple curve segments from nearby pixels. The CFT sums the weights of all curve segments overlapping a given pixel. The CFT relies on computing the curve field in a way that captures the geometric structures of curve-like objects in the image.

The paper is outlined as follows. We begin by presenting the main idea of the method. In the following two sections we provide the details for generating the curve field and how to apply the CFT. We next provide several examples including a synthetic image, a slice from a tomogram of a biological cell, and a noisy image where we extract hand written text. We conclude the paper with a discussion of the method and a conclusion.

2 Idea

In this section, we outline the idea behind the CFT. In the following two sections, we provide a more detailed description of the method.

A *curve field* is the assignment of a parameterized curve segment to each location of an image. With each such curve we associate a (scalar) weight that indicates the “importance” of the curve segment. The lengths of the curve segments are typically fixed and chosen as roughly twice the width of a typical curve-like structure in the image.

The first step of our method is to generate a curve field with the following two properties:

- For a pixel located on a curve-like structure, the associated curve segment should align with the underlying structure, and have a large weight.
- For a pixel far away from a curve-like structure, the associated weight should be small.

We generate a curve field with these two properties using a two-step process. In the first step, we go through each pixel of the image and look for the direction along which the image contains the largest mean intensity. This generates what we will refer to as a primary curve field where the curve segment at each pixel can be represented by an angle and a weight that gives the mean intensity along the line (see Figure 1 a and b).

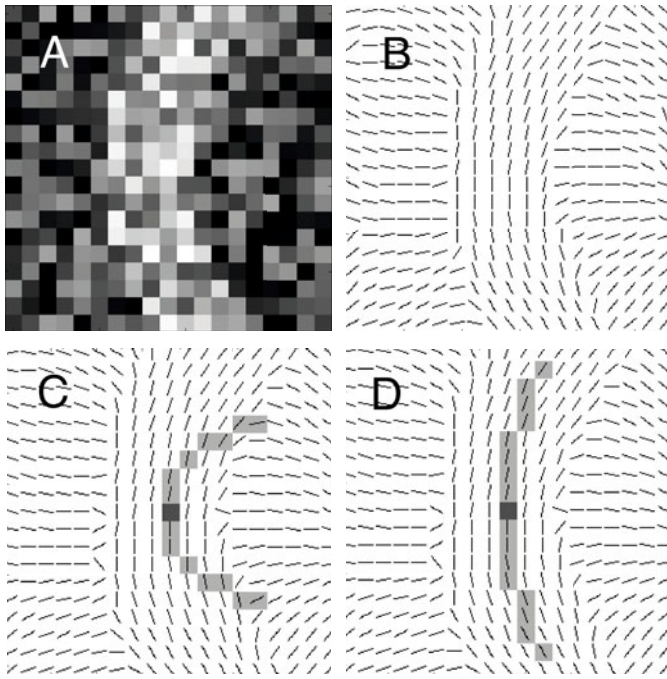


Fig. 1. a) Original image. b) Illustration of the primary curve field. c) Measuring the alignment along the gray curve (high curvature). d) Measuring the alignment along the gray curve (small curvature). In this example, the curve in d) aligns best with the underlying primary curve field, and the secondary curve field at the center pixel is therefore assigned the gray curve in d). Note that in the above figures, we have for clarity plotted each curve segment of the primary curve field as having a length of one pixel. In practice, each curve extends over several pixels. Curve segments have also been plotted with uniform intensity, although in practice the intensity (weight) may vary between curve segments (see also Figure 2 below).



Fig. 2. Illustration of the CFT. The lines illustrate the secondary curve field for the image in Figure 1a. (For clarity, we only plot straight curves.) In this figure, each curve segment is plotted with an intensity that reflects the weight associated with the curve segment. The CFT is computed by at each pixel, summing the weight for each curve segment that intersect the pixel.

In the second step, we look for curves along which the primary curve field from the first step shows strong alignment. We record the curve along which we find the best net alignment, and record the net alignment value as the weight (see Figure Figure 1 c and d). The new curve and weight from this step is referred to as the *secondary curve field*.

Once the secondary curve field has been generated, we apply the CFT as follows: At each pixel we look for all curve segments that overlap a pixel, and sum the weights from these curves (see Figure 2).

3 Generating the Curve Field

We denote a curve field at location $\mathbf{x} = (x, y)$ as $\mathcal{F}(x, y) = \{w(x, y), \mathbf{r}(x, y, t)\}$ where w is a (scalar) weight and $\mathbf{r}(x, y, t)$ denotes a parameterized curve centered at (x, y) and parameterized with parameter $t \in [-\frac{L}{2}, \frac{L}{2}]$ such that $\mathbf{r}(x, y, 0) = (x, y)$. Here L denotes the length of the curve and is typically chosen as roughly twice the width of a typical curve-like structure in the image. For clarity, we will sometimes omit the t argument and simply write $\mathbf{r}(x, y)$.

We first generate what we will refer to as the *primary curve field* by using the method in [5] or [7]. In this section, we will for simplicity review the simpler method in [5] but note that the somewhat more complicated method in [7] gives significantly better results.

Let us first consider a family of straight lines $\mathbf{r}_\theta(x, y, t)$ of length L indexed by the *orientation angle* θ such that

$$\mathbf{r}_\theta(x, y, t) = (x + t \cos \theta) \hat{\mathbf{i}} + (y + t \sin \theta) \hat{\mathbf{j}}, \quad t \in [-\frac{L}{2}, \frac{L}{2}].$$

Let $I(x, y)$ denote the image. The primary curve field \mathcal{F}_1 is given by $\mathcal{F}_1 = \{w(x, y), \mathbf{r}_\theta(x, y)\}$ where

$$w(x, y) = \max_{\theta} \int_{-\frac{L}{2}}^{\frac{L}{2}} I(\mathbf{r}_\theta(x, y, s)) ds$$

and

$$\tilde{\theta}(x, y) = \arg \max_{\theta} \int_{-\frac{L}{2}}^{\frac{L}{2}} I(\mathbf{r}_\theta(x, y, s)) ds.$$

An example of a primary curve field is illustrated in Figure 1a and b.

In order to generate what we will refer to as the *secondary curve field*, we let $\{\mathbf{r}_k(x, y, t)\}_k$ denote a generic family of parameterized curves indexed by k . In order to define more complicated curves, these curves may include more parameters than the orientation angle θ used above. We can, for example, for each tangent angle θ also vary the curvature for the curve (see [8] for details).

Let w_1 and θ_1 denote the weight and orientation angle of the primary curve field \mathcal{F}_1 . We now consider a curve $\mathbf{r}_k(\mathbf{x}, t)$ and introduce the *alignment integral*

$$\Omega[\mathcal{F}_1](\mathbf{x}, k) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} w_1(\mathbf{r}_k(\mathbf{x}, s)) \cos(2(\theta_1(\mathbf{r}_k(\mathbf{x}, s)) - \nu_k(s))) \left\| \frac{d\mathbf{r}_k}{ds} \right\| ds \tag{1}$$

where ν_k denotes the tangent vector $\frac{d\mathbf{r}_k}{dt}$ mapped to the interval $[0, 180^\circ)$ ¹

Using the alignment integral we define the secondary curve field \mathcal{F}_2 as $\mathcal{F}_2(x, y) = \{w(x, y), \mathbf{r}_{\tilde{k}}(x, y)\}$ where

$$\tilde{k} = \arg \max_k |\Omega[\mathcal{F}_1](x, y, k)|$$

and

$$w(x, y) = \Omega[\mathcal{F}_1](x, y, \tilde{k}).$$

This alignment integral can be compared to a work integral over a vector field in physics represented in polar coordinates (see [8] for more details).

In order to interpret the alignment integral, we note that the integrand in (1) gives a large positive response if the tangent of $\mathbf{r}_k(\mathbf{x}, s)$ aligns with the curve segment of the underlying primary curve field. The curve $\mathbf{r}_{\tilde{k}}(x, y, t)$ therefore represents the curve that aligns best with the underlying primary curve field (see Figure 1c and d).

We can visualize the secondary curve field as a curve segment of length L assigned to every pixel, with a “density” (weight) associated with the curve segment. The shape of the curve segment typically mimics the underlying structure of the original image in a neighborhood of a pixel. However, in areas without any significant curve-like structures, the shape of the curve does not matter, as the weight will be close to zero at such locations²

The reason for introducing the secondary curve field is that it measures correlations in orientations, which are less sensitive to noise than correlations in intensity.

4 The Curve Field Transform

Once the secondary curve field has been computed, the weight of the curve segment associated with each pixel is added to the output image for all pixels along the curve segment. Formally, we define the curve filter transform as

$$\mathcal{C}[\mathcal{F}](\mathbf{x}) = \sum_{\mathbf{x}' \in \mathcal{D}} \chi_{\mathbf{r}(\mathbf{x}')}(\mathbf{x}) w(\mathbf{x}')$$

where \mathcal{D} denotes the image domain, $\{w(\mathbf{x}'), \mathbf{r}(\mathbf{x}')\}$ is the secondary curve field, and χ denotes the characteristic function defined by

$$\chi_{\mathbf{r}(\mathbf{x}')}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathbf{r}(\mathbf{x}') \\ 0, & \mathbf{x} \notin \mathbf{r}(\mathbf{x}') \end{cases}.$$

¹ As opposed to vectors, orientation angles lack a sense of “backward/forward”, and are only defined for angles in the range $[0, 180^\circ)$. If the tangent vector has an angle in the interval $[180^\circ, 360^\circ)$, we map such angles to $[0, 180^\circ)$ by subtracting 180° .

² This assumes that the weights are computed according to the method in [7].

Although $\mathbf{r}(\mathbf{x}, t)$ was defined as a vector valued function in Section 3, when using it as an argument to χ we will treat it as a set defined by the values $\mathbf{r}(\mathbf{x}, t)$, $t \in [-\frac{L}{2}, \frac{L}{2}]$.

In practice, we compute the output of the CFT as follows. We first discretize the curve parameter t as the array $\mathbf{t}[0], \mathbf{t}[1], \dots, \mathbf{t}[N-1]$, and then discretize the curve field by introducing the two arrays $\mathbf{r}[\mathbf{x}, \mathbf{y}, \mathbf{t}[k]]$ and $\mathbf{w}[\mathbf{x}, \mathbf{y}]$. The CFT is now computed by the following algorithm:

1. `output = 0`
2. `for all pixels (x,y):`
 - `for k=1, ..., N`
 - `output[r[x,y,t[k]]] += w[x,y]`

The outer loop loops over the discretized curve field $\{\mathbf{w}[\mathbf{x}, \mathbf{y}], \mathbf{r}[\mathbf{x}, \mathbf{y}, \mathbf{t}]\}$, while the inner loop loops over all pixel that each curve segment overlaps. For each pixel that the curve segment $\mathbf{r}[\mathbf{x}, \mathbf{y}, \mathbf{t}]$ overlaps, we add the weight $\mathbf{w}[\mathbf{x}, \mathbf{y}]$ of the curve segment to the output.

5 Results

In this section we apply the method described above, but use the method from 7 to compute the weights and orientation angles for the primary curve field.

In our first example we demonstrate the method for a synthetically generated test image consisting of an S-curve and a point with noise added (Figure 3a). The

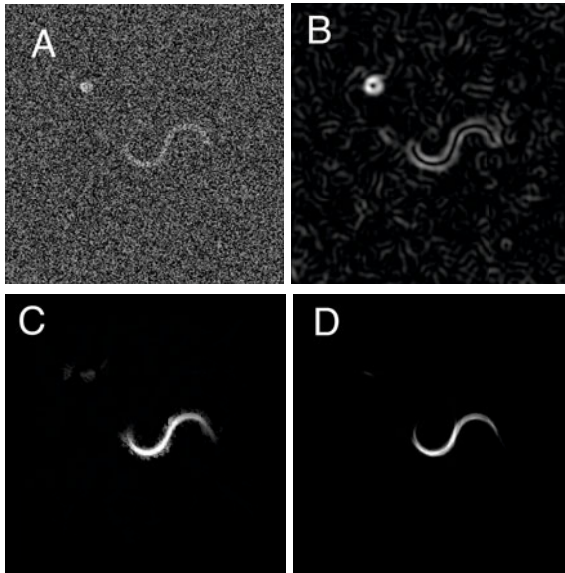


Fig. 3. a) Original image. b) Result using the Canny edge detector. c) Result using the OFT. d) Result using the CFT.

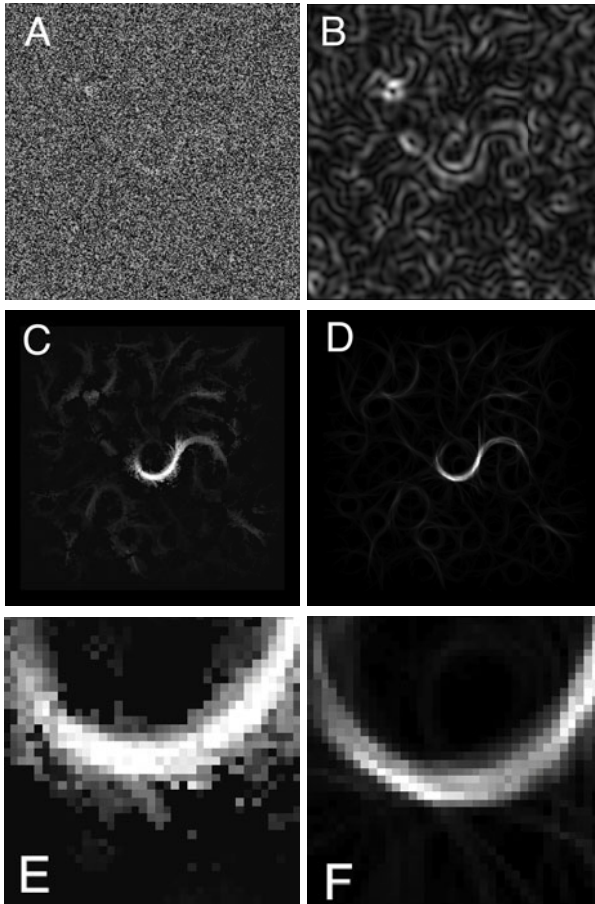


Fig. 4. a) Original image. b) Result using the Canny edge detector. c) Result using the OFT. d) Result using the CFT. e) Close up of Figure 4c (OFT method). f) Close up of Figure 4d (CFT) method. We note that the CFT gives significantly sharper result.

point has twice the intensity of the S-curve. The goal is to enhance the S-curve while suppressing the point in order to illustrate the geometrical aspect of the method. In Figure 3 b-d we compare the result of using the Canny edge detector, the OFT algorithm from [8], and the CFT introduced in this paper. For the result using the Canny edge detector, we used a Gaussian blurring kernel combined with the Sobel edge detector mask [3]. We tuned the width of the Gaussian kernel to produce a result that was a compromise between denoising and good edge

³ The Canny edge detector typically includes an additional step where a thinning operation is applied to the edge. As this paper focus on the enhancement step rather than the segmentation problem, we have omitted the thinning step, but note that this step could be applied to both the Canny method and the CFT method.

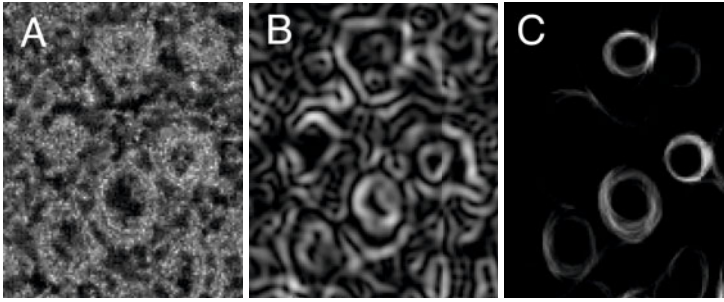


Fig. 5. a) Original section from a tomographic slice of a T-lymphocyte. b) Result after applying the Canny edge detector. c) Result after applying the CFT.

location. From the result in Figure 3 we note that the Canny edge detector cannot distinguish between the point and the curve, but detects the edges of both objects. Despite the inherent denoising in the Canny algorithm, we note that it is more sensitive to noise than both the OFT and the CFT methods, as it picks up more false positives from the background. Comparing Figure 3c and d, we also note that the CFT method produces a sharper image than the OFT method.

In Figure 4 we show the same comparison, but now with more noise added. We see that the CFT is less sensitive to the noise than the other methods, while also producing a sharper image than the OFT method. This is illustrated further in Figure 4e and f, where we have enlarged the bottom of the S-curve for the OFT and the CFT results from Figure 4c and d. We note how the curve is significantly sharper for the CFT method.

In the next example (Figure 5), we look at a section from a tomographic slice of a T-lymphocyte. This slice is a challenging data where the circular structures of interest are surrounded by complex structures that is a problem for most segmentation algorithms. Although the Canny edge detector does pick up the four main structures in the image, it cannot separate them from the complex structures in the background. The CFT enhances the structures of interest, while suppressing the surrounding structures. This is a most important property, as there is a huge interest in segmentation methods for biological cells [6].

In our final example, we consider a handwritten text with a few dots added to simulate ink stains (Figure 6a). The text was digitized at low resolution such that the width of the line is only 1-2 pixels wide. In Figure 6b we have added noise to the image. The noise combined with the coarse digitization makes this a challenging enhancement problem. We apply the CFT to the noisy image in Figure 6b and observe that the method suppresses the points and noise while capturing the text. Suppressing the points is important for hand recognition algorithms, which may be offset by such point-like objects.

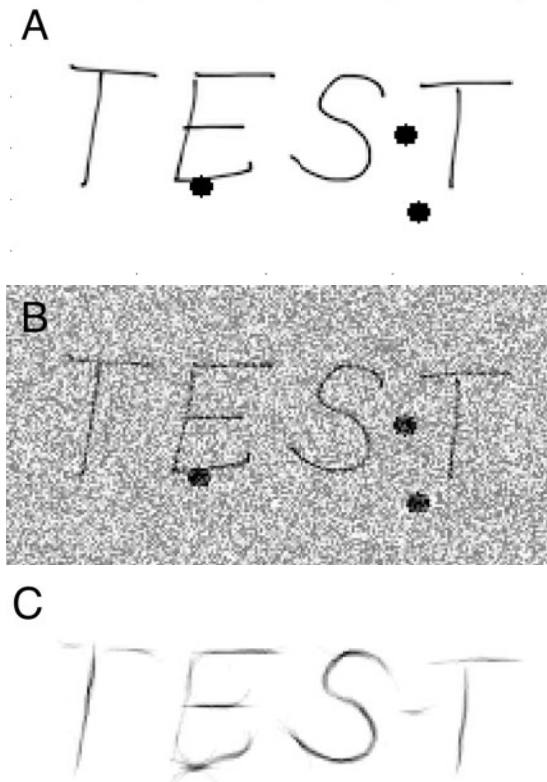


Fig. 6. a) Original hand-written note with simulated ink stains added. b) Same as the top image, but with noise added. c) Enhanced image after applying the CFT to the noisy image in b).

6 Discussion

The examples in the previous section illustrate a most important property of the CFT: The ability to enhance curves based on geometry rather than contrast. This ability makes the method ideal in situations where an image has to be automatically segmented, as many segmentation algorithms are hurt by the presence of high contrast point-like objects. The reason why the CFT works so well in this respect, is that focuses on orientations rather than intensity information in the image.

We also note that the CFT is forgiving for high noise levels. One reason for this is that the main operations are based on addition, rather than notoriously unstable subtraction operations used in typical edge detection methods. The method also does not require any pre-processing in terms of denoising, but denoising is built-in to the algorithm. The nature of the denoising is also anisotropic, which prevents blurring of the edges.

Finally, we stress that the proposed method only requires one parameter to be tuned. The parameter is related to the typical thickness of the structures to be enhanced and the method is therefore trivial to learn for a new user. Also,

the same method works well for a wide variety of problems. In this paper we applied the same method for enhancing cell membranes in a tomogram for a biological cell, and to enhance the text on a hand written note. Despite these quite different areas of application, the same algorithm could be used for both applications. In fact, the method is quite insensitive to variations in the one and only parameter used by the algorithm, and we were able to use the same parameter value for both applications.

7 Conclusion

We have presented a new method for enhancing curve-like structures in images. The new method improves on the previously suggested method using the Orientation Field Transform (OFT), but is less sensitive to noise and gives sharper images than the OFT. We have demonstrated the method's excellent performance in enhancing curves in noisy images, and its ability to suppress high contrast point-like objects.

Acknowledgment

The author would like to thank the Boulder Laboratory for 3-D Electron Microscopy of Cells at University of Colorado for providing the tomogram used in the paper.

References

1. Ballard, D.H.: Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition* 13, 111–112 (1981)
2. Brega, M.: Orientation Fields and Their Application to Image Processing. Master's Thesis, University of Colorado at Boulder (2005)
3. Canny, J.: A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis* 8, 679–698 (1986)
4. Gu, J., Zhou, J.: A novel model for orientation field of fingerprints. In: *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 493–498 (2003)
5. Sandberg, K., Brega, M.: Segmentation of thin structures in electron micrographs using orientation fields. *J. Struct. Biol.* 157, 403–415 (2007)
6. Sandberg, K.: Methods for image segmentation in cellular tomography. In: McIntosh, J.R. (ed.) *Methods in Cell Biology: Cellular Electron Microscopy*, vol. 79, pp. 769–798. Elsevier, Amsterdam (2007)
7. Sandberg, K.: Curve enhancement using orientation fields. In: Bebis, G. (ed.) *ISVC 2009, Part 1. LNCS*, vol. 5875, pp. 564–575. Springer, Heidelberg (2009)
8. Sandberg, K.: The Generalized Orientation Field Transform. In: Barneva, R.P., et al. (eds.) *Object Modeling, Algorithms, and Applications*, pp. 107–112. Research Publishing (2010)
9. Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart (1998)

Split Bregman Method for Minimization of Region-Scalable Fitting Energy for Image Segmentation^{*}

Yunyun Yang^{1,2}, Chunming Li³, Chiu-Yen Kao^{1,4,**}, and Stanley Osher⁵

¹ Department of Mathematics, The Ohio State University, OH 43202, U.S.

² Department of Mathematics, Harbin Institute of Technology, Harbin, 150001, China

³ Department of Radiology, University of Pennsylvania, PA 19104, U.S.

⁴ Mathematical Biosciences Institute, The Ohio State University, OH 43210, U.S.

Tel.: 614-292-8609; Fax: 614-292-1479

kao@math.ohio-state.edu

⁵ Department of Mathematics, University of California, Los Angeles, CA 90095, U.S.

Abstract. In this paper, we incorporate the global convex segmentation method and the split Bregman technique into the region-scalable fitting energy model. The new proposed method based on the region-scalable model can draw upon intensity information in local regions at a controllable scale, so that it can segment images with intensity inhomogeneity. Furthermore, with the application of the global convex segmentation method and the split Bregman technique, the method is very robust and efficient. By using a non-negative edge detector function to the proposed method, the algorithm can detect the boundaries more easily and achieve results that are very similar to those obtained through the classical geodesic active contour model. Experimental results for synthetic and real images have shown the robustness and efficiency of our method and also demonstrated the desirable advantages of the proposed method.

Keywords: split Bregman, region-scalable model, image segmentation, intensity inhomogeneity.

1 Introduction

Image segmentation [1,2,3,4] is a fundamental and important task in image analysis and computer vision. Most of existing methods for image segmentation can be categorized into two classes: region-based methods [1,5,6,7,8,9] and edge-based methods [3,4,10,11,12,13]. In general, the region-based methods are more robust than the edge-based methods. However, the former type of methods [1,5,6,7] typically relies on the homogeneity of the image intensities, which is often not satisfied by real world images.

Intensity inhomogeneity has been a challenging difficulty for region-based methods. It often occurs in real images from different modalities such as medical

^{*} C.Y.K. is partially supported by NSF DMS-0811003 grant and Sloan Fellowship. S.O. is supported by an ARO MURI subcontract through the University of South Carolina.

^{**} Corresponding author.

images. Segmentation of such medical images usually requires intensity inhomogeneity correction as a preprocessing step [14]. Intensity inhomogeneity can be addressed by more sophisticated models than piecewise constant (PC) models. Two piecewise smooth (PS) models were proposed in Vese and Chan [9] and Tsai et al. [8] independently, aiming at minimizing the Mumford-Shah functional [15]. These PS models have exhibited certain capability of handling intensity inhomogeneity. However, they are computationally expensive and suffer from other difficulties. Michailovich et al. [16] proposed an active contour model which does not rely on the intensity homogeneity and, therefore, to some extent, overcomes the limitation of PC models. Recently, Li et al. [17] proposed an efficient region-based model, called a region-scalable fitting (RSF) energy model, which is able to deal with intensity inhomogeneities. The RSF model was formulated in a level set framework, which is quite sensitive to contour initialization.

Recently the split Bregman method has been adopted to solve image segmentation more efficiently. This method has the advantage that it does not require regularization, continuation or the enforcement of inequality constraints and therefore is extremely efficient. Several applications of the split Bregman method are Rudin-Osher-Fatemi (ROF) denoising [18,19,20] and image segmentation [21,22]. In [19], they applied this technique to the ROF functional for image denoising and to a compressed sensing problem that arose in Magnetic Resonance Imaging. In [22], the authors applied the split Bregman concept to image segmentation problems and built fast solvers. However, this method was based on the PC model and thus it was not able to deal with intensity inhomogeneity.

In this paper, we incorporate the global convex segmentation (GCS) method and the split Bregman technique into the RSF model [17]. We first drop the regularization term of the original gradient flow equation in [17]. Following the idea in Chan et al. [23], we then get a simplified flow equation which has coincident stationary solution with the original one. In order to guarantee the global minimum, we restrict the solution to lie in a finite interval. Then we modify the simplified energy to incorporate information from the edge by using a non-negative edge detector function, and get results that are very similar to those obtained through the classical geodesic active contour (GAC) model [10]. We thus apply the split Bregman method to the proposed minimization problem of region-scalable fitting energy for segmentation and demonstrate many numerical results. As a result, the proposed algorithm can be used to segment images with intensity inhomogeneity efficiently.

The remainder of this paper is organized as follows. We first review some well-known existing region-based models and their limitations in Section 2. The new proposed method is introduced in Section 3. The implementation and results of our method are given in Section 4. This paper is concluded in Section 5.

2 Region-Based Active Contour Models

2.1 Chan-Vese Model

Chan and Vese [1] proposed an active contour approach to the Mumford-Shah problem [15]. This image segmentation method works when the image consists of

homogeneous regions. Let $\Omega \subset \mathbb{R}^2$ be the image domain, and $I : \Omega \rightarrow \mathbb{R}$ be a given gray level image, the idea is to find a contour C which segments the given image into non-overlapping regions. The model they proposed is to minimize the following energy:

$$\mathcal{F}^{CV}(C, c_1, c_2) = \lambda_1 \int_{outside(C)} |I(x) - c_1|^2 dx + \lambda_2 \int_{inside(C)} |I(x) - c_2|^2 dx + \nu |C|, \tag{1}$$

where λ_1, λ_2 and ν are positive constants, $outside(C)$ and $inside(C)$ represent the regions outside and inside the contour C , respectively, c_1 and c_2 are two constants that approximate the image intensities in $outside(C)$ and $inside(C)$, and $|C|$ is the length of the contour C . The optimal constants c_1 and c_2 that minimize the above energy turn out to be the averages of the intensities in the entire regions $outside(C)$ and $inside(C)$, respectively. For inhomogeneous images, as demonstrated in [17,24], the PC model [15,9] may fail to provide correct image segmentation. Thus PS model [8,9] was proposed to overcome this limitation. Instead of constant approximations c_1 and c_2 in PC model, two smooth functions u^+ and u^- were used to estimate the intensities outside and inside the contour C . However, this approach requires solving two elliptic PDEs for u^+ and u^- and one evolution equation for ϕ . The complexity of the algorithm limits its applications in practice.

2.2 Region-Scalable Fitting Energy Model

Recently, Li et al. [17] proposed a new region-based model to use the local intensity information in a scalable way. The energy functional they tried to minimize is:

$$\mathcal{E}(C, f_1(x), f_2(x)) = \sum_{i=1}^2 \lambda_i \int \int_{\Omega_i} K_\sigma(x - y) |I(y) - f_i(x)|^2 dy dx + \nu |C|, \tag{2}$$

where $\Omega_1 = outside(C)$ and $\Omega_2 = inside(C)$, λ_1, λ_2 and ν are positive constants, and $f_1(x)$ and $f_2(x)$ are two functions that approximate image intensities in Ω_1 and Ω_2 , respectively. The aim of the kernel function K_σ is to put heavier weights on points y which are close to the center point x . For simplicity, a Gaussian kernel with a scale parameter $\sigma > 0$ was used:

$$K_\sigma(u) = \frac{1}{2\pi\sigma^2} e^{-|u|^2/2\sigma^2}. \tag{3}$$

To handle topological changes, the authors in [17] converted (2) to a level set formulation.

As in level set methods [25], the contour $C \subset \Omega$ is represented by the zero level set of a level set function $\phi : \Omega \rightarrow \mathbb{R}$. Thus, the energy \mathcal{E} in (2) can be written as:

$$\mathcal{E}_\epsilon(\phi, f_1, f_2) = \int \mathcal{E}_\epsilon^x(\phi, f_1(x), f_2(x)) dx + \nu \int |\nabla H_\epsilon(\phi(x))| dx, \tag{4}$$

where

$$\mathcal{E}_\epsilon^x(\phi, f_1(x), f_2(x)) = \sum_{i=1}^2 \lambda_i \int K_\sigma(x-y) |I(y) - f_i(x)|^2 M_i^\epsilon(\phi(y)) dy \quad (5)$$

is the region-scalable fitting energy, $M_1^\epsilon(\phi) = H_\epsilon(\phi)$ and $M_2^\epsilon(\phi) = 1 - H_\epsilon(\phi)$. H_ϵ is a smooth function approximating the Heaviside function H which is defined by:

$$H_\epsilon(x) = \frac{1}{2} \left[1 + \frac{2}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \right]. \quad (6)$$

In order to preserve the regularity of the level set function ϕ , they used a level set regularization term [12]:

$$\mathcal{P}(\phi) = \int \frac{1}{2} (|\nabla\phi(x)| - 1)^2 dx. \quad (7)$$

Therefore, the energy functional they proposed to minimize is:

$$\mathcal{F}(\phi, f_1, f_2) = \mathcal{E}_\epsilon(\phi, f_1, f_2) + \mu \mathcal{P}(\phi), \quad (8)$$

where μ is a positive constant.

To minimize this energy functional, the standard gradient descent method is used. By calculus of variations, for a fixed level set function ϕ , the optimal functions $f_1(x)$, $f_2(x)$ that minimize $\mathcal{F}(\phi, f_1, f_2)$ are obtained by:

$$f_i(x) = \frac{K_\sigma(x) * [M_i^\epsilon(\phi(x))I(x)]}{K_\sigma(x) * M_i^\epsilon(\phi(x))}, \quad i = 1, 2. \quad (9)$$

For fixed f_1 and f_2 , the level set function ϕ that minimizes $\mathcal{F}(\phi, f_1, f_2)$ can be obtained by solving the following gradient flow equation:

$$\frac{\partial\phi}{\partial t} = -\delta_\epsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) + \nu \delta_\epsilon(\phi) \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \mu[\nabla^2\phi - \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)], \quad (10)$$

where δ_ϵ is the derivative of H_ϵ , and e_i ($i = 1$ or 2) is defined as:

$$e_i(x) = \int K_\sigma(y-x) |I(x) - f_i(y)|^2 dy, \quad i = 1, 2. \quad (11)$$

3 Split Bregman Method for Minimization of Region-Scalable Fitting Energy

In this section we introduce a new region-scalable model which incorporates the GCS method and the split Bregman technique. In fact, the energy functional (4) of the RSF model in section 2.2 is nonconvex, so the evolution can be easily trapped to a local minimum. We thus apply the GCS method to the RSF model to make the fitting energy convex. The split Bregman technique is used to minimize the energy functional in a more efficient way. The proposed new model thus

can improve the robustness and efficiency, while inheriting the desirable ability to deal with intensity inhomogeneity in image segmentation.

Considering the gradient flow equation (10), we first drop the last term which regularized the level set function to be close to a distance function:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)[(-\lambda_1 e_1 + \lambda_2 e_2) - \nu \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right)], \quad (12)$$

without loss of generality, we take $\nu = 1$. The Chan-Vese Model can be considered as a special case of (12), i.e. $K_\sigma(y - x) = 1_\Omega/|\Omega|$.

Following the idea in Chan et al. [23], the stationary solution of (12) coincides with the stationary solution of:

$$\frac{\partial \phi}{\partial t} = [(-\lambda_1 e_1 + \lambda_2 e_2) - \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right)]. \quad (13)$$

The simplified flow represents the gradient descent for minimizing the energy:

$$E(\phi) = |\nabla \phi|_1 + \langle \phi, \lambda_1 e_1 - \lambda_2 e_2 \rangle. \quad (14)$$

This energy does not have a unique global minimizer because it is homogeneous of degree one. By restricting the solution to lie in a finite interval, e.g. $a_0 \leq \phi \leq b_0$, the global minimum can be guaranteed, i.e.

$$\min_{a_0 \leq \phi \leq b_0} E(\phi) = \min_{a_0 \leq \phi \leq b_0} |\nabla \phi|_1 + \langle \phi, r \rangle, \quad (15)$$

where $r = \lambda_1 e_1 - \lambda_2 e_2$. Once the optimal ϕ is found, the segmented region can be found by thresholding the level set function for some $\alpha \in (a_0, b_0)$:

$$\Omega_1 = \{x : \phi(x) > \alpha\}. \quad (16)$$

As in [26], we modify the energy (15) to incorporate information from an edge detector. This is accomplished by using the weighted TV norm:

$$TV_g(\phi) = \int g |\nabla \phi| = |\nabla \phi|_g, \quad (17)$$

where g is the non-negative edge detector function. One common choice for the edge detector is:

$$g(\xi) = \frac{1}{1 + \beta |\xi|^2}, \quad (18)$$

where β is a parameter that determines the detail level of the segmentation. By replacing the standard TV norm $TV(\phi) = \int |\nabla \phi| = |\nabla \phi|_1$ in (15) with the weighted version (17), we make the model more likely to favor segmentation along curves where the edge detector function is minimal. Then the minimization problem becomes:

$$\min_{a_0 \leq \phi \leq b_0} E(\phi) = \min_{a_0 \leq \phi \leq b_0} |\nabla \phi|_g + \langle \phi, r \rangle. \quad (19)$$

To apply the split Bregman approach [22] to (19), we introduce the auxillary variable, $\vec{d} \leftarrow \nabla\phi$. To weakly enforce the resulting equality constraint, we add a quadratic penalty function which results in the following unconstrained problem:

$$(\phi^*, \vec{d}^*) = \arg \min_{a_0 \leq \phi \leq b_0} |\vec{d}|_g + \langle \phi, r \rangle + \frac{\lambda}{2} \|\vec{d} - \nabla\phi\|^2. \quad (20)$$

We then apply Bregman iteration to strictly enforce the constraint $\vec{d} = \nabla\phi$. The resulting optimization problem is:

$$(\phi^{k+1}, \vec{d}^{k+1}) = \arg \min_{a_0 \leq \phi \leq b_0} |\vec{d}|_g + \langle \phi, r \rangle + \frac{\lambda}{2} \|\vec{d} - \nabla\phi - \vec{b}^k\|^2. \quad (21)$$

$$\vec{b}^{k+1} = \vec{b}^k + \nabla\phi^{k+1} - \vec{d}^{k+1}. \quad (22)$$

For fixed \vec{d} , the Euler-Lagrange equation of optimization problem (21) with respect to ϕ is:

$$\Delta\phi = \frac{r}{\lambda} + \nabla \cdot (\vec{d} - \vec{b}), \text{ whenever } a_0 < \phi < b_0. \quad (23)$$

By using central discretization for Laplace operator and backward difference for divergence operator, the numerical scheme for (23) is:

$$\alpha_{i,j} = d_{i-1,j}^x - d_{i,j}^x + d_{i,j-1}^y - d_{i,j}^y - (b_{i-1,j}^x - b_{i,j}^x + b_{i,j-1}^y - b_{i,j}^y). \quad (24)$$

$$\beta_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j}\phi_{i,j-1} + \phi_{i,j+1} - \frac{r}{\lambda} + \alpha_{i,j}). \quad (25)$$

$$\phi_{i,j} = \max\{\min\{\beta_{i,j}, b_0\}, a_0\}. \quad (26)$$

For fixed ϕ , minimization of (21) with respect to \vec{d} gives:

$$\vec{d}^{k+1} = \mathit{shrink}_g(\vec{b}^k + \nabla\phi^{k+1}, \frac{1}{\lambda}) = \mathit{shrink}(\vec{b}^k + \nabla\phi^{k+1}, \frac{g}{\lambda}), \quad (27)$$

where

$$\mathit{shrink}(x, \gamma) = \frac{x}{|x|} \max(|x| - \gamma, 0). \quad (28)$$

4 Implementation and Experimental Results

4.1 Implementation

The split Bregman algorithm for the minimization problem (19) in section 3 can be summarized as follows:

- 1:** while $\|\phi^{k+1} - \phi^k\| > \epsilon$ do
- 2:** Define $r^k = \lambda_1 e_1^k - \lambda_2 e_2^k$
- 3:** $\phi^{k+1} = GS(r^k, \vec{d}^k, \vec{b}^k, \lambda)$

- 4: $\vec{d}^{k+1} = \text{shrink}_g(\vec{b}^k + \nabla\phi^{k+1}, \frac{1}{\lambda})$
- 5: $\vec{b}^{k+1} = \vec{b}^k + \nabla\phi^{k+1} - \vec{d}^{k+1}$
- 6: Find $\Omega_1^k = \{x : \phi^k(x) > \alpha\}$
- 7: Update e_1^k and e_2^k
- 8: **end while**

Here, we have used $GS(r^k, \vec{d}^k, \vec{b}^k, \lambda)$ to denote one sweep of the Gauss-Seidel formula (24)-(26).

In this paper, the level set function ϕ can be simply initialized as a binary step function which takes a constant value b_0 inside a region and another constant value a_0 outside. Then the thresholding value α is chosen as $\alpha = (a_0 + b_0)/2$ when one needs to find the segmented region $\Omega_1 = \{x : \phi(x) > \alpha\}$. We choose $a_0 = 0$, $b_0 = 1$ in most of the experiments shown in this paper, while for some images, we will choose different values to get better result or faster convergence. The details will be shown in next subsection 4.2. In our implementation, the functions f_1 and f_2 are updated at every time step according to (9) before the update of the level set function ϕ . As in [17], in order to compute the convolutions in (9) more efficiently, the kernel K_σ can be truncated as a $w \times w$ mask, here we also choose $w = 4\sigma + 1$. For most of our experiments, the scale parameter is chosen as $\sigma = 3.0$ unless specified.

4.2 Results

The proposed method has been tested with synthetic and real images from different modalities. Unless otherwise specified, we use the following parameters in this paper: $\sigma = 3.0$, $a_0 = 0$, $b_0 = 1$, $\epsilon = 1$, and $\lambda = 0.001$. We use $\beta = 100$ for all images except for the brain image in the last column of Fig. 2 we use $\beta = 10$, and for the noisy synthetic images in column 1 of Fig. 2 and row 3 of Fig. 3 we choose $\beta = 20$. We use $\lambda_1 = 1.1e - 5$, $\lambda_2 = 1e - 5$ for most images in this paper, for several other images we use different parameters λ_1 , λ_2 for better and faster results. In general, our method with a small scale σ can produce more accurate location of the object boundaries.

We first show the result for a synthetic inhomogeneous image in Fig. 1. Column 1 is the original image with the initial contour, column 2 is the result of our method, column 3 is the result of the split Bregman on PC model in [22]. From this figure, we can see the advantage of our proposed method for inhomogeneous image. Our method can segment the images correctly even though the image is very inhomogeneous, while the split Bregman on PC model cannot segment it correctly as shown in column 3.

Fig. 2 shows the results for one synthetic image, one X-ray image of vessel, and two real images of a T-shaped object and an MR image from left to right. All of them are typical images with intensity inhomogeneity. The top row are the original images with the initial contours, the bottom row are the results with the final contours. As shown in Fig. 2, our method successfully extracts the object boundaries for these challenging images. For the images in the first two columns, we choose $\lambda_1 = \lambda_2 = 1e - 5$. For the real brain image in the last column, we use

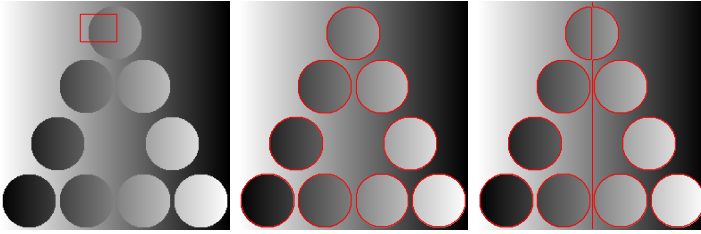


Fig. 1. Segmentation of a synthetic image with our proposed method and split Bregman on PC model. Column 1: the original image and the initial contour. Column 2: the result of our proposed method. Column 3: the result of the split Bregman on PC model.

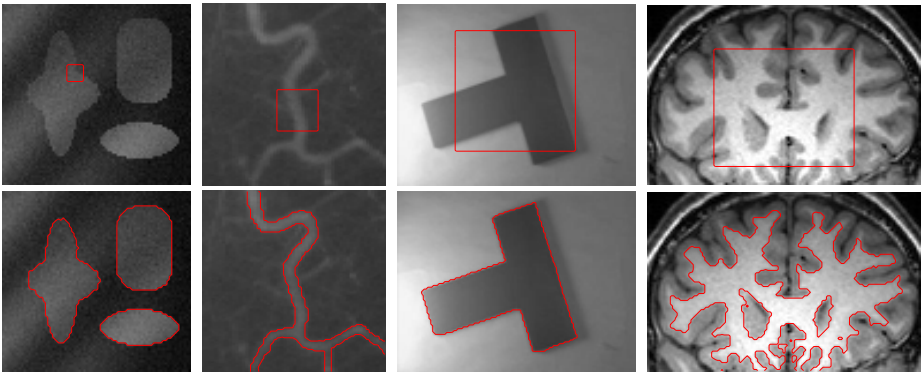


Fig. 2. Results of our method for synthetic images and real images. Top row: original images with initial contours. Bottom row: segmentation results with final contours.

$\lambda_1 = 1.25e - 5$ and $\lambda_2 = 1e - 5$ in order to put a larger penalty on the area of $inside(C)$. In this way the emergence of new contour outside the initial contour, which would increase the area of $inside(C)$, is to some extent prevented.

The results in Fig. 2 are similar to the results with the original RSF model in [17]. However, by comparing the computational procedures in the original RSF model and our model, it is clear that our method is more efficient than the RSF model because we apply the split Bregman approach to the optimization problem. This is demonstrated by comparing the iteration number and computation time in both methods for four images in Table 1, which were recorded from our experiments with Matlab code run on a Dell Precision 390 PC, Genuine Intel(R) Xeon (R), X7350, 2.93 GHz, 4 GB RAM, with Matlab 7.9. The sizes of these images are also shown in this table. In the experiments with the images in Fig. 2, the CPU time of our model is about one second.

We show the results for three synthetic flower images in Fig. 3. These images all have the same flower in the center but different distribution of intensities. The curve evolution process from the initial contour to the final contour is shown in every row for the corresponding image. The intensity of the image in the first

Table 1. Iteration number and CPU time (in second) for our model and RSF model for the images in Fig. 2 in the same order. The sizes of images are 75×79 , 110×110 , 96×127 , and 78×119 pixels, respectively.

	image1	image2	image3	image4
Our model	32(0.33)	67(1.13)	26(0.49)	48(0.70)
RSF model	200(1.40)	150(1.74)	300(3.72)	300(3.01)

row is piecewise constant. The second and third rows in Fig. 3 show the results for two images corrupted by intensity inhomogeneity. The image in third row was generated by adding random noise to the clean image in the second row. The standard deviation of the noise is 5.0. We can see that the segmentation results for the clean image and the noise contaminated version are very close. This demonstrates the robustness of our method to the noise.

Then we show the result for another synthetic image in Fig. 4. This image has been used in [1], there are three objects in this image with different intensities. The initial and the final contours are plotted on the images in the first row and the second row, respectively. The first column is the result for the piecewise constant image, the second column shows the result for the image corrupted by intensity inhomogeneity in the background. In this experiment, we choose $a_0 = -2$, $b_0 = 2$ instead of $a_0 = 0$, $b_0 = 1$, because when $a_0 = 0$, $b_0 = 1$ are chosen, the algorithm fails to detect the interior contour correctly as shown

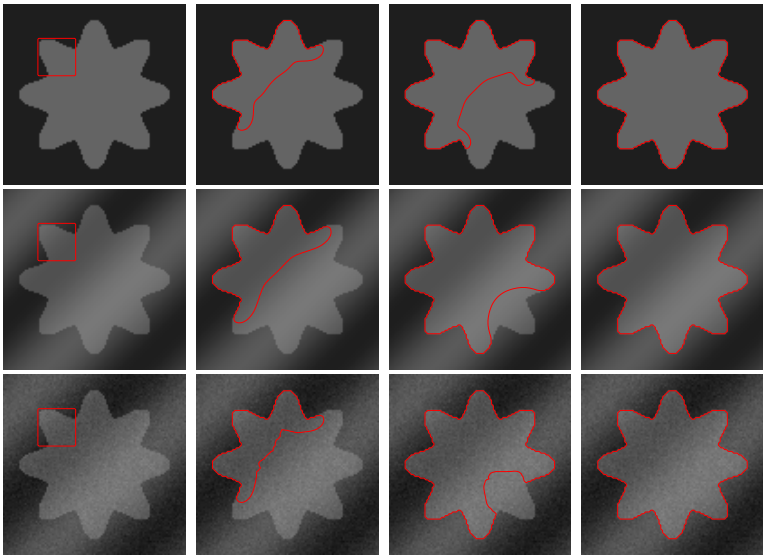


Fig. 3. Results of our method for three synthetic images. The curve evolution process from the initial contour (in the first column) to the final contour (in the fourth column) is shown in every row for the corresponding image.

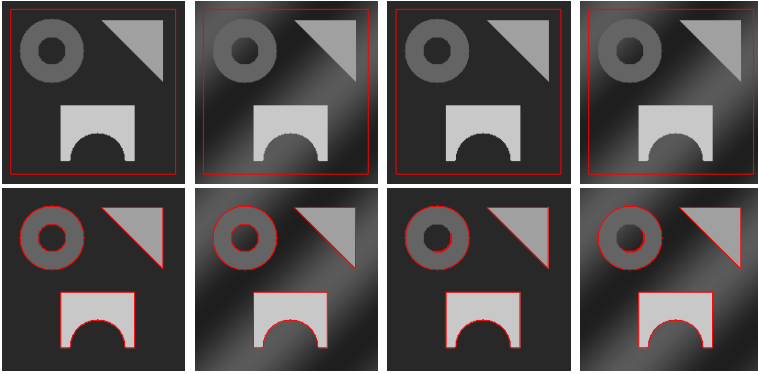


Fig. 4. Results of our method for a synthetic image. Row 1: the original image and the initial contour. Row 2: the final contour. Column 1: the piecewise constant image, $a_0 = -2$, $b_0 = 2$. Column 2: the inhomogeneous image, $a_0 = -2$, $b_0 = 2$. Column 3: the piecewise constant image, $a_0 = 0$, $b_0 = 1$. Column 4: the inhomogeneous image, $a_0 = 0$, $b_0 = 1$.



Fig. 5. The result of our method for a color image of flower. It shows the curve evolution process from the initial contour to the final contour.

in the third and the fourth columns. We choose $\lambda_1 = \lambda_2 = 2e - 6$ for the inhomogeneous images in the second and the fourth columns. From the second column, we can see that the intensity in the background is inhomogeneous and part of the background has very close intensities to the circular ring, but our method can successfully extract the object boundary in this image. The results also show that our method is able to segment images with multiple distinct means of image intensities.

In Fig. 5, we apply our model to a color image of flower. In this experiment, we choose $a_0 = -2$, $b_0 = 2$ and $\lambda_1 = \lambda_2 = 2e - 6$. The evolution of active contours from its initial state to the converged state is shown. This experiment shows that our method can also segment color images well.

5 Conclusion

This paper incorporates the GCS method and the split Bregman technique into the RSF model, which was originally formulated in a level set framework for

segmentation of inhomogeneous images. The proposed method significantly improves the efficiency and robustness of the RSF model, while inheriting its desirable ability to deal with intensity inhomogeneity in image segmentation. Furthermore, a non-negative edge detector function is used to detect the boundaries more easily. Our method has been applied to synthetic and real images with promising results. Comparisons with the split Bregman method on PC model and the original RSF model demonstrate desirable advantages of the proposed method.

References

1. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* 10, 266–277 (2001)
2. Cohen, L., Cohen, I.: Finite element methods for active contour models and balloons for 2d and 3d images. *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 1131–1147 (1991)
3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comput. Vis.* 1, 321–331 (1988)
4. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 158–175 (1995)
5. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vis.* 46, 223–247 (2002)
6. Ronfard, R.: Region-based strategies for active contour models. *Int. J. Comput. Vis.* 13, 229–251 (1994)
7. Samson, C., Blanc-Feraud, L., Aubert, G., Zerubia, J.: A variational model for image classification and restoration. *IEEE Trans. Patt. Anal. Mach. Intell.* 22, 460–472 (2000)
8. Tsai, A., Yezzi, A., Willsky, A.S.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. Image Process.* 10, 1169–1186 (2001)
9. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.* 50, 271–293 (2002)
10. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vis.* 22, 61–79 (1997)
11. Kimmel, R., Amir, A., Bruckstein, A.: Finding shortest paths on surfaces using level set propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 635–640 (1995)
12. Li, C., Xu, C., Gui, C., Fox, M.D.: Level set evolution without re-initialization: a new variational formulation. In: *Conference on Computer Vision and Pattern Recognition*, pp. 430–436. IEEE, Los Alamitos (2005)
13. Vasilevskiy, A., Siddiqi, K.: Flux maximizing geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 1565–1578 (2001)
14. Hou, Z.: A review on MR image intensity inhomogeneity correction. *Int. J. Biomed. Imag.* (2006)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* 42, 577–685 (1989)
16. Michailovich, O., Rathi, Y., Tannenbaum, A.: Image segmentation using active contours driven by the Bhattacharyya gradient flow. *IEEE Trans. Image Process.* 16, 2787–2801 (2007)

17. Li, C., Kao, C., Gore, J.C., Ding, Z.: Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Imag. Proc.* 17, 1940–1949 (2008)
18. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica. D.*, 259–268 (1992)
19. Goldstein, T., Osher, S.: The split Bregman method for L1 regularized problems. *UCLA CAM Report 08-29* (2008)
20. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. *Simul.* 4, 460–489 (2005)
21. Houhou, N., Thiran, J.P., Bresson, X.: Fast texture segmentation based on semi-local region descriptor and active contour. *Numer. Math. Theor. Meth. Appl.* 2, 445–468 (2009)
22. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split Bregman method: segmentation and surface reconstruction. *UCLA CAM Report 09-06* (2009)
23. Chan, T., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* 66, 1932–1948 (2006)
24. Li, C., Kao, C., Gore, J., Ding, Z.: Implicit active contours driven by local binary fitting energy. In: *Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, pp. 1–7. *IEEE Computer Society, Los Alamitos* (2007)
25. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* 79, 12–49 (1988)
26. Bresson, X., Esedoglu, S., Vanderghelynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imag. Vis.* 28, 151–167 (2007)

A Correlation-Based Approach for Real-Time Stereo Matching

Raj Kumar Gupta and Siu-Yeung Cho

Forensics and Security Laboratory, School of Computer Engineering
Nanyang Technological University, Singapore
{raj0005,assycho}@ntu.edu.sg

Abstract. In this paper, we present a new area-based stereo matching algorithm that computes dense disparity maps for a real time vision system. While many stereo matching algorithms have been proposed in recent years, correlation-based algorithms still have an edge due to speed and less memory requirements. The selection of appropriate shape and size of the matching window is a difficult problem for correlation-based algorithms. We use two correlation windows (one large and one small size) to improve the performance of the algorithm while maintaining its real-time suitability. Unlike other area-based stereo matching algorithms, our method works very well at disparity boundaries as well as in low textured image areas and computes a sharp disparity map. Evaluation on the benchmark Middlebury stereo dataset has been done to demonstrate the qualitative and quantitative performance of our algorithm.

1 Introduction

The estimation of depth from a pair of stereo images is one of the most challenging problems in the field of computer vision and is vital for many areas like robotics and virtual reality. An accurate disparity map can help robots navigate in the real environment while in virtual reality; disparity maps play an important role in 3D reconstruction from the image sets. A fast and accurate matching at object boundaries is necessary for proper rendering and reconstruction of an object in virtual environment. The 3D reconstruction problem can be viewed as stereo correspondence problem which includes finding a set of points in one image that can be identified as the same points in another image.

A large number of algorithms have been proposed to compute dense disparity map. A detailed overview on these stereo matching algorithms can be found in [1]. In general, the stereo matching algorithms can be divided into two categories: local methods and global methods. Local algorithms are statistical methods and are usually based on correlation. Global algorithms make explicit smoothness assumption and then solve it through various optimization techniques. Global algorithms are computationally very expensive which makes them impractical for real-time systems.

Local stereo matching algorithms can be subdivided into two broad categories: area-based algorithms and feature-based algorithms. Area-based algorithms use neighboring pixels of a given pixel to find the suitable match in another image by

using the intensity values of the pixels. These algorithms mostly use normalized cross-correlation (NCC) [2] or sum of absolute differences (SAD) [3, 4] or sum of squared differences (SSD) [5, 6, 7, 8] technique during the window matching process. The performance of area-based algorithms is highly influenced by the shape and size of the image region used during the matching process. Feature-based algorithms rely on feature extraction and match local cues (e.g. edge, corners). Through, these algorithms work very fast but they generate sparse disparity maps.

In this paper, a real-time correlation-based stereo matching method is presented that computes accurate disparity map from a stereo pair of images. The proposed algorithm uses two correlation windows (one large and one small size) to compute the disparity map. While large correlation window gives good results at non-textured image regions, the small window improves the performance at depth discontinuities. To demonstrate the performance of the proposed method, we have evaluated our algorithm using Middlebury datasets [9]. The rest of the paper organized as follows: In Section 2, we briefly cover the related literature; Section 3 describes the proposed algorithm. Section 4 contains the experimental results on the Middlebury dataset and a detailed comparison with other real-time stereo matching algorithm. The last section presents our conclusion and discusses the future work.

2 Related Work

The correlation-based methods find the corresponding match by measuring the similarity between two image areas. Most common correlation-based methods use Cross Correlation or the Sum of Squared or Absolute differences. The performance of these methods is strongly influenced by the size and shape of the matching image area. Usually, rectangular matching windows [3, 6, 7, 10, 11] are used to achieve high computational performance. The size of the matching window determines the number of pixels to be used for correlation. For the reliable disparity estimation, the matching window must be large enough to cover enough intensity variations, but small enough to cover only the pixels having the same disparity value. This requirement raises the need of different shape and size windows at different pixels within the same image as no fixed window size works well. While the large size window blurs the object boundaries, the small size window results are unreliable in low textured image regions. The pixels near to disparity discontinuity require windows of different shapes to avoid crossing the disparity.

Kanade and Okutomi [8] proposed an adaptive window-based method which starts with an initial estimation of the disparity map and updates it iteratively for each point by choosing the size and shape of a window till it converges. It uses the intensity and disparity variance to choose the window with the least uncertainty. This method is sensitive to the initial disparity estimations.

Boykov et al. [12] proposed a variable window algorithm, which is considerably fast and suitable for a real-time implementation. However, this method suffers from different systematic errors. Fusiello et al. [5] proposed a simple

multiple window approach. For each pixel and disparity, the correlation with a limited number of windows is performed and it retains the disparity with the best correlation value. Since it uses a limited number of windows, it cannot cover the complete windows range of required shapes and sizes.

Hirschmuller et al. [10] proposed a multiple window-based approach that uses different size windows, mainly focused on reducing the errors at depth discontinuities. The algorithm uses a border correction filter to improve matches at object borders. Many other multiple window approaches [7, 11, 13] have been proposed which use multiple windows of different sizes or use windows of different orientations to compute the matching cost.

Veksler [14] proposed an algorithm which chooses the appropriate window shape by optimizing over a large class of compact window by using the minimum ratio cycle algorithm. The window cost is computed as the average window error and bias to larger windows. While this method works very well, it is not efficient enough for real-time system. In [15], Veksler introduced another approach by computing the correlation value of several different window sizes for the pixel of interest and selects the window size with least matching error. However, this algorithm needs many user defined parameters for matching cost computation.

Yoon and Kweon [16] proposed a locally adaptive support weight approach which computes the support weights for each pixel in the support window based on their color dissimilarity and the spatial distance from the center pixel. These weights regulate the pixel's influence in the matching process. This approach gives very good results but it is computationally very expensive and also prone to image noise. The reported computation time of this algorithm on a fast machine is around one minute which makes it unsuitable for real-time systems.

3 Algorithm

In this section, we briefly describe the proposed algorithm. The proposed algorithm consists of the following four processing modules: (1) Initial matching; (2) Unreliable pixel detection; (3) Disparity interpolation; (4) Disparity refinement. In initial matching step, we compute the initial disparity map by using two different sizes of correlation windows. In unreliable pixel detection step, we use left-right consistency check to identify unreliable pixels. The left-right consistency check enforces the uniqueness constraint and identifies those pixels which have unreliable disparity. In disparity interpolation step, we estimate the disparity for unreliable pixels identified by left-right consistency check [6]. In final step, we refine the disparity map by using the reference image to improve the accuracy at depth discontinuities. We assume rectified image pair as an input i.e. the epipolar lines are aligned with the corresponding scanlines.

3.1 Initial Matching

To compute the initial disparity map, we use the sum of absolute difference (SAD) based matching approach by using the large correlation window. The

matching cost $C(x, y, d)$ of pixel (x, y) for disparity value d is given as follows:

$$C(x, y, d) = \sum_{i=-\omega/2}^{i=\omega/2} \sum_{j=-\omega/2}^{j=\omega/2} |I_l(x+i, y+j) - I_r(x+i, y+j-d)| + \xi, \quad (1)$$

where $I_l(x, y)$ and $I_r(x, y)$ are the intensities of the pixel in left and right image, respectively. ω represents the size of the matching window. The matching cost is computed for all possible disparity values and the disparity value with minimum matching cost is selected as shown in Fig. 1(a). However, in non-textured image regions, the matching window can have many such minima as shown in Fig. 1(b). It becomes very hard to determine the correct disparity values of the pixels that resides in such image regions. We use the disparity values of neighboring pixels to determine the disparity of such pixels. We propose a penalty term ξ based on gradient and the disparity values of neighboring pixels to estimate the disparity value in non-textured image regions accurately. The penalty term ξ is given as:

$$\xi = T(|d - d'|) \left(1 - \frac{|I_l(x, y) - I_l(x, y')|}{255} \right), \quad (2)$$

where d' is the disparity of the neighboring pixel (x, y') and T is the constant. The value of ξ changes according the change in gradient and becomes higher in low gradient image regions (e.g. low or non-textured areas) and lower in high gradient image regions (e.g. depth discontinuities). Fig. 1(c) demonstrates the matching costs computed for different disparity values in non-textured region of Tsukuba image at point (205, 230) without using ξ . We can see that there is ambiguity due to the number of local minima. This problem can be solved by using gradient and disparity values of the neighboring pixels. Fig. 1(d) demonstrates the matching cost computed by using equation (1). We can see that the disparity value can be easily determined now by using *winner-takes-all* approach. The matching costs C_l and C_r are computed for both left and right neighbors of a pixel respectively. The disparity of a pixel (x, y) can be computed as:

$$d_c(x, y) = \min \left(\arg \min_d C_l(x, y, d), \arg \min_d C_r(x, y, d) \right). \quad (3)$$

After computing the disparity values using the large correlation window, we use the small correlation window near the depth discontinuities. The use of large correlation window blurs the objects boundaries and the actual positions of these boundaries are usually within the distance of half the size of correlation window [10]. We compute the disparity values of all such pixels that reside near the depth discontinuities within the range of half the size of large correlation window. While computing the disparity of such pixels, we restrict the evaluation of the small correlation window only within those disparity values that are carried by neighboring pixels. The disparity of such pixels can be computed as:

$$d_c(x, y) = \arg \min_{d \in N} C(x, y, d), \quad (4)$$

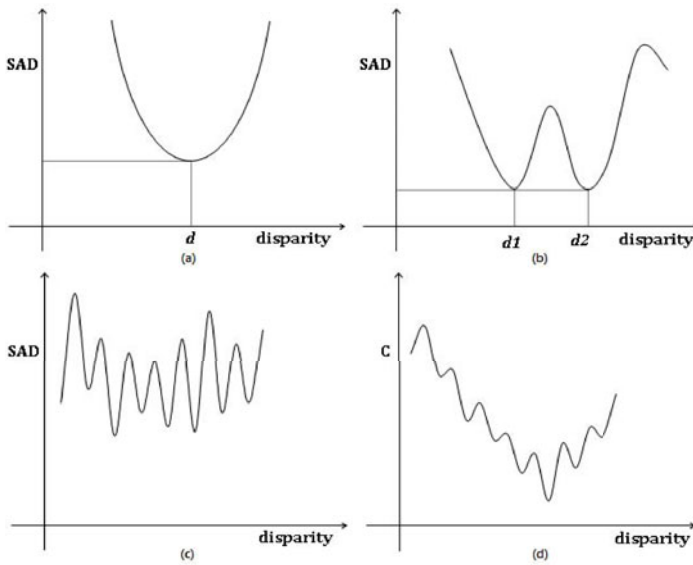


Fig. 1. Shows the problem in disparity selection. In (a), the disparity can be easily determined as d due to unique minimum value. It becomes ambiguous in case of multiple local minima as shown in (b). (c) shows the matching cost calculated at point (205, 230) of Tsukuba image. (d) displays the matching cost calculated by using the penalty term ξ for the same image point. The disparity can be uniquely determined in (d).

where $d_c(x, y)$ is the disparity of the pixel (x, y) and N represents the disparity values of the neighboring pixels. The cost $C(\cdot)$ is computed without using the penalty term as mentioned in Equation 1.

3.2 Unreliable Pixel Detection

The left-right consistency check is a very effective way to detect the unreliable pixels. The left-right consistency check is based on uniqueness constraint that assumes the one-to-one mapping between the stereo image points. We compute left and right initial disparity maps by using the method described in Section 3.1 by choosing the left and right image as reference image respectively. For each pixel of the left disparity map, we check whether it carries the same disparity value as its matching point in the right disparity map. A valid correspondence should match in both directions. A simple test of left-right cross checking can be written as:

$$|d_l(x, y) - d_r(x, y'')| < 1, \quad (5)$$

where (x, y) and (x, y'') are the correspondence pair of pixels and $d_l(x, y)$ and $d_r(x, y'')$ are left and right disparities for the points (x, y) and (x, y'') , respectively. All the pixels that fail to satisfy the Equation 5 are marked as unreliable pixels.

3.3 Disparity Interpolation

The left-right consistency check filters out occluded pixels as well as the unreliable pixels. In this step, we assign new disparity values to all such pixels with the help of their reliable neighboring pixels. For each unreliable pixel, we search for valid pixels in its 8 neighboring pixels. For all these valid pixels, we compute the distance between the intensities of unreliable pixel and its valid neighbor in the reference image. We assign the disparity value of the valid pixel which has the minimum distance from unreliable pixel in reference image.

3.4 Disparity Refinement

To remove the outliers for each color segment, plane fitting method is used widely. It is assumed that the neighboring pixels which have the same intensity values will also have the same disparity values. While this method increases the accuracy of the algorithm, it requires the color segmented image as an input which makes it computationally very expensive. Here, we use the disparity refinement approach that has been proposed in [17]. The algorithm uses the reference color information to refine the disparity map without performing any image segmentation.

4 Experiments

The Middlebury stereo benchmark dataset [9] has been used to evaluate the performance of the proposed algorithm. In our experiment, the small window

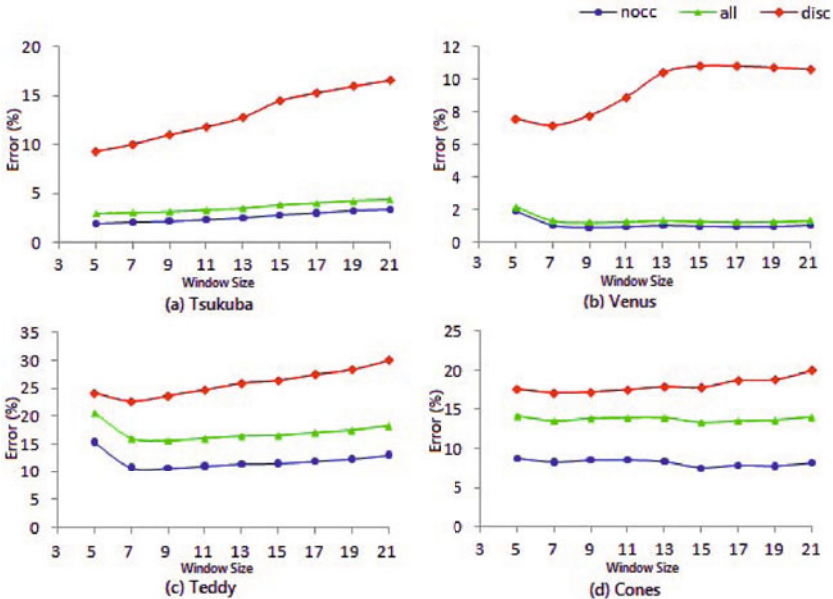


Fig. 2. Shows the percentage error in non-occluded (*nocc*), whole image (*all*) and near depth discontinuities (*disc*) for different window sizes for all four images

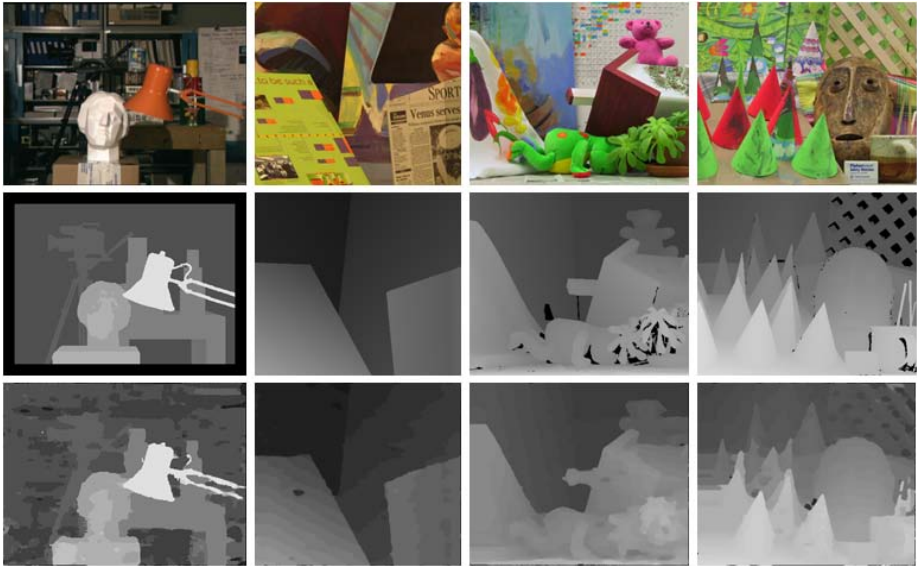


Fig. 3. Results on the Middlebury data set (Tsukuba, Venus, Cones and Teddy). The first row shows the left images, the second row shows the corresponding ground truth disparity maps and the third row shows the results obtained by using our algorithm.

size is chosen as 3×3 for all test images while the large window size is set to 9×9 for all images. Figure 3 shows the qualitative results of our approach for all four images. These image pairs along with their ground truth disparity map have been taken from the Middlebury database [9]. The performance of the proposed method for the Middlebury dataset is summarized in Table 1. The values shown in Table 1 represent the percentage of the bad pixels with an absolute disparity error greater than one for different regions: they are non-occluded (*nocc*), whole

Table 1. Comparison of the proposed method with other real-time algorithms listed on Middlebury evaluation table for absolute disparity error > 1 . The complete set of results can be found at <http://vision.middlebury.edu/stereo/eval/>

Algorithm	Tsukuba			Venus			Teddy			Cones			% Error
	<i>nocc</i>	<i>all</i>	<i>disc</i>	<i>nocc</i>	<i>all</i>	<i>disc</i>	<i>nocc</i>	<i>all</i>	<i>disc</i>	<i>nocc</i>	<i>all</i>	<i>disc</i>	
RTBFV [18]	1.71	2.22	6.74	0.55	0.87	2.88	9.90	15.0	19.5	6.66	12.3	13.4	7.65
RTABW [17]	1.26	1.67	6.83	0.33	0.65	3.56	10.7	18.3	23.3	4.81	12.6	10.7	7.90
<i>Our Results</i>	<i>2.25</i>	<i>3.08</i>	<i>11.6</i>	<i>0.92</i>	<i>1.31</i>	<i>7.53</i>	<i>10.7</i>	<i>15.7</i>	<i>23.6</i>	<i>8.25</i>	<i>13.5</i>	<i>16.6</i>	<i>9.59</i>
RTCensus [19]	5.08	6.25	19.2	1.58	2.42	14.2	7.96	13.8	20.3	4.10	9.54	12.2	9.73
RTGPU [20]	2.05	4.22	10.6	1.92	2.98	20.3	7.23	14.4	17.6	6.41	13.7	16.5	9.82
DCBGrid [21]	5.90	7.26	21.0	1.35	1.91	11.2	10.5	17.2	22.2	5.34	11.9	14.9	12.5
SSD+MF [1]	5.23	7.07	24.1	3.74	5.16	11.9	16.5	24.8	32.9	10.6	19.8	26.3	15.7

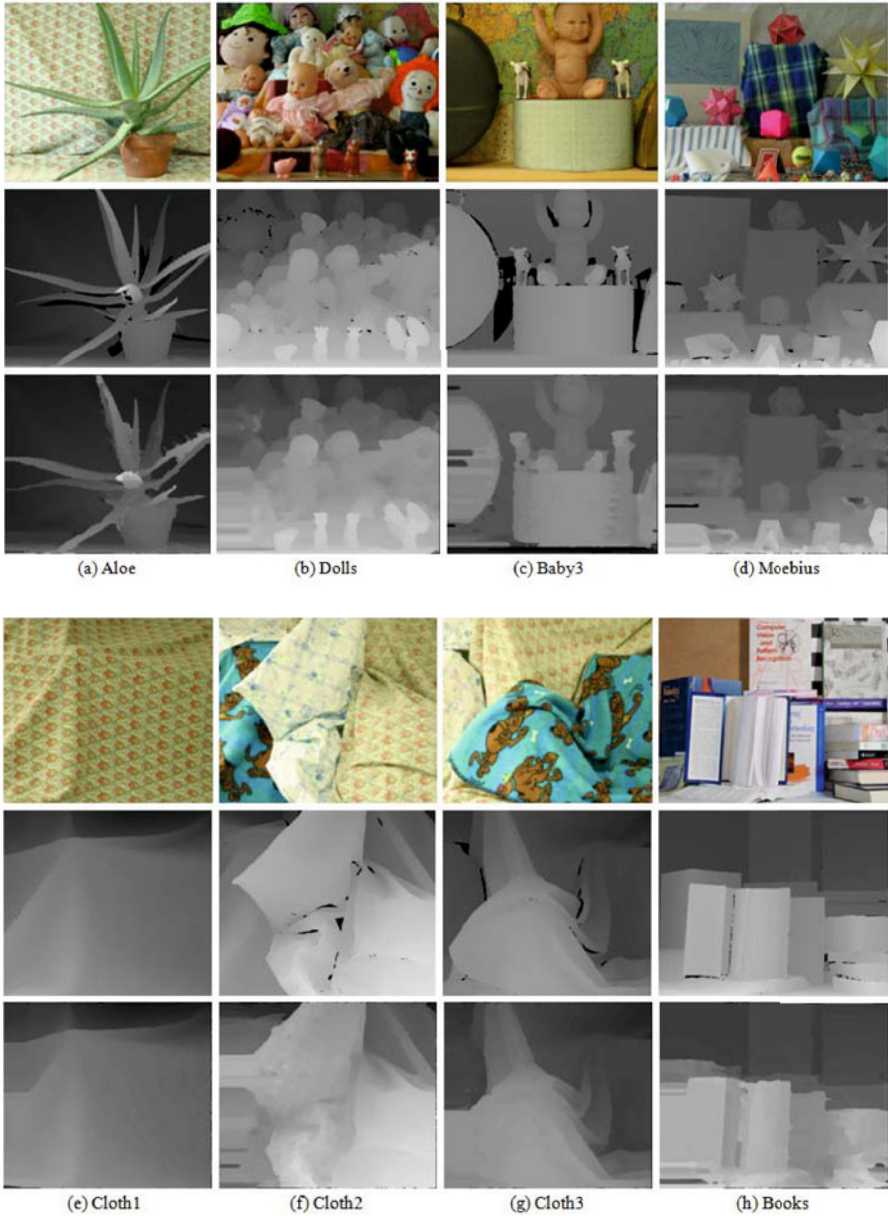


Fig. 4. Results on the new Middlebury dataset. The first row shows the left images, the second row shows the corresponding ground truth disparity maps and the third row shows the results obtained by using the proposed algorithm.

image (*all*) and pixels near discontinuities (*disc*). The last column of the table shows the overall performance of the algorithm for all four images.

Fig. 2 demonstrates the performance of the proposed algorithm with different window sizes. While small window size is fixed to 3×3 , we change the size of the first window (described as large window in proposed algorithm) used during initial matching operation. The window size has been changes from 5×5 to 21×21 . It shows the percentage error in non-occluded, whole image and near depth discontinuities for Tsukuba, Venus, Teddy and Cones images on using different window size. The error graphs show that the change in the size of matching window does not affect the performance of the algorithm significantly.

Fig. 4 shows the qualitative results for new Middlebury dataset images. These test images are taken from both Middlebury 2005 and 2006 datasets. These datasets consist of variety of image features i.e. complex geometry (dolls and Moebius), repetitive patterns (Aloe and Cloth1) and non-textured image regions (Books and Cloth2). The images in these datasets also have large disparity ranges, resulting in large occlusions. Due to large occlusions and higher percentage of untextured surfaces, the new Middlebury datasets are much more challenging as compared to standard stereo benchmark dataset which contains images such as Teddy and Cones. The experimental results clearly show that the proposed algorithm works very well in case of repetitive patters, object boundaries, as well as in occluded and non-textured image regions. These experimental results are obtained by taking same parameters (window size and constant value T) for all the images.

5 Conclusions and Future Work

In this paper, we present a new correlation-based stereo matching approach. The algorithm uses two correlation windows (one large and one small size) to compute the disparity map. While large correlation window gives good results at non-textured image regions, the small window improves the performance at depth discontinuities. The algorithm use simple mathematical operations and can be easily implemented on GPU. Although, the proposed method works very fast, the parallel implementation of the algorithm can reduce the computation time significantly. The computation time can also be reduced by using the sliding window approach at the time of correlation. In our future work, we plan to extend this work with all these investigation to improve the efficiency of the algorithm.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal on Computer Vision* 47, 7–42 (2002)
2. Faugeras, O., Hotz, B., Mathieu, M., Viville, T., Zhang, Z., Fua, P., Thron, E., Moll, L., Berry, G.: Real-time correlation-based stereo: algorithm, implementation and applications. INRIA Technical Report No. 2013 (1993)

3. Yoon, S., Park, S.K., Kang, S., Kwak, Y.: Fast correlation-based stereo matching with the reduction of systematic errors. *Pattern Recognition Letters* 26, 2221–2231 (2005)
4. Stefano, L., Marchionni, M., Mattoccia, S.: A fast area-based stereo matching algorithm. *Image and Vision Computing* 22, 983–1005 (2004)
5. Fusiello, A., Roberto, V., Trucco, E.: Efficient stereo with multiple windowing. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 858–863 (1997)
6. Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications* 6, 35–49 (1993)
7. Adhyapak, S., Kehtarnavaz, N., Nadin, M.: Stereo matching via selective multiple windows. *Journal of Electronic Imaging* 16 (2007)
8. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiments. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16, 920–932 (1994)
9. <http://vision.middlebury.edu/stereo>
10. Hirschmuller, H., Innocent, P., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. *Int. J. on Computer Vision* 47, 229–246 (2002)
11. Okutomi, M., Katayama, Y., Oka, S.: A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *Int. Journal on Computer Vision* 47 (2002)
12. Boykov, Y., Veksler, O., Zabih, R.: A variable window approach to early vision. *IEEE Trans. Pattern Analysis and Machine Intelligence* 20, 1283–1294 (1998)
13. Jeon, J., Kim, C., Ho, Y.S.: Sharp and dense disparity maps using multiple windows. In: Chen, Y.-C., Chang, L.-W., Hsu, C.-T. (eds.) *PCM 2002*. LNCS, vol. 2532, pp. 1057–1064. Springer, Heidelberg (2002)
14. Veksler, O.: Stereo matching by compact window via minimum ratio cycle. In: *IEEE Int. Conf. Computer Vision*, vol. 1, pp. 540–547 (2001)
15. Veksler, O.: Fast variable window for stereo correspondence using integral images. In: *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 556–561 (2003)
16. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28, 650–656 (2006)
17. Gupta, R., Cho, S.Y.: Real-time stereo matching using adaptive binary window. *3D Data Processing, Visualization and Transmission* (2010)
18. Zhang, K., Lu, J., Lafruit, G., Lauwereins, R., Gool, L.V.: Real-time accurate stereo with bitwise fast voting on cuda. In: *ICCVW* (2009)
19. Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: A fast stereo matching algorithm suitable for embedded real-time systems. In: *CVIU* (2010)
20. Gong, M., Yang, Y.: Near real-time reliable stereo matching using programmable graphics hardware. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2005)
21. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.: Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *Computer Vision – ECCV 2010*. LNCS, vol. 6311. Springer, Heidelberg (2010)

Photometric Stereo under Low Frequency Environment Illumination

Rui Huang and William A.P. Smith

University of York, UK
{rui,wsmith}@cs.york.ac.uk

Abstract. The well-studied problem of photometric stereo has almost exclusively made the assumption that illumination is provided by distant point light sources. In this paper, we consider for the first time the problem of photometric shape recovery from images in which an object is illuminated by environment lighting, i.e. where the illumination is modelled as a function over the incident sphere. To tackle this difficult problem, we restrict ourselves to low frequency illumination environments in which the lighting is known and can be well modelled using spherical harmonics. Under these conditions we show that shape recovery from one or more colour images requires only the solution of a system of linear equations. For the single image case we make use of the properties of spherical harmonics under rotations. We assume homogeneous Lambertian reflectance (with possibly unknown albedo) but discuss how the method could be extended to other reflectance models. We show that our method allows accurate shape recovery under complex illumination, even when our assumptions are breached, and that accuracy increases with the number of input images.

1 Introduction

Photometric methods for 3D surface recovery use information contained in the pixel brightness to infer object geometry. In contrast to geometric methods, this allows the information in every pixel to be utilised and potentially allows shape recovery from a single image (i.e. shape-from-shading). The idea is to invert a surface reflectance model, which is usually a function of the local surface normal, viewer and light source vectors. For this reason, photometric methods typically recover shape in the form of surface orientation estimates (i.e. surface normals). However, unlike geometric methods, strict assumptions about the illumination environment have meant that photometric methods have had little success on real world imagery.

The most restrictive assumption, and one which is made in almost all previous work, is that objects are illuminated by single point light sources. In this paper, we present what we believe is the first attempt to use photometric methods to recover shape from objects illuminated by complex, extended illumination. In order to tackle this problem, we must make a number of simplifying assumptions:

1. **Homogenous material properties** - reflectance properties are fixed over the whole surface and can be described by a parametric reflectance model with known parameters. We focus on Lambertian surfaces with unknown uniform albedo.
2. **Known low frequency illumination environment** - the illumination environment is known and we restrict ourselves to low frequency lighting which can be well approximated by spherical harmonics.
3. **No occlusion effects** - the hemisphere visible from each point on the surface is unoccluded by other parts of the surface. This assumption is equivalent to assuming the surface is convex. We present results for a real world non-convex surface in our experimental results.

Under these assumptions, we develop a robust and efficient method for photometric shape recovery from one or more colour images. Our method is based on spherical harmonic lighting and the properties of spherical harmonics. From multiple images, we are able to recover surface shape simply by solving a system of linear equations. From a single image, we use the properties of spherical harmonics under rotations to solve for the surface normal direction.

We present results on complex objects rendered under environment illumination and show that our method provides results of comparable accuracy to traditional photometric stereo for single source images. We also show single image results which are of comparable accuracy to shape-from-shading results for single source images.

1.1 Related Work

Photometric shape recovery has a long history in computer vision. The early work on shape-from-shading was done by Horn in the 1970s. The classical regularisation approach [1] used energy minimisation to estimate a field of surface normals which sought to satisfy image irradiance and surface smoothness constraints. Woodham [2] was the first to propose photometric stereo which uses images of a scene from a fixed viewpoint and varying illumination direction. Under assumptions of known light source directions, global convexity of the object and Lambertian reflectance, the surface normal and albedo can be recovered from as few as 4 images.

This work has been extended in a number of ways. One line of research has focused on uncalibrated photometric stereo, whereby the light source directions are unknown [3]. For a Lambertian surface, this leads to the bas-relief ambiguity, where the surface can only be recovered up to a linear transformation [4]. Some work has also considered more complex illumination environments than single point sources. Koenderink et al [5] allow an additional ambient component on top of a point source. Yuille et al [6] considered a similar scenario in which an object is illuminated by a single point source and a diffuse component which remains constant over all the images. In shape-from-shading, Langer and Zucker [7] have considered the case of completely ambient illumination provided by skylight. In this case, shading is a function of ambient occlusion and is hence related to the global geometry of the surface. Prados et al. [8] have recently reformulated

this problem as the solution of an integro-partial differential equation. Vogiatzis et al. [9] solve a nonlinear optimisation which includes estimation of the lighting environment within a multiview photometric stereo framework. This problem is geometrically constrained by the use of frontier points.

Other work on photometric methods has focussed on non-Lambertian surfaces. Georghiades et al. [10] showed how a parametric reflectance model could be incorporated into an uncalibrated photometric stereo framework. Hertzmann and Seitz [11] showed how the use of a gauge object (i.e. an object of known shape and the same material properties as the object under study) allow the recovery of geometry with unknown surface reflectance and illumination. Although their setup is restrictive, their approach is state-of-the-art in terms of the quality of the recovered shape. In shape-from-shading, Ragheb and Hancock [12] used a probabilistic model to fit a Lambertian plus specular model using geometric shape-from-shading.

Spherical harmonics have been used previously in photometric analysis. Basri et al. [13] use the projection of the Lambertian reflectance function onto a spherical basis to derive harmonic images. These can be used to approximate an image of an object under arbitrary illumination. This allows shape recovery upto a bas-relief ambiguity using an SVD of the observed image intensities. Recently, Kemelmacher and Basri [14] used spherical harmonic images in the context of facial images. A single reference face shape is used to estimate the lighting coefficients and albedo information first. Then the depth of the facial information is derived. Unlike previous work, the use of the spherical harmonic lighting in this paper is different. We use separate spherical harmonic projections for reflectance and illumination in a manner which is popular in the graphics literature [15]. This renders shape recovery as a linear problem.

The remainder of this paper is organised as follows: in the Section 2 we review spherical harmonic lighting, while in Section 3 we explain how spherical harmonic lighting can be used to recover the shape of an object. Experimental results are given in Section 4 and in Section 5 we provide some conclusions.

2 Spherical Harmonics Lighting

We begin by showing how to represent the illumination environment and surface reflectance in terms of a spherical harmonic basis. A spherical harmonic of degree l and order m , $Y_l^m(\theta, \phi) : |m| \leq l \in N$ is defined on the unit sphere as:

$$Y_l^m(\theta, \phi) = k_{l,m} P_l^m(\cos\theta) e^{im\phi}, \quad (1)$$

where $\theta \in [0, \pi]$, $\phi \in [0, 2\pi]$, $k_{l,m}$ is a constant and P_l^m is the associated Legendre polynomial. The spherical harmonics are orthogonal functions such that a spherical function can be expressed by a unique linear combination of spherical harmonic bases:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_{l,m} Y_l^m(\theta, \phi), \quad (2)$$

where the coefficients $c_{l,m}$ are derived by projecting the spherical function into the spherical harmonics basis:

$$c_{l,m} = \int_0^{2\pi} \int_0^\pi f(\theta, \phi) \overline{Y_l^m(\theta, \phi)} \sin(\theta) d\theta d\phi. \quad (3)$$

The quality of the approximation is determined by the degree of spherical harmonic used. Note that in practice, we limit the maximum degree, typically to a value of $l = 2$ or $l = 3$.

Spherical harmonics are an effective way to approximate a complex illumination environment. They have been used in computer graphics to speed up environment map rendering by replacing a numerical double integration with a matrix multiplication. Rendering using spherical harmonics is achieved by approximating the environment map function and bidirectional reflectance distribution function (BRDF), respectively. The rendered image is then simply given by the inner product of the two coefficient vectors.

Consider a Lambertian object with diffuse albedo $\rho \in [0, 1]$. If the local surface normal direction at point p is given by the vector \mathbf{N}_p , then the image intensity is described by an integral over the local incident hemisphere Ω_p :

$$I_p = \int_{\Omega_p} \rho E(\mathbf{L}) V_{p,\mathbf{L}} (\mathbf{L} \cdot \mathbf{N}_p) d\mathbf{L}, \quad (4)$$

where $E(\mathbf{L})$ is the illumination function (i.e. the incident radiance from direction \mathbf{L}) and $V_{p,\mathbf{L}}$ is the visibility function, defined to be zero if p is occluded in the direction \mathbf{L} and one otherwise. We make the assumption that the object is globally convex, i.e. that $\forall p, \mathbf{L} : V_{p,\mathbf{L}} = 1$ and hence the visibility function can be ignored. We test the effect of this simplification in our experimental results.

The illumination function is stored in a discretised environment map stored in a latitude-longitude map: $E(\theta, \phi)$ (see for example Figure 1). Such environment maps can be acquired by photographing a mirrored ball (a ‘light probe’) or using a fish-eye lens.

We can approximate (4) by decomposing the function into two spherical harmonic expansions: one for the environment map and one for the Lambertian reflectance function:

$$c_i = \sum_{\theta, \phi} E(\theta, \phi) Y_i(\theta, \phi) \quad (5)$$

$$d_i = \sum_{\theta, \phi} \max(0, \mathbf{N} \cdot [\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta]^T) Y_i(\theta, \phi) \quad (6)$$

The coefficients c_i describe a specific illumination environment, $E(\theta, \phi)$. The quality of this approximation is dependent on the frequency composition of the environment map and the number of coefficients used. The coefficients d_i approximate the Lambertian reflectance function for a particular normal direction \mathbf{N} .

An attractive feature of this representation is that the rendered image can be efficiently computed as the inner product of these two groups of coefficients:

$$I \approx \sum_i c_i \rho d_i, \quad (7)$$

or equivalently:

$$I \approx \rho \mathbf{c}^T \mathbf{d}. \quad (8)$$

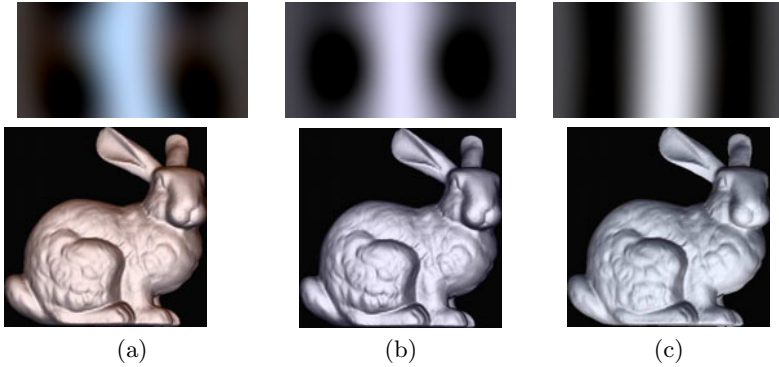


Fig. 1. Examples of rendering using spherical harmonic lighting. The first row show three examples of environment maps containing low frequency illumination. The second row shows an object rendered with the corresponding environment map.

Note that we work with colour images. In this case, we have a different set of coefficients for each colour channel of the environment map.

Figure 1 shows an example of using spherical harmonic lighting to render a Lambertian object under three different environment maps. These environment maps contain only low frequency illumination variations and are therefore well approximated by the spherical harmonics.

3 Shape Recovery

We now present our photometric shape recovery method based on spherical harmonic lighting. We present different methods for multiple images and single image recovery. In both cases, we assume that the environment map is known. In practice, this would correspond to capturing a light probe image in addition to each image of the object.

3.1 Shape Recovery from Multiple Images

For the case of multiple images, we derive a linear system of equations which is overdetermined. Consider the case of k colour images, giving $3k$ observations per pixel. The intensity at pixel p in the red channel for image i is denoted $I_{i,r}^p$, similarly for the green and blue channels. The vector of coefficients obtained by a spherical harmonic projection of the red channel of the i th environment map is denoted $\mathbf{c}_{i,r}$. Hence, for each pixel p we have the following linear system of equations:

$$\begin{aligned}
 I_{1,r}^p &= \mathbf{c}_{1,r}^T \rho \mathbf{d}^p \\
 I_{1,g}^p &= \mathbf{c}_{1,g}^T \rho \mathbf{d}^p \\
 I_{1,b}^p &= \mathbf{c}_{1,b}^T \rho \mathbf{d}^p \\
 &\vdots \\
 I_{k,b}^p &= \mathbf{c}_{k,b}^T \rho \mathbf{d}^p
 \end{aligned} \tag{9}$$

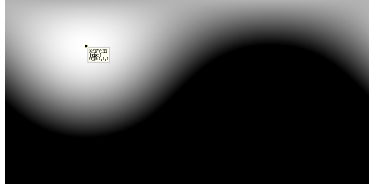


Fig. 2. The Cosine function of one specific surface normal. The surface normal is located at the point which has the highest intensity.

In other words, the observed intensities $\mathbf{b}^p = [I_{1,r}^p \ I_{1,r}^p \ I_{1,r}^p \ \dots \ I_{k,b}^p]^T \in \mathbb{R}^{3k}$ can be written in terms of the matrix of environment map coefficients: $\mathbf{A}^p \in \mathbb{R}^{3k \times n}$, where the first row of \mathbf{A}^p equals $\mathbf{c}_{1,r}$ etc, yielding the linear equation:

$$\mathbf{A}^p \rho \mathbf{d}^p = \mathbf{b}^p, \quad (10)$$

with \mathbf{d}^p being the unknown vector of coefficients which describes a Lambertian reflectance function for a point with normal direction \mathbf{N}_p . The number of unknowns, n , in this system of equations (i.e. the elements of \mathbf{d}^p) is determined by the number of bands, l , used in the spherical harmonic expansion of the environment map and reflectance function: $n = (l + 1)^2$. Hence, for $l = 2$ bands, the 9 observations given by 3 colour images are sufficient. In general, shape recovery using l bands requires at least $\lceil \frac{(l+1)^2}{3} \rceil$ colour images. We also require that $\text{rank}(\mathbf{A}^p) \geq n$. This requires that the colour channels of the environment map be linearly independent.

Note that the recovered vector of coefficients will be subject to a constant scaling, determined by the albedo. We therefore solve for $\rho \mathbf{d}^p$ for each pixel p using linear least squares. The remaining task is to recover the surface normal direction, \mathbf{N}_p , which corresponds to the Lambertian reflectance function described by \mathbf{d}^p . The most obvious way to accomplish this is to compute the spherical harmonic expansion of the coefficients \mathbf{d}^p using Equation 2 and find the maximum of the resulting function. The scaling by the albedo will apply a uniform scale factor but will not change the location of the maximum. However, such an approach is highly unstable as the exact location of the maximum is sensitive to noise. Instead, we optimise for the normal direction whose Lambertian spherical harmonic projection matches that of the recovered coefficients optimally in a least squares sense. This gives the recovered surface normal direction in global spherical coordinates. See Figure 2 for an example.

3.2 Shape Recovery from a Single Image

A similar form of shape recovery is still possible given only a single colour image. The restriction is that only $l = 1$ bands can be used. Although this reduces the accuracy of the recovered shape it has been shown previously [16] that, assuming all lighting directions are equally likely, the accuracy of first order spherical

harmonic approximations is at least 87.5%. Although recovery using a first order approximation would imply 4 unknowns per pixel with only 3 observations, we show that this problem can be made well-posed.

The first 4 terms of the spherical harmonic expansion of the Lambertian reflectance function with respect to Euclidian coordinates: $x = \sin \theta \cos \phi$, $y = \sin \theta \sin \phi$ and $z = \cos \theta$, are [?]:

$$\begin{aligned}
 Y_0^0 &= \sqrt{\frac{1}{4\pi}} \\
 Y_1^{-1} &= \sqrt{\frac{3}{4\pi}}x \\
 Y_1^0 &= \sqrt{\frac{3}{4\pi}}z \\
 Y_1^1 &= \sqrt{\frac{3}{4\pi}}y
 \end{aligned}
 \tag{11}$$

The first term is a constant and is unaffected by the surface normal direction. We follow the work of Ramamoorthi et al. [17], who show how to accelerate environment map rendering by performing a spherical harmonic projection of the environment map in a global coordinate system and of the Lambertian reflectance function under a local coordinate system. To perform rendering, a rotation matrix $D_{l,m}$ is used to convert the global coordinates to local ones. They prove that the rotation matrix can only change the coefficients within the same band. Hence, the rotation matrix for the first band is simply the identity matrix. The rotation matrix for the second band is given by:

$$D_2 = \begin{bmatrix} \cos \theta \sin \theta \sin \varphi & -\cos \varphi \sin \theta \\ 0 & \cos \varphi \sin \varphi \\ \sin \theta & -\cos \theta \sin \varphi \cos \theta \cos \varphi \end{bmatrix}
 \tag{12}$$

With the rotation matrix, the image is derived by

$$I = \sum_{l,m} c_{lm} D_l \mathbf{d}_{lm}
 \tag{13}$$

where c is the coefficient of the light source projected to the spherical harmonics, and d is the coefficient of the half cosine function projected to the spherical harmonics. $D_2 d_{2m} \in \mathbb{R}^3$ for the second band. In other words, the coefficients of the Lambertian reflectance function in a reference coordinate frame, \mathbf{d}_{lm} , can be related to the coefficient for a specific normal direction by rotation: $\mathbf{d}_r = D_l \mathbf{d}_{lm}$.

The rotation matrix for the first band is an identity matrix with rank 1. This means the first band will never change and can be eliminated while solving the linear equation. We still project the environment map under the global coordinates but the half-cosine function is represented under local coordinates. The linear equation we are solving is now:

$$\begin{aligned}
 I^1 - c_0^1 d_0 &= \mathbf{c}^1{}^T \rho \mathbf{d}_r \\
 I^2 - c_0^2 d_0 &= \mathbf{c}^2{}^T \rho \mathbf{d}_r \\
 I^3 - c_0^3 d_0 &= \mathbf{c}^3{}^T \rho \mathbf{d}_r
 \end{aligned}
 \tag{14}$$

where the albedo ρ must be known or assumed to be unity. The rotation matrix which relates the reflectance function under local and global coordinates can be recovered by

$$D_{2m} = \mathbf{d}_r \mathbf{d}_{lm}^{-1}.
 \tag{15}$$

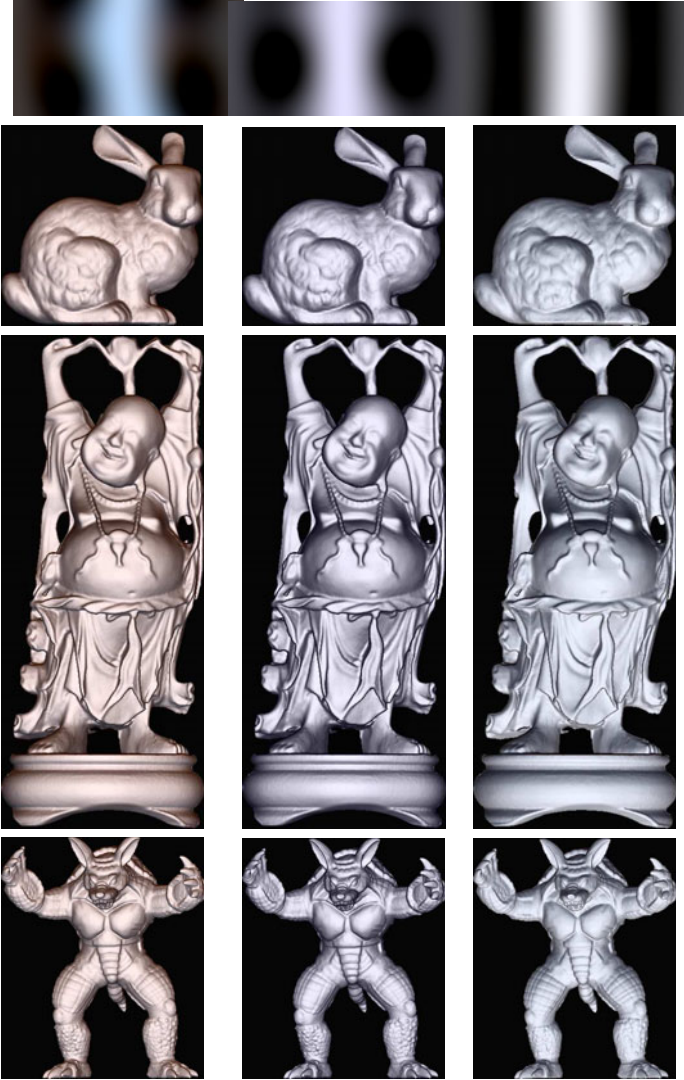


Fig. 3. The environment maps on the top with the corresponding input images

The components of the surface normal are embedded within the angular components of the rotation matrix. To recover surface normal direction, we simply solve for the spherical coordinates which are consistent with the recovered rotation matrix.

4 Experiments

We now present experimental results of our method. We begin by showing results for synthetic imagery rendered under contrived low frequency environment illumination. We select illumination environments which ensure our rank constraints are met. Figure 3 shows the three environment maps and three objects used in our experiment. The ground truth shape and the shape estimated by applying our method to three input images are shown in Figure 5. We integrate heightmaps from the estimated surface normals. The recovered surfaces retain much of the fine detail of the ground truth surfaces whilst also adhering to the global pattern of convexity and concavity.

In Figure 5 we show an example of surface recovery from one input image. Although there is clearly an increase in the global distortion, much of the fine surface detail is still correctly recovered.

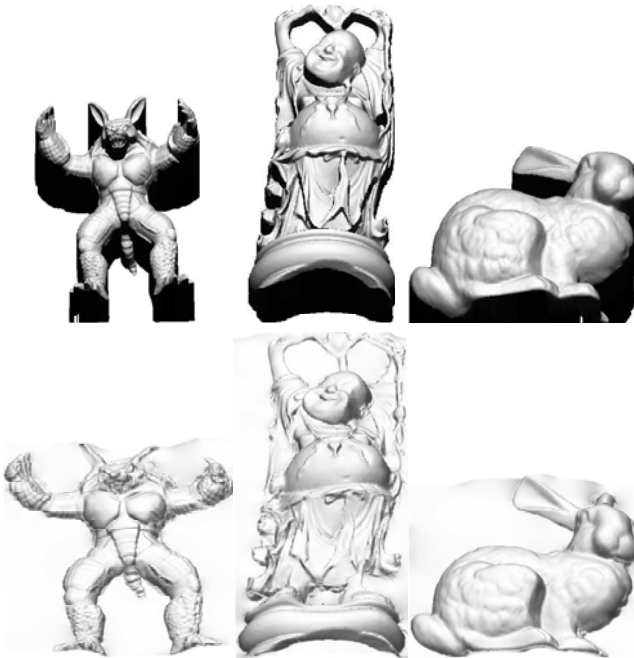


Fig. 4. Top row: ground truth surfaces, bottom row: estimated surfaces

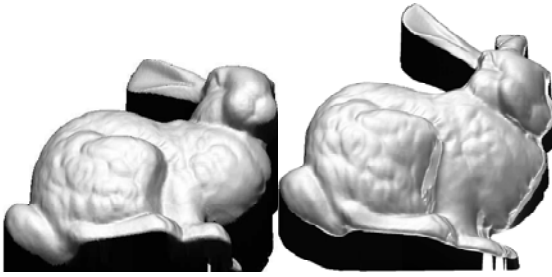


Fig. 5. Heightmap recovered from a single image (right) versus ground truth (left)

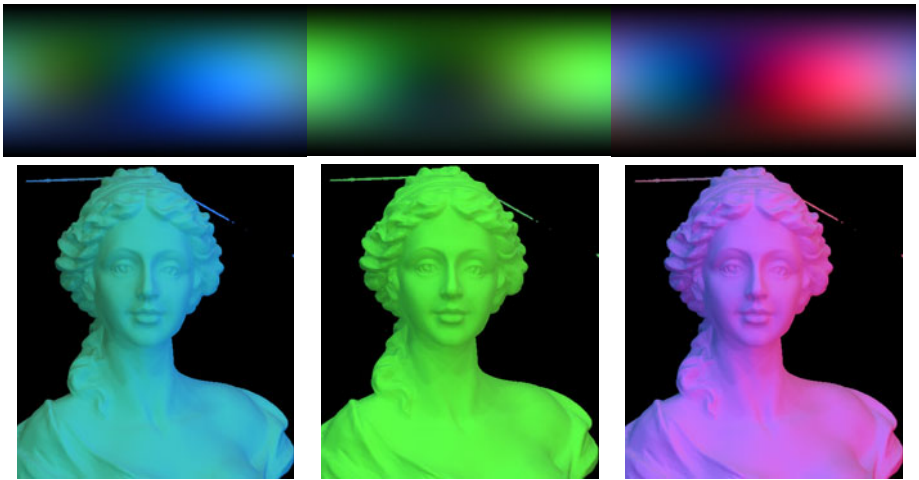


Fig. 6. Real world input images (bottom) imaged in a light stage approximating the low frequency illumination shown in the top row



Fig. 7. Estimated height maps using the method of [18] followed by the reconstructions from the images shown in Figure 6

Finally, we apply our method to some real world images. In order to approximate a low frequency illumination environment, we photograph a white statue with approximately Lambertian reflectance inside a ‘light stage’ [18]. The illumination environments we approximate can be seen in the top row of Figure 6. Note that since the light stage only contains 42 LED lights, in practice the environment is a discrete approximation to those shown. Photographs of the object in the light stage under the corresponding illumination conditions are shown in the bottom row of Figure 6. We perform single image shape recovery for each of the three images. The object contains many non-convex regions and hence portions of the incident hemisphere are occluded. Despite this, the reconstructions shown in Figure 7 still capture the global and local shape of the face well. We show the result of reconstructing the statue using the more restrictive spherical gradient method of Ma et al. [18] on the left of the Figure (this method requires 4 images under a specific lighting setup).

5 Conclusions

When an illumination environment can be accurately approximated by a spherical harmonic expansion, then photometric stereo under known environment illumination can be effected by simply solving a system of linear equations. Shape recovery is even possible from a single image, though fewer coefficients can be recovered resulting in a loss of accuracy in the recovered shape. The quality of the recovered shape depends heavily on how well the spherical harmonic coefficients can approximate the environment map.

As this is the first work to consider photometric shape reconstruction under complex environment illumination, there are many obvious limitations to our method. These provide several worthwhile avenues for future work. The first is to examine extending the method to non-Lambertian reflectance. Results in the graphics literature suggest that the spherical harmonic projection is still an efficient representation, though more coefficients are required to accurately represent highly specular reflectance. The second is to consider the effect of self occlusion, which may not only be a hindrance, but also potentially a cue to global geometry. Finally, we would like to explore whether any form of shape recovery is possible when the illumination environment is unknown.

References

1. Horn, B.K.P., Brooks, M.J.: The variational approach to shape from shading. *Comput. Vis. Graph. Image Process.* 33, 174–208 (1986)
2. Woodham, R.: Photometric method for determining surface orientation from multiple images. *Optical Engineerings* 19, 139–144 (1980)
3. Shashua, A.: On photometric issues in 3d visual recognition from a single 2d image. *International Journal of Computer Vision* 21, 99–122 (1997)
4. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The bas relief ambiguity. *Int. J. Comput. Vision* 35, 33–44 (1999)

5. Koenderink, J.J., Doorn, A.J.V.: The generic bilinear calibration-estimation problem. *International Journal of Computer Vision* 23, 1573–1605 (1997)
6. Yuille, W.L., Snow, D., Belhumeur, R.E., Determing, P.: generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability. *International Journal of Computer Vision* 35, 203–222 (1999)
7. Langer, M.S., Zucker, S.W.: Shape-from-shading on a cloudy day. *J. Opt. Soc. Am. A* (11), 467–478
8. Prados, E., Jindal, N., Stefano, S.S.: A non-local approach to shape from ambient shading. In: *Proc. 2nd International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 696–708 (2009)
9. Vogiatzis, G., Favaro, P., Cipolla, R.: Using frontier points to recover shape, reflectance and illumination. In: *The 10th IEEE International Conference on Computer Vision (ICCV 2005)*, vol. 1, pp. 228–235 (2005)
10. Georgiades, A.S.: Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In: *ICCV*, pp. 816–823 (2003)
11. Hertzmann, A., Seitz, S.: Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 27, 1254–1264 (2005)
12. Ragheb, H., Hancock, E.R.: A probabilistic framework for specular shape from shading. *Pattern Recognit.* 36, 407–427 (2003)
13. Basri, R., Jacobs, D.: Photometric stereo with general unknown lighting. *International Journal of Computer Vision* 72, 239–257 (2007)
14. Kemelmacher, I., Basri, R.: 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99, 1–14 (2010)
15. Green, R.: Spherical harmonic lighting: The gritty details. In: *Proceedings of the Game Developers Conference*
16. Frolova, D., Simakov, D., Bastri, R.: Accuracy of spherical harmonic approximations for images of lambertian objects under far and near lighting. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004. LNCS*, vol. 3021, pp. 574–587. Springer, Heidelberg (2004)
17. Ramamoorthi, R., Hanrahan, P.: On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A*, 2448–2459 (2001)
18. Ma, W.-C., et al.: Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In: *Proc. EGSR* (2007)

Simultaneous Vanishing Point Detection and Camera Calibration from Single Images

Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha

The Key Lab of Machine Perception (Ministry of Education), Peking University, Beijing
P.R. China

Abstract. For images taken in man-made scenes, vanishing points and focal length of camera play important roles in scene understanding. In this paper, we present a novel method to quickly, accurately and simultaneously estimate three orthogonal vanishing points (TOVPs) and focal length from single images. Our method is based on the following important observations: If we establish a polar coordinate system on the image plane whose origin is at the image center, angle coordinates of vanishing points can be robustly estimated by seeking peaks in a histogram. From the detected angle coordinates, altitudes of a triangle formed by TOVPs are determined. Novel constraints on both vanishing points and focal length could be obtained from the three altitudes. By using the constraints, radial coordinates of TOVPs and focal length can be estimated simultaneously. Our method decomposes a 2D Hough parameter space into two cascaded 1D Hough parameter spaces, which makes our method much faster and more robust than previous methods without losing accuracy. Enormous experiments on real images have been done to test feasibility and correctness of our method.

Keywords: vanishing point detection, calibration using vanishing points, perceptual grouping.

1 Introduction

Under a pinhole camera model, a set of parallel lines in 3D space are projected to a set of lines in the image which converge to a common point. This point of intersection, perhaps at infinity, is called the vanishing point. The understanding and interpretation of man-made scene can be significantly simplified by the detection of vanishing points. Its applications [2, 3, 4, 7, 8, 9] range from robotic navigation, camera calibration, 3D reconstruction, augmented reality, image understanding and etc. For instance, for images taken in man-made scenes, without any 3D geometric information in Euclidean space, the spatial layouts of the scenes are very difficult to understand. In this case, vanishing points corresponding to three orthogonal directions may provide important information. Therefore, the task of detecting the three mutually orthogonal directions of a man-made scene has considerable attraction.

1.1 Related Work

Previous vanishing point detection techniques can be roughly divided into two categories: The first category uses an accumulator cell to accumulate lines passing through the corresponding image point [1, 10, 13]. Peaks in the accumulator space represent the potential vanishing points. The second category does not require seeking the peaks in an accumulator space. Instead, some iterative algorithms, such as the Expectation-maximization algorithm, are used to group lines [9, 12].

Here we present some work related to the first category since our method can also be put into this category. Previous methods vary in the choice of accumulator space. For example, Barnard [1] suggested the unbounded image space can be mapped into the bounded surface of a Gaussian sphere. Tuytelaars et al. [13] mapped points into different bounded subspaces according to their coordinates. Rother [10] pointed out that these methods could not preserve the original distances between lines and points. In his method, the intersections of all pairs of non-collinear lines are considered as accumulator cells instead of a parameter space. But since these accumulator cells are difficult to index, searching for the maximal from the accumulator cells is slow.

A common problem among previous methods [1, 13] is that they do not consider constraints between vanishing points and focal length. For images of man-made scene, there are constraints between TOVPs and focal length. In general, coordinates of two orthogonal vanishing points are enough to calculate focal length using these constraints. If the third vanishing point is detected, it could be used to refine the result. Without considering the constraints in detection, sometimes the detected vanishing points may be incorrect and focal length cannot be calculated. In order to solve this problem, Van Den Heuvel [6] used the constraints as additional criterion in detection. But this method is based on the assumption that the camera is calibrated. Rother [10] also used the constraints in searching for the vanishing points. But due to the reason mentioned above, this approach requires more computational efforts.

In this paper, we present novel constraints developed from the previous constraints mentioned above. Our method firstly detects altitudes of the triangle formed by TOVPs. The triangle is called the TOVPs triangle in this paper. The novel constraints on both vanishing points and focal length are obtained from the three altitudes. Based on the constraints, the 2D accumulator space could be simplified into two cascaded 1D accumulator spaces. Vanishing point detection by our method is much faster than previous methods. Focal length and vanishing points could be estimated simultaneously and the previous constraints are still guaranteed.

2 Notations and Basic Principles

2.1 Pinhole Camera Model

Consider a point in 3D world $\mathbf{M} = [x, y, z, 1]^T$. $\mathbf{m} = [u, v, 1]^T$ is its image point. In the pinhole camera model, the two homogeneous coordinates satisfy:

$$\mu \mathbf{m} = \mathbf{K}[\mathbf{R} \ \mathbf{t}]\mathbf{M} \quad (1)$$

where μ is a scale factor. \mathbf{K} is the intrinsic matrix defined as:

$$\mathbf{K} = \begin{bmatrix} f & s & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where f is the focal length of camera. α is the aspect ratio. (u_0, v_0) represents the principle point of the camera. s is the skew parameter. $[\mathbf{R} \ \mathbf{t}]$ is the extrinsic matrix, determined by the position and orientation of the camera. Further information about pinhole camera model can be found in [5].

In this paper, we assume the skew parameter to be zero, the aspect ratio to be one, and the principal point to be centered. The only intrinsic parameter that we consider is the focal length f .

2.2 Relationship between TOVPs and Focal Length

Let x, y, z be an orthogonal system of coordinates associated with a viewing camera, such that the origin of the system coincides with the optical center and the z -axis with the optical axis. The image plane is defined by the equation $z = f$ where f is the focal length. Let (x_1, y_1) , (x_2, y_2) , (x_3, y_3) be coordinates of the TOVPs v_1, v_2, v_3 in an image of the man-made architectural environment. One important property of the TOVPs is that for triangle $v_1v_2v_3$, its orthocenter coincides with the principal point of the image, which is assumed to be the image center in this paper. Relationship between vanishing points and focal length can be presented as:

$$\begin{cases} x_1x_2 + y_1y_2 + f^2 = 0 \\ x_2x_3 + y_2y_3 + f^2 = 0 \\ x_3x_1 + y_3y_1 + f^2 = 0 \end{cases} \quad (3)$$

Detailed explanations for these properties could be found in [2].

3 Distribution of Intersections of All Pairs of Converging Lines

Due to errors in line detection [11], real intersections of converging lines corresponding to one vanishing point are distributed around the true vanishing point. We discovered that if these intersections are not too close to the image center, their distribution has very different variances in different directions. This distribution could be approximately interpreted as an elliptical Gaussian distribution. Its minor axis is very short with respect to major axis. Meanwhile, the included angle between the major axis and a line determined by the true vanishing point and the image center is very small. Figure 1a shows an image with line segments converging to TOVPs. Lines corresponding to different vanishing points are shown in different colors. We add noises to these lines so that their intersections do not coincide with the vanishing points. Blue crosses represent intersections of all pairs of detected lines. As shown in Figure 1a, distributions of the blue crosses are approximately elliptical Gaussian distributions around vanishing points respectively.

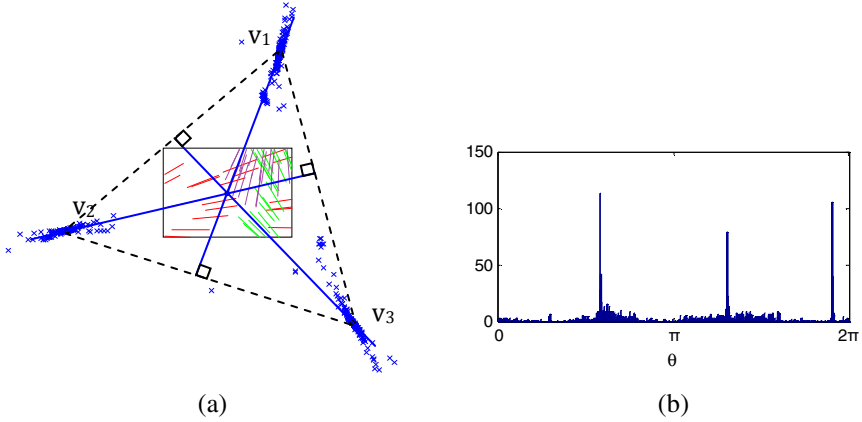


Fig. 1. (a) An image of converging lines with noise added. The TOVPs v_1, v_2, v_3 form a triangle called the TOVPs triangle. Lines corresponding to different vanishing points are shown in different colors. Blue crosses represent intersections of all pairs of detected lines. Orthocenter of the TOVPs triangle coincides with the image center. Altitudes of the TOVPs triangle are shown as blue solid lines. (b) Histogram of angle coordinates of intersections in (a).

Consider distribution of intersections of converging lines corresponding to one vanishing point in a polar coordinate system with origin at the image center. The major axis of the elliptical Gaussian distribution approximately passes through the origin. Angle coordinates of true vanishing points could be obtained by seeking peaks in histogram of angle coordinates of intersections. Figure 1b shows the histogram of angle coordinates of intersections of all pairs of lines in Figure 1a. Three significant peaks correspond to angle coordinates of the three vanishing points in Figure 1a.

As discussed in Section 2.2, orthocenter of the TOVPs triangle coincides with the image center, which is considered as origin of the polar coordinate system. Therefore, if angle coordinates of the TOVPs are detected, altitudes of the TOVPs triangle could be obtained. In our method, we detect altitudes of the triangle firstly. Then constraints from these altitudes are used to detect radial coordinates of the TOVPs and focal length simultaneously (described in Section 4). The detected altitudes of the triangle are shown as blue solid lines in Figure 1a.

4 Approach

4.1 Detecting Altitudes of the TOVPs Triangle

Let the image center be the origin of a Cartesian coordinate system in the image plane. The Cartesian coordinates of the three vanishing points v_1, v_2, v_3 are denoted by $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, respectively. The polar coordinates of v_1, v_2, v_3 is denoted by $(\theta_1, \rho_1), (\theta_2, \rho_2), (\theta_3, \rho_3)$, respectively. Polar transformation of the Cartesian coordinates (x, y) is defined as:

$$\begin{aligned} \theta &= \tan^{-1} \left(\frac{y}{x} \right) \\ \rho &= \sqrt{x^2 + y^2} \end{aligned} \quad (4)$$

According to Section 3, angle coordinates of the TOVPs could be detected by seeking peaks in θ -histogram of intersections of all pairs of detected lines. Consider an altitude of the TOVPs triangle $v_1v_2v_3$, which passes through vertex v_i . The altitude also passes through the image center, which is defined as origin in the polar coordinate system. Given angle coordinates $\theta_1, \theta_2, \theta_3$ of the three vanishing points, the altitudes are also determined. It can be represented as:

$$x \sin \theta_i - y \cos \theta_i = 0, i = 1, 2, 3 \tag{5}$$

In Figure 2a, intersections of all pairs of detected lines are represented by blue crosses. Figure 2b shows the θ -histogram of the intersections in Figure 2a. The three peaks correspond to the TOVPs. The detected altitudes are shown in Figure 2a.

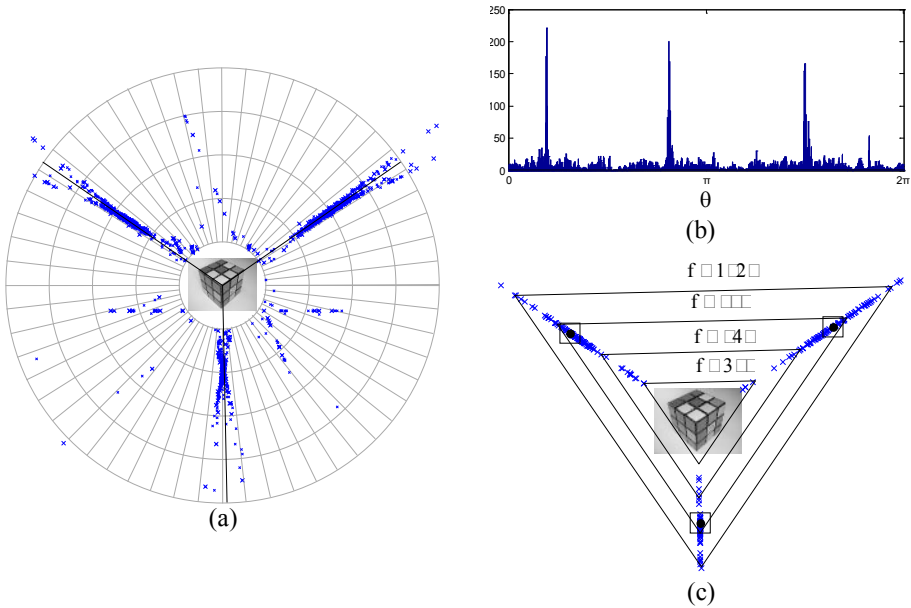


Fig. 2. (a) A polar coordinate system whose origin is at the image center. Intersections of all pairs of detected lines are marked by blue crosses. Detected altitudes of the TOVPs triangle are shown as solid black lines. (b) θ -histogram of the intersections. (c) A series of TOVPs triangles corresponding to different potential TOVPs and focal length. Detected TOVPs are marked by black squares.

4.2 Constraints from Three Altitudes of the TOVPs Triangle

Consider coordinates of the TOVPs $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and focal length f as seven unknowns. Since the TOVPs lie on the three detected altitudes respectively, we have the following equations from (5).

$$x_i \sin \theta_i - y_i \sin \theta_i = 0, i = 1, 2, 3 \tag{6}$$

Constraints provided by the detected altitudes of the TOVPs triangle $v_1v_2v_3$ could be represented as an equation system formed by (3) and (6). By solving the equation system, radial coordinates of the TOVPs could be denoted as:

$$\rho_i = \sqrt{x^2 + y^2} = \eta_i f, \quad (7)$$

where $i = 1, 2, 3$ and

$$\begin{aligned} \eta_1 &= \sqrt{-\frac{\cos(\theta_2 - \theta_3)}{\cos(\theta_1 - \theta_2) \cos(\theta_3 - \theta_1)}} \\ \eta_2 &= \sqrt{-\frac{\cos(\theta_3 - \theta_1)}{\cos(\theta_1 - \theta_2) \cos(\theta_2 - \theta_3)}} \\ \eta_3 &= \sqrt{-\frac{\cos(\theta_1 - \theta_2)}{\cos(\theta_2 - \theta_3) \cos(\theta_3 - \theta_1)}} \end{aligned} \quad (8)$$

Then we obtain:

$$\begin{aligned} x_i &= \rho_i \cos \theta_i \\ y_i &= \rho_i \sin \theta_i \end{aligned} \quad (9)$$

where $i = 1, 2, 3$. According to (7) and (9), for different f , the corresponding TOVPs triangles are similar triangles, as shown in Figure 2c.

4.3 Simultaneously Detecting Radial Coordinates of the TOVPs and Focal Length

The intersections of all pairs of detected lines can be divided into three sets according to the nearest altitudes. Since altitudes have already been detected in Section 4.1 as shown in Figure 2a, we project intersections on the corresponding altitudes and use the projections as candidates for the TOVPs. In Figure 2c, projections of intersections are marked by blue crosses. These projections are divided into three sets S_i , $i = 1, 2, 3$ according to corresponding altitudes. Consider the distribution of radial coordinates ρ of points in S_i . Define $g_i(\rho)$, $i = 1, 2, 3$ as the distribution function. Radial coordinate ρ_i of the i -th vanishing point should be the value that maximizes $g_i(\rho)$. Using (7), we can present this distribution function as a function of f :

$$g_i(\rho) = g_i(\eta_i f) \quad (10)$$

Focal length should be chosen to maximize three $g_i(\eta_i f)$, $i = 1, 2, 3$ simultaneously. We define $h(f)$ as a weighted sum of the three distributions:

$$h(f) = w_1 g_1(\eta_1 f) + w_2 g_2(\eta_2 f) + w_3 g_3(\eta_3 f) \quad (11)$$

where w_i , $i = 1, 2, 3$, is the weight of g_i . f should be a solution of:

$$f = \arg \max_f (h(f)) \quad (12)$$

Since f is measured by pixel and within a bounded range, this equation can be solved by simply enumerating all possible values of f . Given the estimation of f , coordinates of the TOVPs can be calculated by (7) and (9).

Computational complexity of our approach is determined by the following:

- (a) The number of intersections. In an image with n detected lines, the number of intersections could be $O(n^2)$,
- (b) The number of accumulator cells used to detect angle coordinates θ of the vanishing points, denoted by N_1 .
- (c) The number of accumulator cells used to detect focal length f and radial coordinates ρ of the vanishing points, denoted by N_2 .

Generally N_1 and N_2 are no more than 3000. The complexity of this approach can be $O(n^2 + N_1 + N_2)$.

5 Experiments

Many experiments have been done to test validity and correctness of our method. In our tests we use 640×480 images and the image center is considered as the origin of the image space. Accumulator space $[0, 2\pi]$ of θ is discretized into 600 accumulator cells. The range of f is set to $[500, 3200]$ and also discretized into accumulator cells. Weights used in (11) are all set to 1. Since flexibility of our method relies on the constraints provided by the TOVPs, images we use all contain three significant vanishing points.

In first experiment, images taken with different focal lengths are used to test our method. Figure 3.a-c show three images of same object with $f_a < f_b < f_c$. For images taken with longer focal length, detected lines are much closer to parallel. Many previous methods fail to obtain a reliable focal length when true focal length is relatively long because the vanishing points are close to infinity and difficult to estimate. In our method, novel constraints are used as additional criterion to reduce error. Detected coordinates of vanishing points and focal lengths of Figure 3a-c are shown in Table 1. The vanishing points are not marked in the figures because some of them are too far from image. Focal length errors in the three cases are all below 10%, compared with values calculated by method proposed in [14].

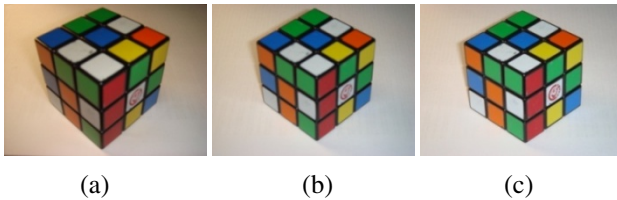


Fig. 3. Images taken with different focal lengths. We have $f_a < f_b < f_c$.

Table 1. Focal length and vanishing point coordinates of Figure 3a-c estimated by our method

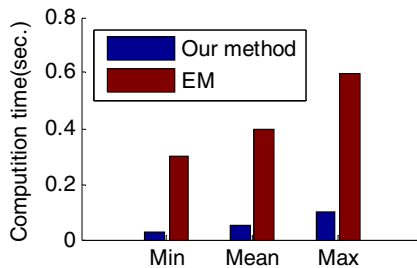
Image	f	TOVPs
Fig 3.a	713	(-949,-649), (1009,-693), (17,758)
Fig 3.b	1319	(1728,-1186), (-1767,1108), (34,1516)
Fig 3.c	1880	(2325,1463), (82,2546), (-2345,-1312)

Table 2. Focal length and vanishing point coordinates of Figure 4a-d estimated by our method

Image	f	TOVPs
Fig 4.a	713	(28,-1437), (897,443), (-883,408)
Fig 4.b	772	(1166,567), (750,489), (-47,-1146)
Fig 4.c	695	(-891,-366), (631,-217), (-169,1731)
Fig 4.d	721	(882,530), (-1033,-738), (-90,831)

In second experiment, we test our method by using images of different scenes. Figure 4a and 4b are taken by us. Figure 4c and 4d are from the ZuBuD¹ database. Intersections of detected lines in the images are marked by blue crosses. Detected vanishing points are marked by black squares. Detected coordinates of TOVPs and focal lengths are shown in Table 2.

We also compared our method with that using EM algorithm [9], which is considered as a quite efficient one among previous methods. Both methods are implemented in MATLAB M-files. They use same line detection algorithm and here we only compare the time cost by vanishing point detection. The results of vanishing point detection using the two methods are comparable. But our method costs much less time. Minimal, average and maximal computational time are reported in Figure 5. Computational time cost by our method in most cases is no more than 0.1 second. Our method can be used in real-time applications.

**Fig. 4.** Comparison of computation time

¹ <http://www.vision.ee.ethz.ch/showroom/zubud>

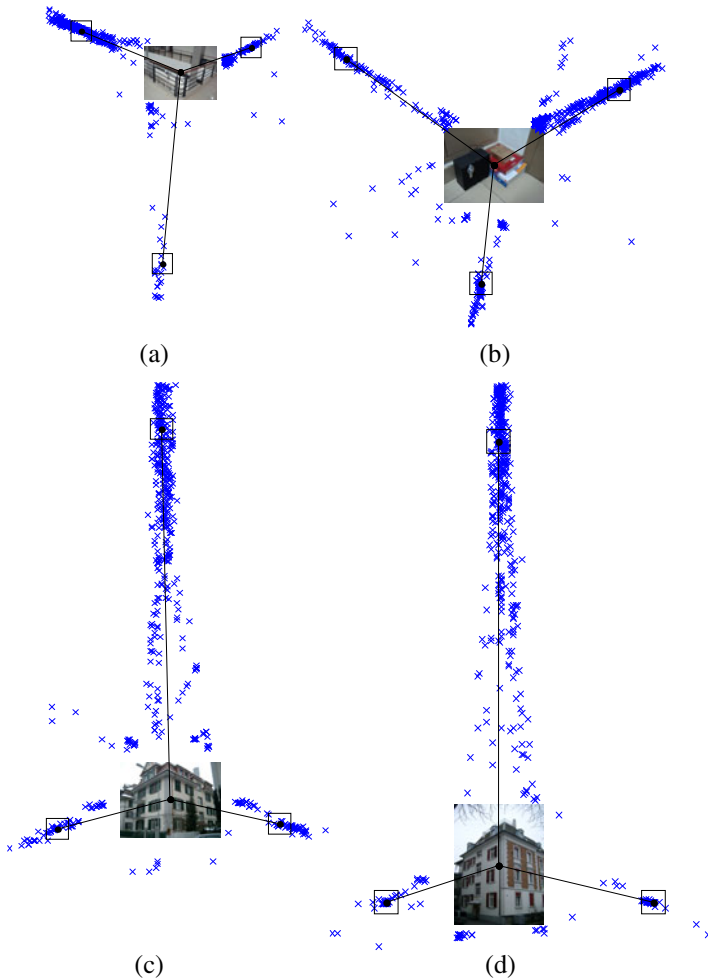


Fig. 4. Experiments on images of both indoor and outdoor scenes. (a) and (b) are taken by us. (c) and (d) are from the ZuBuD database. Intersections of detected lines in the images are marked by blue crosses. Detected vanishing points are marked by black squares.

5 Conclusions

A novel method for simultaneous vanishing point detection and camera calibration is proposed in this paper. The method can be described in two steps: Firstly, angle coordinates of the three vanishing points are detected. Secondly, focal length and radial coordinates of the vanishing points are estimated simultaneously.

This method is based on an observation that angle coordinates of vanishing points can be estimated easily and robustly. Altitudes of the TOVPs triangle may be determined from the detected angle coordinates of the TOVPs. The three altitudes provide constraints on both vanishing points and focal length, which largely simplifies the

estimation problem of vanishing points and focal length. Compared to previous methods, our method requires much less time and memory.

Acknowledgement

This work was supported in part by the NSFC Grant (No.61075034), the NHTRDP 863 Grant No. 2009AA01Z329, and the NHTRDP 863 Grant No.2009AA012105.

References

1. Barnard, S.T.: Interpreting perspective images. *Artificial Intelligence* 21, 435–462 (1983)
2. Caprile, B., Torre, V.: Using vanishing points for camera calibration. *International Journal of Computer Vision* 4, 127–140 (1990)
3. Cipolla, R., Drummond, T., Robertson, D.: Camera calibration from vanishing points in images of architectural scenes. In: *Proc. British Machine Vision Conference*, vol. 2, pp. 382–392 (1999)
4. Van Gool, L., Zeng, G., Van Den Borre, F., Müller, P.: Towards mass-produced building models. In: *Photogrammetric Image Analysis, PIA (2007)*
5. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge (2003)
6. Van Den Heuvel, F.A.: Vanishing point detection for architectural photogrammetry. *International Archives of Photogrammetry and Remote Sensing* 32(Part 5), 652–659 (1998)
7. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: *Proc. International Conference on Computer Vision (2009)*
8. Kong, H., Audibert, J., Ponce, J.: Vanishing point detection for road detection. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 96–103 (2009)
9. Košecká, J., Zhang, W.: Video compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002. LNCS*, vol. 2353, pp. 657–673. Springer, Heidelberg (2002)
10. Rother, C.: A new approach for vanishing point detection in architectural environments. *Image and Vision Computing* 20, 647–655 (2002)
11. Shufelt, J.A.: Performance evaluation and analysis of vanishing point detection Techniques. *IEEE Trans. Pattern Analysis and Machine Intelligence* 21(3), 282–288 (1999)
12. Tardif, J.: Non-iterative approach for fast and accurate vanishing point detection. In: *Proc. International Conference on Computer Vision (2009)*
13. Tuytelaars, T., Van Gool, L., Proesmans, M., Moons, T.: The cascaded Hough transform as an aid in aerial image interpretation. In: *Proc. International Conference on Computer Vision*, pp. 67–72 (1998)
14. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: *Proc. International Conference on Computer Vision (1999)*

Inferring Planar Patch Equations from Sparse View Stereo Images

Rimon Elias

Department of Digital Media Engineering and Technology
German University in Cairo
New Cairo City, Egypt
rimon.elias@guc.edu.eg

Abstract. This paper presents an approach to derive the equations of planar patches given a stereo pair of wide baseline or sparse view images. The algorithm starts by detecting junctions in both images using the so-called JUDOCA operator. A transformation matrix can be estimated using the junction attributes. In addition, a rough estimate for the fundamental matrix can be obtained to guide a matching process between the two views. Triangulation is used to reconstruct the locations of 2-edge junctions in space as well as to obtain the equations of 3D lines forming the junctions. This is done by obtaining four planes going through the projections of the junction edges and the optical centers. The equations of the 3D lines are used to get the equation of the plane of the junction.

1 Introduction

Stereo point matching and 3D reconstruction are basic problems in computer vision. Stereo point matching is concerned with finding corresponding points between a pair of images taken for the same scene. The difficulty of this problem varies according to how far the cameras are located with respect to each other and with respect to the viewed scene. If the cameras are close to each other in relationship to the scene (short baseline case), the point matching problem turns into an easy one. As cameras get far from each other (wide baseline or sparse view case) [1,2], more perspective deformation and more occlusion are present in the images taken which makes the problem more difficult. In this case, invariant measures should be utilized [3,4]. Examples of those invariant measures are affine [5] and homographic [6]. Another factor to be considered in this regard is whether the camera parameters are known or not (i.e., calibrated versus uncalibrated cases).

Another fundamental problem is the reconstruction of 3D models from multiple images. This topic has been tackled extensively in the literature. In general, techniques developed to reconstruct objects in 3D space can be divided into two main classes: “active methods” that use laser scanners, and “passive methods”

that use groups of images taken by cameras. There are many categories that exist under the title of passive methods; among them:

1. Reconstruction from contours and shape from silhouette.
2. Texture correlation and surface reconstruction
3. Feature based matching and visual fusion
4. Space carving/volumetric scene reconstruction.

Our paper can be classified under the second and third categories as we try to solve the problem of reconstructing planar patches in 3D space given a stereo pair of wide baseline images when camera parameters are known approximately.

The paper is organized as follows. Sec. 2 presents the steps of our algorithm. Sec. 3 presents some experimental results while Sec. 4 derives some conclusions.

2 Algorithm

Our algorithm consists of several steps as follows.

1. Detect junctions in both images using the so-called JUDOCA operator (Sec. 2.1). If n -edge junctions are detected, split them into a series of 2-edge junctions.
2. Determine a set of putative matches using the available rough epipolar geometry (Sec. 2.2).
3. Use the detected junctions to establish an invariant affine transformation matrix between each two putative matches (Sec. 2.3).
4. Use a correlation technique to pick up the best match (Sec. 2.4). Each match pair represents a planar patch in 3D space.
5. Use match pairs and information of the junctions forming it to estimate the equation of the planar patch in space (Sec. 2.5).

2.1 Detecting Junctions

Like a corner, a junction may be defined as a location where two or more edges meet (or where uniform regions join). However, a junction provides more information than a corner. This is concerning the orientations of edges forming the junction.

A good junction detector is the JUDOCA operator [7,8]. An illustration of this operator is shown in Fig. 1. The operation starts with determining the gradient magnitude through vertical and horizontal Gaussian filters according to some variance value. This is followed by creating two binary edge maps out of the gradient magnitude. These are called \mathbf{B} , which contains thick edges and $\mathbf{B}+$ where thin edges are included. The binary image \mathbf{B} is obtained by imposing a threshold on the gradient magnitude while the image $\mathbf{B}+$ is obtained by calculating the local maxima. As shown in Fig. 1, a circular mask whose radius is λ is placed at each point in \mathbf{B} and a list of points that belong to the image $\mathbf{B}+$ on the circumference is determined. Each of these points is called a circumferential

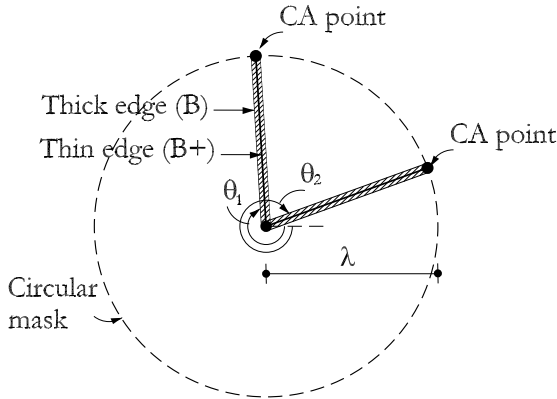


Fig. 1. An example of a JUDOCA 2-edge junction. The dashed circle of radius λ represents the circular mask. The center of this circle is the location of the junction. The thick edges in \mathbf{B} appear as the hatched regions and the thin edges of $\mathbf{B}+$ appear as solid lines.

anchor (CA) point. The radial lines in image \mathbf{B} joining the center and each CA point are scanned. A junction is detected if at least two radial lines exist (where the angle between them $\neq 180^\circ$).

In our algorithm, we use the JUDOCA operator to detect junctions in both images. Searching for the corresponding junctions in both images is guided by the epipolar relationships (Sec. 2.2) and the information obtained from the JUDOCA detector is used to estimate a 3×3 affine matrix (Sec. 2.3).

2.2 Epipolar Geometry

The epipolar constraint is defined using a 3×3 fundamental matrix. This matrix establishes the relationship between a point \mathbf{m} in one image and its corresponding epipolar line \mathbf{l}' in the other image.

$$\mathbf{l}' = \mathbf{Fm} \quad (1)$$

The corresponding point \mathbf{m}' must lie on the epipolar line \mathbf{l}' (i.e., $\mathbf{m}'^T \mathbf{l}' = 0$). There are many approaches to estimate the fundamental matrix \mathbf{F} [9]. If the camera parameters are known, \mathbf{F} can be estimated as [10]:

$$\mathbf{F} = [\mathbf{e}_2]_{\times} [\mathbf{C}'\mathbf{R}'] [\mathbf{C}\mathbf{R}]^{-1} \text{ such that } \mathbf{e}_2 = \mathbf{P}' \begin{bmatrix} \mathbf{T} \\ 1 \end{bmatrix} \quad (2)$$

where $[\mathbf{e}_2]_{\times}$ is a 3×3 skew-symmetric matrix representing the second epipole \mathbf{e}_2 where all epipolar lines intersect; \mathbf{C} and \mathbf{C}' are the 3×3 calibration matrices of both cameras; \mathbf{R} and \mathbf{R}' are the 3×3 rotation matrices of both cameras; \mathbf{P}' is the

3×4 projection matrix of the second camera and \mathbf{T} is the 3D translation vector of the first camera.

Once the fundamental matrix is calculated using the approximate camera parameters, an epipolar line in one image can be obtained for each point in the other image. In case of inaccuracy or errors, the corresponding point is likely to lie around the epipolar line (i.e., $\mathbf{m}^T \mathbf{I}' \neq 0$). In this case, a strip surrounding the epipolar line is considered as a search range for the corresponding point.

Unfortunately, if two corresponding points are correlated directly in case of wide baseline or sparse view stereo images, the correlation may lead to incorrect results. Hence, a transformation step must be applied prior to applying correlation. Sec. 2.3 derives an affine matrix through junction information to perform the required transformation.

2.3 Affine Transformation

In this section, we will derive the affine transformation matrix that is used for point correlation afterwards. Consider the 2-edge junction shown in Fig. 2(a) and located at $\dot{\mathbf{p}} = [x, y]^T$. The inclination angles of its two edges are θ_1 and θ_2 . Hence, the internal angle of the junction can be expressed as $\theta_2 - \theta_1$. Notice that the circular mask of the JUDOCA operator whose radius is λ is enclosing a rectangle whose diagonals are indicated by the inclination angles θ_1 and θ_2 . It is obvious that the lengths of its width and height are represented as $2\lambda \cos\left(\frac{\theta_2 - \theta_1}{2}\right)$ and $2\lambda \sin\left(\frac{\theta_2 - \theta_1}{2}\right)$. Similarly, we can define the internal angle of

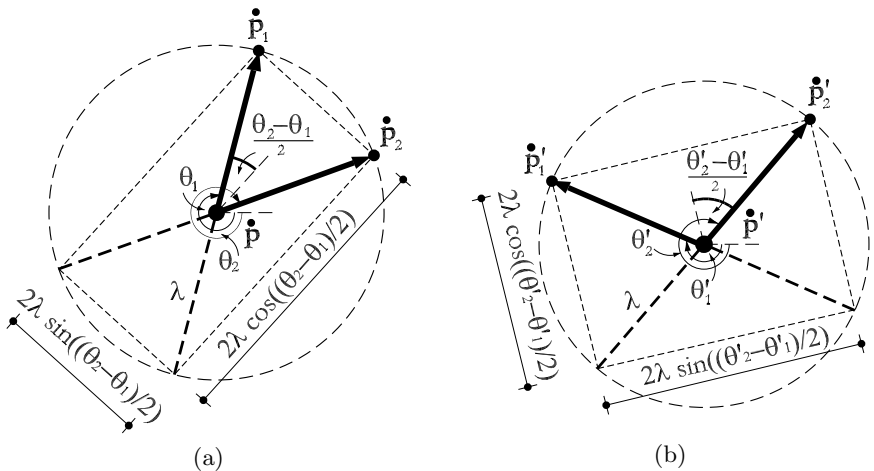


Fig. 2. Corresponding 2-edge junctions in two different images. (a) The inclination angles in the first image are θ_1 and θ_2 . (b) The orientation angles in the second image are θ'_1 and θ'_2 .

the corresponding junction located at $\dot{\mathbf{p}}' = [x', y']^T$ and shown in Fig. 2(b) as $\theta'_2 - \theta'_1$. Also, the rectangle in this case is bounded by borders of lengths $2\lambda \cos\left(\frac{\theta'_2 - \theta'_1}{2}\right)$ and $2\lambda \sin\left(\frac{\theta'_2 - \theta'_1}{2}\right)$.

In order to transform from triangle $\dot{\mathbf{p}}\dot{\mathbf{p}}_1\dot{\mathbf{p}}_2$ in Fig. 2(a) to its corresponding one $\dot{\mathbf{p}}'\dot{\mathbf{p}}'_1\dot{\mathbf{p}}'_2$ in Fig. 2(b), a series of translation, rotation, scaling, rotation and then translation must be performed. The translation operations can be expressed in homogeneous coordinates as:

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{T}_2 = \begin{bmatrix} 1 & 0 & x' \\ 0 & 1 & y' \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The rotation operations can be expressed as:

$$\mathbf{R}_1 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{R}_2 = \begin{bmatrix} \cos(\theta') & -\sin(\theta') & 0 \\ \sin(\theta') & \cos(\theta') & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

and the scaling operation can be expressed as:

$$\mathbf{S} = \begin{bmatrix} \frac{\cos\left(\frac{\theta'_2 - \theta'_1}{2}\right)}{\cos\left(\frac{\theta_2 - \theta_1}{2}\right)} & 0 & 0 \\ 0 & \frac{\sin\left(\frac{\theta'_2 - \theta'_1}{2}\right)}{\sin\left(\frac{\theta_2 - \theta_1}{2}\right)} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1 + \cos(\theta'_2 - \theta'_1)}{1 + \cos(\theta_2 - \theta_1)}} & 0 & 0 \\ 0 & \sqrt{\frac{1 - \cos(\theta'_2 - \theta'_1)}{1 - \cos(\theta_2 - \theta_1)}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Finally, the overall affine matrix is expressed as:

$$\mathbf{A} = \mathbf{T}_2 \mathbf{R}_2 \mathbf{S} \mathbf{R}_1 \mathbf{T}_1 \quad (6)$$

The resulting affine matrix \mathbf{A} is used in the following section before applying correlation in order to pick up the best match for a given junction.

2.4 Point Correlation

Many correlation techniques exist in the literature to test how near (or far) two points from being corresponding to each other. Among those methods is the sum of absolute differences (SAD) correlation, which can be used to correlate the $N \times N$ neighborhoods surrounding the pixels $\dot{\mathbf{p}} = [x, y]^T$ in the left image and $\dot{\mathbf{p}}' = [i, j]^T$ in the right image as [11]:

$$C_{SAD}(\dot{\mathbf{p}}; \dot{\mathbf{p}}') = \sum_{m, n = -\frac{N}{2}}^{\frac{N}{2}} |\mathbf{I}_l(x + m, y + n) - \mathbf{I}_r(i + m, j + n)| \quad (7)$$

where $\mathbf{I}(x, y)$ represents the intensity at $[x, y]^T$ and $N + 1$ is the side length of the correlation window in pixels. Applying affine transformation before correlation can be expressed as:

$$C_{SAD}(\mathbf{p}; \mathbf{p}') = \sum_{m, n = -\frac{N}{2}}^{\frac{N}{2}} |\mathbf{I}_l(x + m, y + n) - \mathbf{I}_r(\mathbf{p}')| \text{ such that } \mathbf{p}' = \mathbf{A} \begin{bmatrix} x + m \\ y + n \\ 1 \end{bmatrix} \quad (8)$$

Other types of correlation techniques are also available including sum of squared differences correlation (SSD), average of squared differences correlation (ASD) and variance normalized correlation (VNC).

The result of this correlation step is a set of matches. This set along with the camera parameters are used in the next step to estimate the equations of the planar patches where the junctions reside in 3D space.

2.5 Patch Equation

Any corresponding two edges \mathbf{I}_1 and \mathbf{I}'_1 of a junction can be used to reconstruct their actual 3D line L_1 in space (Fig. 3). This is done by intersecting two planes Π_1 and Π'_1 going through the optical centers \mathbf{C} and \mathbf{C}' and the lines in images (i.e., the projections of the 3D line). The planes are expressed as:

$$\begin{aligned} \Pi_1 &= \mathbf{P}^T \mathbf{I}_1 = \mathbf{P} \begin{bmatrix} \tan(\theta_1) \\ -1 \\ y - \tan(\theta_1)x \end{bmatrix} \\ \Pi'_1 &= \mathbf{P}'^T \mathbf{I}'_1 = \mathbf{P}'^T \begin{bmatrix} \tan(\theta'_1) \\ -1 \\ y' - \tan(\theta'_1)x' \end{bmatrix} \end{aligned} \quad (9)$$

where \mathbf{P} and \mathbf{P}' are the projection matrices; θ_1 and θ'_1 are the inclination angles of \mathbf{I}_1 and \mathbf{I}'_1 respectively; and $[x, y]^T$ and $[x', y']^T$ are the locations of the corresponding junctions. Thus, the equations of the 3D lines L_1 and L_2 representing the edges of the junction are expressed as:

$$\begin{aligned} L_1 &= \begin{bmatrix} \mathbf{I}_1^T \mathbf{P} \\ \mathbf{I}'_1{}^T \mathbf{P}' \end{bmatrix} = \begin{bmatrix} \mathbf{N}_1^T | d_1 \\ \mathbf{N}'_1{}^T | d'_1 \end{bmatrix} \\ L_2 &= \begin{bmatrix} \mathbf{I}_2^T \mathbf{P} \\ \mathbf{I}'_2{}^T \mathbf{P}' \end{bmatrix} = \begin{bmatrix} \mathbf{N}_2^T | d_2 \\ \mathbf{N}'_2{}^T | d'_2 \end{bmatrix} \end{aligned} \quad (10)$$

where $[\mathbf{N}_1^T | d_1]^T$ is the equation of the plane Π_1 going through the first optical center and the projection \mathbf{I}_1 such that \mathbf{N}_1 is the normal to this plane and d_1 is the perpendicular distance to this plane from the world coordinate system origin. The same notation is applied for $\Pi'_1 = [\mathbf{N}'_1{}^T | d'_1]^T$, $\Pi_2 = [\mathbf{N}_2^T | d_2]^T$, and $\Pi'_2 = [\mathbf{N}'_2{}^T | d'_2]^T$. The plane of the junction in space has a normal that can be estimated through the normal vectors to the previous four planes. This is expressed as:

$$\mathbf{N} = (\mathbf{N}_1 \times \mathbf{N}'_1) \times (\mathbf{N}_2 \times \mathbf{N}'_2) \quad (11)$$

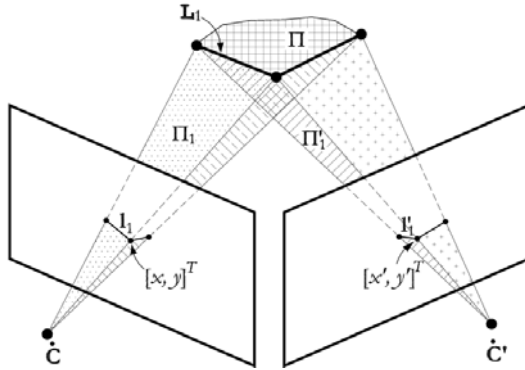


Fig. 3. Any corresponding two edges L_1 and L'_1 of a junction can be used to reconstruct their actual 3D line L_1 in space. Hence, the equation Π of the junction plane can be estimated.

where \times denotes the cross product and \mathbf{N} should be normalized to unit magnitude. Note that the intersection of the 3D lines L_1 and L_2 is the location of the junction in space. This can be estimated as [12]:

$$\dot{\mathbf{M}} = \left([\mathbf{I} - \dot{\mathbf{M}}_\infty \dot{\mathbf{M}}_\infty^T] + [\mathbf{I} - \dot{\mathbf{M}}'_\infty \dot{\mathbf{M}}'^T_\infty] \right)^{-1} \left(\dot{\mathbf{C}} + \dot{\mathbf{C}}' - [\dot{\mathbf{C}}^T \dot{\mathbf{M}}_\infty] \dot{\mathbf{M}}_\infty - [\dot{\mathbf{C}}'^T \dot{\mathbf{M}}'_\infty] \dot{\mathbf{M}}'_\infty \right) \quad (12)$$

where $\dot{\mathbf{M}}$, $\dot{\mathbf{C}}$, $\dot{\mathbf{C}}'$, $\dot{\mathbf{M}}_\infty$ and $\dot{\mathbf{M}}'_\infty$ are the inhomogeneous coordinates of the junction point in space, the optical centers and the intersection of the rays with the plane at infinity respectively; \mathbf{I} is a 3×3 identity matrix and $\dot{\mathbf{M}}_\infty$ and $\dot{\mathbf{M}}'_\infty$ are normalized to unit magnitude. Consequently, the perpendicular distance to the origin from the junction plane can be written as:

$$d = -\mathbf{N} \bullet \dot{\mathbf{M}} \quad (13)$$

where \bullet represents the dot product. Hence, the equation of the junction plane is:

$$\Pi = \begin{bmatrix} \mathbf{N} \\ d \end{bmatrix} \quad (14)$$

3 Experimental Results

Fig. 4(a) and (b) shows a wide baseline test pair of 640×480 images. Using the approximate camera parameters, the fundamental matrix \mathbf{F} is estimated using Eq. (2). Hence, an epipolar line in the one image can be estimated for each junction in the other image. A strip is considered surrounding each epipolar line as a range for putative matches. The affine matrix of Eq. (6) and correlation of Eq. (8) are applied to putative matches.

Consider the junctions indicated in Fig. 4(a) and 4(b). Fig. 4(c) shows the *left* junction while Fig. 4(d) shows the *right* junction after transformation. Also,

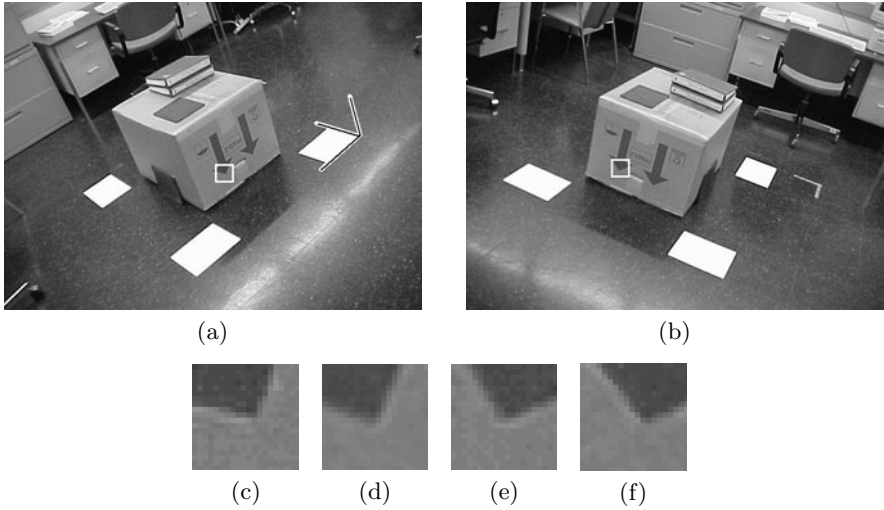


Fig. 4. A pair of 640×480 wide baseline images (format size 3.5311×2.7446 mm). The approximate camera parameters are $f = 4.3$ mm, $C_x = -1535.21/-1694.56$ mm, $C_y = 1720.84/-142.25$ mm, $C_z = 1693.22/1585.19$ mm, $\omega = -35.16/25.03^\circ$, $\phi = -46.77/-55.56^\circ$, $\kappa = -155.57/-47.08^\circ$. (a) and (b) Original images with corresponding junctions indicated and the origin of the world coordinate system. (c) Original left junction magnified. (d) Right junction transformed using Eq. (6). (e) Original right junction magnified. (f) Left junction transformed using Eq. (6).

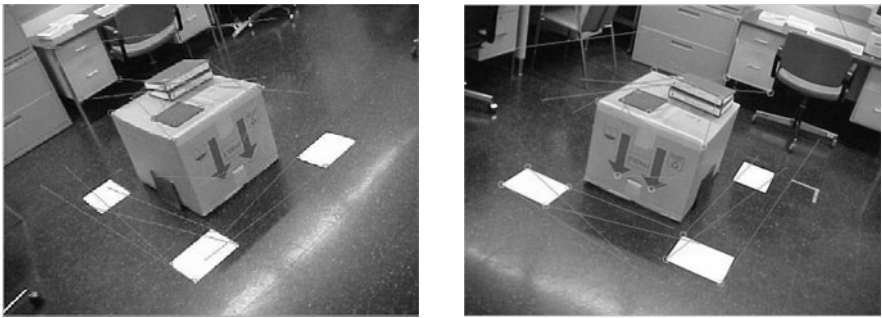


Fig. 5. Corresponding points where disparity vectors are shown

Fig. 4(e) shows the *right* junction while Fig. 4(f) shows the *left* junction after transformation. Hence, the patches displayed in Fig. 4(c) and 4(d) can be correlated. Also, correlation can be applied to patches in Fig. 4(e) and 4(f). A correct match is declared when the correlation value obtained is the best in both directions. The overall results are shown in Fig. 5. Applying the steps of Sec. 2.5, the equation of the planar patch shown in Fig. 4 is $\Pi = [0.98, 0.02, 0.19, -331.15]^T$. The 3D location of the junction is $\mathbf{M} = [298.77, 706.12, 135.96]^T$ (millimeters). Compare this to the location of the world coordinate system origin.

4 Conclusion

Given a stereo pair of wide baseline or sparse view images, we presented an algorithm to achieve good wide baseline matching and to derive the equations of planar patches when approximate camera parameters are known. The approach detects junctions in both images using the JUDOCA operator. It uses the information of the junction orientation to establish an affine transformation matrix that is used with the estimate of the fundamental matrix to get point correspondences at the junction locations. The projections of the junction edges are used to obtain the equation of the junction plane in space.

References

1. Kanazawa, Y., Uemura, K.: Wide Baseline Matching using Triplet Vector Descriptor. In: Proc. BMVC, Edinburgh, UK, vol. I, pp. 267–276 (2006)
2. Bay, H., Ferrari, V., Gool, L.V.: Wide-Baseline Stereo Matching with Line Segments. In: Proc. CVPR, San Diego, CA, USA, vol. 1, pp. 329–336 (2005)
3. Lowe, D.: Object recognition from local scale invariant features. In: Proc. ICCV, Kerkyra, Greece, vol. 2, pp. 1150–1157 (1999)
4. Andreasson, H., Treptow, A., Duckett, T.: Localization for mobile robots using panoramic vision, local features and particle filters. In: Proc. ICRA, Barcelona, Spain, pp. 3348–3353 (2005)
5. Tuytelaars, T., Gool, L.V.: Matching Widely Separated Views based on Affine Invariant Regions. *IJCV* 59, 61–85 (2004)
6. Elias, R.: Wide baseline matching through homographic transformation. In: Proc. ICPR, Washington, DC, USA, vol. 4, pp. 130–133. IEEE Computer Society Press, Los Alamitos (2004)
7. Laganière, R., Elias, R.: The Detection of Junction Features in Images. In: Proc. ICASSP, Montréal, Québec, Canada, vol. III, pp. 573–576. IEEE, Los Alamitos (2004)
8. Elias, R.: Modeling of Environments: From Sparse Views to Obstacle Reconstruction. LAP Lambert Academic Publishing, Germany (2009) ISBN: 3838322207
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
10. Elias, R.: Geometric modeling in computer vision: An introduction to projective geometry. In: Wah, B. (ed.) Wiley Encyclopedia of Computer Science and Engineering, vol. 3, pp. 1400–1416. John Wiley & Sons, Inc., Chichester (2008)
11. Laganière, R., Labonté, F.: Stereokineopsis: A survey. Technical Report CRPR-RT-9603, École Polytechnique de Montréal, Groupe de Recherche en Perception et Robotique (1996)
12. Beardsley, P., Zisserman, A., Murray, D.: Sequential update of projective and affine structure from motion. *IJCV* 23, 235–259 (1997)

Single Camera Stereo System Using Prism and Mirrors*

Gowri Somanath, Rohith MV, and Chandra Kambhamettu

Video/Image Modeling and Synthesis (VIMS) Lab, Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA

<http://vims.cis.udel.edu>

Abstract. Stereo and 3D reconstruction are used by many applications such as object modeling, facial expression studies and human motion analysis. But synchronizing multiple high frame rate cameras require special hardware or sophisticated software solutions. In this paper, we propose a single camera stereo system by using a setup made of prism and mirrors. Our setup is cost effective, portable and can be calibrated similar to two camera stereo to obtain high quality 3D reconstruction. We demonstrate the application of the proposed system in dynamic 3D face expression capture, depth super-resolution in stereo video and general depth estimation.

1 Introduction

The knowledge of depth in a scene has been known to simplify many problems in computer vision, for example, face and expression recognition [1,2], tracking [3,4], object discovery [5], 3D video content creation and many other applications. Stereo systems are easy to setup in cases where the scene is stationary. For many applications, high frame rate or video cameras are required. Synchronizing multiple such cameras is a challenging task and requires expensive hardware or sophisticated software solutions. For newly deployed systems, the above solutions can be employed. But in systems where cameras have already been deployed, such as surveillance, changing the existing camera setup is not always practical. Portability and maintenance of multicamera setups is harder than single camera. On the other hand, the calibration of stereo/multi-view systems provides true scale reconstruction which is useful and even essential in many real world applications. A portable, cost effective and single camera system which can be calibrated like stereo to provide metric reconstruction is hence desirable. In this paper, we present a solution using prism and two mirrors, which can be used to turn almost any camera into a stereo camera. The system can be calibrated to obtain high quality metric reconstruction. The proposed solution is cost effective and can be easily used as a detachable accessory for a camera lens. An equilateral prism and two first surface mirrors are used to build the system shown in

* This work was made possible by NSF Office of Polar Program grants, ANT0636726 and ARC0612105.

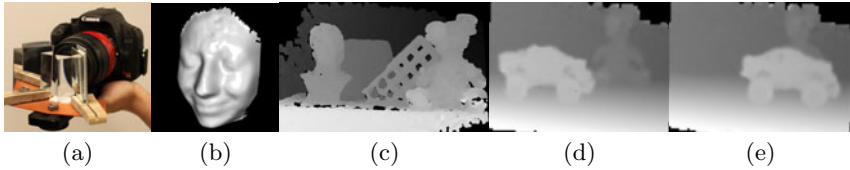


Fig. 1. (a) Our camera setup, (b) 3D reconstruction from a facial expression video sequence, (c) Disparity map of a sample scene, (d),(e) Two frames from a sequence of super-resolved disparity maps of a dynamic scene. The stereo video is 640 X 720 pixels, which is super-resolved to 3168 X 2376 pixels using a low frame rate, motion compensated full resolution still image.

Figure 1(a). The camera used is a Canon T1i DSLR, which can capture HD video (720p) at 30fps and 15MP still images.

The paper is organized as follows. In Section 2 we discuss existing systems. We detail the proposed setup in Section 3. We demonstrate the use of the system for two types of applications which have different needs. In Section 4.1 we show how the system can be used to perform depth super-resolution of dynamic scenes. We present calibrated 3D reconstruction results in Section 4.2. Section 4.3 provides more results and we conclude in Section 5.

2 Previous Work

The idea of converting single camera into stereo using a combination of mirrors, lenses and prisms has been proposed earlier by many researchers. Figure 2 shows a schematic comparison of the different existing setups. One of the oldest known systems, known as the Stereoscopic Transmitter, was built in 1894 by Theodore Brown. A similar system was proposed by Gluckman and Nayar [6,7], which used a pair of hinged mirrors placed in front of the camera with the reflecting surface facing the lens. The scene being imaged was to the back and one side of the system. The placement of prisms and mirrors at considerable distance away from the lens, makes the above system less portable. Also the idea cannot be easily implemented to convert existing single camera systems, like surveillance, to stereo systems. Many catadioptric stereo systems have been proposed using hyperbolic [8] and parabolic mirrors [9]. Nene and Nayar [10] provide some possible configurations using planar, parabolic, elliptic, and hyperbolic mirrors. Mitsumoto et al. [11] and Zhang and Tsui [12] propose a system which images the object and its reflection in a set of planar mirrors. Pyramidal arrangement of planar mirrors was used in [13]. Gluckman and Nayar also proposed systems using single and three mirror configurations [14,15], where the system was constrained to produce rectified images instead of 3D sensitivity (number of levels in disparity map). Gao and Ahuja [16] used a single camera and a rotating planar plate in front of the camera, similar to that proposed by Nishimoto and Shirai [17]. Due to need to capture multiple images with two or more different plate positions, these systems are restricted to static scenes. In [18], Lee et al. used a

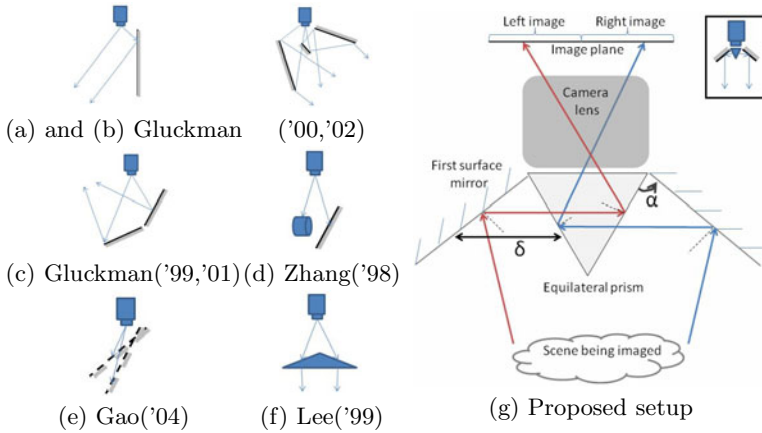


Fig. 2. (a)-(f) Schematic of previous systems. (g) Schematic top view of the proposed mirror and prism arrangement.

biprism placed 15cm in front of the camera to obtain two views of the scene. The idea was generalized in [19] using trinocular prisms. [20,21] proposed a setup using a prism and two mirrors, where the prism edge was used as a reflective surface. This required the use of specialized prisms and also resulted in a setup where the prism had to be kept 12-13cm from the lens of the camera. This made the system less portable and difficult to be used as an add-on attachment for existing systems. Also, due to use of specialized prism, the system is expensive. A commercial product, Tri-Delta stereo adapter, works on similar principle as our system. However the adapter is mounted on the lens and the camera would be placed such that the lens points towards the top (sky) or bottom (ground), to image the scene in front of the observer. Our setup contrasts the adapter in the following manner. The tri-delta is a completely mirror based setup and is compatible with only a few point-and-shoot cameras. Our setup is flexible in terms of baselines and vergence control, which is essential for machine vision studies. Similar to many previous works, the Tri-Delta would also require changes to the way a camera is setup with respect to the viewed scene. Also the cost of our construction is only a fraction of that of the tri-delta. A different approach to obtain stereo video is used in 'NuView Adapter' available for video cameras only. The adapter uses dual LCD shutters, a beamsplitter and a mirror arrangement to provide left and right views as odd and even fields. Our setup is much easier to construct, can be used with both still and video cameras and is much more cost effective.

3 Proposed System

In this paper, we propose a single camera system with the following characteristics:(1) Simple and portable construction: The proposed extension to a camera can be used as a lens attachment. The prism and mirror setup mount in front of

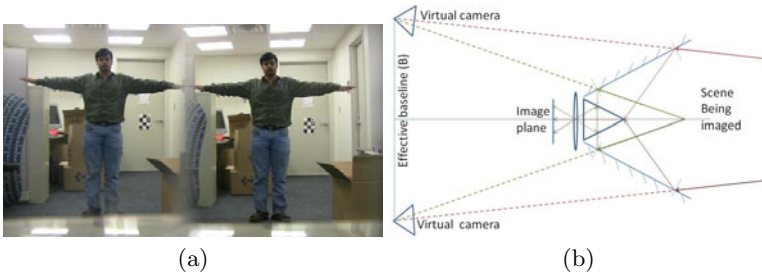


Fig. 3. (a) Raw image captured by our setup. The stereo images are extracted as two halves of the raw image. (b) Ray diagram illustrating the formation of the two images, the virtual cameras and baseline.

the lens and hence the entire system is portable. (2) Easy to deploy on existing single camera systems: Many applications like surveillance and monitoring have single cameras already in place. The proposed setup can be easily attached to such cameras to obtain stereo video/images. Unlike many previously proposed systems, the position of imaged scene with respect to camera does not have to be changed. (3) Can be calibrated to obtain metric reconstruction: Some applications only use depth separation or disparity information, while many others require true scale metric 3D reconstruction (for example, 3D modeling of objects, person tracking etc). The proposed setup can be calibrated like a two camera stereo system. (4) Flexible: Baseline and vergence are some of the flexible parameters in a stereo setup. As will be explained later, the placement of the mirrors can be varied to change the baseline and vergence in our system.

3.1 Setup

The setup is made using an equilateral prism and two first surface mirrors. As shown in Figure 2(g), the prism is placed at the end of the camera lens. The mirrors are placed at an angle α , with respect to the corresponding prism surface. The distance δ , between the mirror and prism can also be varied. Different configurations of α and δ can be used to obtain suitable baseline, vergence and overlap between the left and right images. Intuitively, smaller α and δ are used when objects being imaged are within a few feet of the camera. A detailed analysis of the geometry is discussed in the following section. We have attached the setup to various types of cameras including DSLR, camcorder and network/IP cameras with wide angle lens. Figure 1(a) shows the setup with a DSLR camera. In our experiments, we used a prism of edge length two inches and base of one inch. The mirror is 2 inches by 2 inches.

3.2 Analysis of the Setup

Figure 3(b) shows the ray diagram illustrating the stereo image formation and virtual cameras. We now derive the conditions on the mirror angle and the

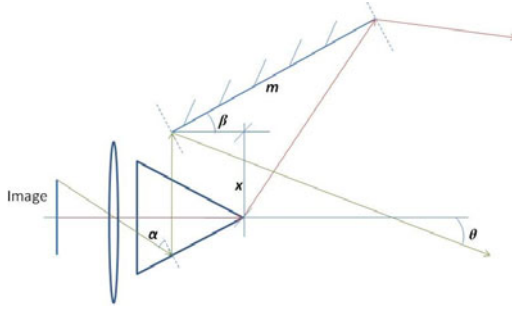


Fig. 4. Ray diagram showing angles and distances used in the analysis (Section 3.2)

effective baseline of our setup. Figures 3(b) and 4 show the ray diagrams with angles and distances used in the following analysis. We define the following angles and distances in the setup: ϕ is the horizontal field of view of camera in degrees, α is the angle of incidence at prism, β is the angle of inclination of mirror, θ is the angle of scene ray with horizontal, x is the distance between mirror and camera axis, m is the mirror length and B is the effective baseline (Fig. 3(b)). To calculate the effective baseline, we trace the rays in reverse. Consider a ray starting from the image sensor, passing through the camera lens and incident on the prism surface at an angle α . This ray will then get reflected from the mirror surface and go towards the scene. The final ray makes an angle of θ with the horizontal as shown in the figure. It can be shown that $\theta = 150 - 2\beta - \alpha$. In deriving the above, it has been assumed that there is no inversion of image from any of the reflections. This assumption is violated at large fields of view. More specifically, it is required that $\phi < 60^\circ$. Since we are not using any lens apart from the camera lens, the field of view in resulting virtual cameras should be exactly half of the real camera. In Figure 3(b), consider two rays from the image sensor, one ray from the central column of image ($\alpha_0 = 60^\circ$, shown in green) and another ray from the extreme column ($\alpha = 60^\circ - \phi/2$, shown in red). The angle between the two scene rays is then $\phi/2$. For stereo, the images from the two mirrors should contain some common part of the scene. Hence, the scene rays must be towards the optical axis of the camera rather than away from it. Also, the scene rays must not re-enter the prism as this would lead to internal reflection and not provide an image of the scene. Applying the two conditions above, we can show that the inclination of the mirror is bound by the following inequality $\phi/4 < \beta < 45^\circ + \phi/4$. We can now estimate the effective baseline based on the angle of scene rays, mirror length and distance of the mirror from the axis as follows.

$$B = 2 \frac{x \tan(2\beta - \phi/2) - m \cos(\beta) - (x + m \cos(\beta)) \tan(2\beta)}{\tan(2\beta - \phi/2) - \tan(2\beta)}.$$

In our setup, the parameters used were focal length of 35 mm corresponding to $\phi = 17^\circ$, $\beta = 49.3^\circ$, $m = 76.2\text{mm}$ and $x = 25.4\text{mm}$. The estimated baseline is

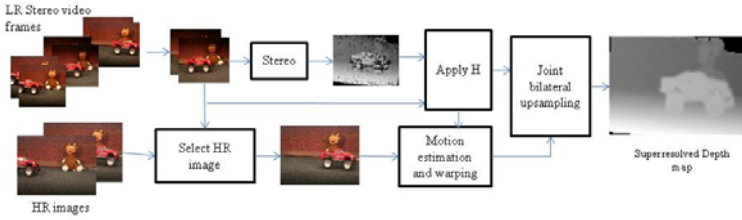


Fig. 5. Depth super resolution pipeline

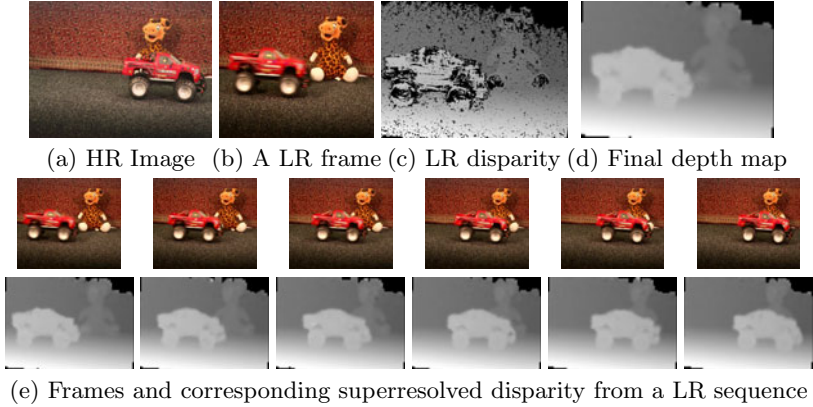


Fig. 6. Super resolution of stereo video

49.62mm which is close to the obtained calibration value of 48.15mm (see Section 4.2). Varying the mirror angle provides control over the effective baseline as well as the vergence of the stereo system.

4 Applications and Results

We demonstrate the system with three applications with different requirements. First, we use the system for depth super resolution of dynamic scenes, which utilizes the high frame rate stereo video capability. Next we apply the calibrated system for the case of metric 3D reconstruction of dynamic and close range scenes. Lastly we show stereo results for different distances ranging from close range (within a feet), medium range (within 5 feet) and far range outdoor scene. We include results for static and dynamic scenes.

4.1 Depth Super Resolution

In this section we discuss the first application of the system for depth super resolution. The resolution of depth map and the time taken to capture is inversely related for most sensors. For example, laser scanners can be used to obtain high

quality and resolution depth maps but are limited to static scenes due to the large scanning time required. For dynamic scenes, methods such as stereo/multi-view or time-of-flight cameras have to be used, which are restricted by the spatial resolution possible for a given frame rate. Many schemes have been thus proposed to enhance the resolution of the depth maps obtained by the above systems, called depth super resolution. In general, the process involves noise filtering, hole filling at missing pixels and enhancing the spatial resolution of the depth map. The depth map is usually upsampled using a higher resolution color image of the scene or using multiple low resolution depth maps taken from slightly different view points. Some of the works using time-of-flight cameras include [22,23,24,25]. Most of these works have been only demonstrated on static scenes. Recently, [24] was extended for dynamic scenes [26]. Many algorithms have been proposed for the case of stereo systems as well [27,28,29]. In [28], the proposed joint bilateral filter was used to upsample both the images and the depth maps. In [29], a hybrid camera setup with low resolution, high frame-rate stereo pair and one high resolution, low frame rate camera was proposed. But the authors demonstrated depth super resolution only for a static case. The DSLR used in our setup is capable of 15MP still image capture at 1fps, during video capture (720 lines at 30fps).

We perform super resolution for a dynamic scene by applying a cross (or joint) bilateral filter with the edge information from the high resolution image. Since that image is only captured at certain instances, we must estimate the motion and warp it to correspond to each stereo frame. The pipeline is shown in Figure 5. Unlike other reported systems, we do not use a separate camera to capture the high resolution (HR) image. The prism setup provides a stereo image even in the still capture. We use one image as the HR image, thus the transform between the low resolution (LR) image and the HR image is a 2D homography. The camera does not use the entire sensor for video capture, since the aspect ratio is different for High-Definition video capture and still image. To obtain the homography H , we use a pre-calibration step with a known pattern in the scene. Using correspondence between the HR image and a LR frame, we estimate the transform H for suitable translation and scaling of the LR frame to HR image. This transformation is fixed given the prism-mirror arrangement and the camera lens zoom, and is thus calculated only once for a given setup. The HR images were captured at random instances for the sequence and are thus non-equally spaced in time. For each LR frame we select the HR image closest in time. We transform the reference stereo frame to the HR image using H and estimate the motion between this pair. Due to the blur caused by the motion of the object and interpolation during the transform, pixel based optic flow schemes do not provide robust results. To maintain high quality edges in the HR image, the warping has to be correct. Since the motion is rigid in our case, we perform motion analysis at the object level. We initially capture an image of the setup without the moving objects. Using this as a reference background image, we obtain the moving objects in the scene. To eliminate errors due to shadows, lighting changes or noise, we use dense SIFT matching instead of simple

Table 1. Calibration results




Errors (mm, mm^2)	Plane pose			
Triangulation error	Mean(Variance)	0.0467 (0.0012)	0.0585 (0.0018)	0.0519 (0.0016)
Plane fit error	Mean(Variance)	0.4076 (0.0965)	0.4324 (0.1891)	0.3429 (0.0866)

image subtraction at this stage. Another feature matching step is used between the objects detected in the LR and HR images to obtain the motion for each individual object. The HR image is then warped to compensate for the object motion. The background image is used to fill the holes in the warped image. The LR stereo pair is rectified using the method proposed in [30] and the initial disparity map is obtained using the Growing Correspondence Seeds method [31]. We then apply joint/cross bilateral filter on the transformed disparity map and the motion compensated HR image. For the disparity map D and HR image I , the joint bilateral filter can be written as

$$D(i) = \frac{1}{W_i} \sum_{j \in N} G_s(\|i - j\|) G_r(\|I(i) - I(j)\|) D'(j)$$

where i, j are i^{th} and j^{th} pixel and N is the predefined neighborhood for a pixel. G_s and G_r are the spatial and range kernel centered at the pixel i and the disparity at i respectively. W_i is the normalization factor defined as, $W_i = \sum_{j \in N} G_s(\|i - j\|) G_r(\|I(i) - I(j)\|)$. The image edge constraint $\|I(i) - I(j)\|$ can be defined in different ways, we use the grayscale value difference at the pixels. Figure 6 shows results for a scene with a toy car moving. Figure 6(a) shows a captured HR image, (b) shows left image of a LR stereo frame, (c) shows the initial disparity map for the image in (b), and (d) the corresponding final super resolved disparity map. Figure 6(e) shows more frames from the sequence, which used HR image in (a) for super resolution (motion compensated as required). Note that we have only showed some intermediate frames from the sequence.

4.2 3D Reconstruction

Here we describe the use of the system for 3D reconstruction which has wide ranging applications in 3D immersive technology [32], face and expression recognition [12], action recognition [33], archeology [34]. Many of the above applications demand portability without sacrificing the quality of the reconstruction. Systems with two or more cameras have been proposed for the above, which have been successful in lab setting. In order to employ the algorithms for 3D face analysis, action recognition and others in their real environments such as court rooms or airports, a large change in the camera setup is impractical. Most environments have existing single camera installations in place. Our setup can provide a cost effective and simple mechanism to convert those to stereo. As discussed before, previously proposed single camera stereo systems would require

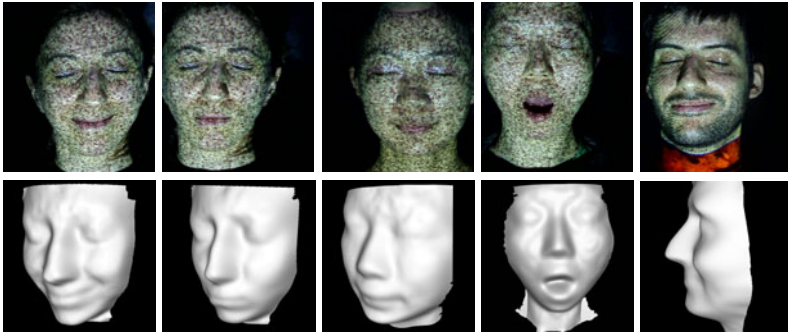


Fig. 7. Snapshots from facial expression video. Top: Reference stereo image, Bottom: Corresponding 3D reconstruction.

change in mounting, cannot be attached easily to the lens of a camera or are have limits on size of object being imaged.

We calibrated our system using a standard stereo calibration technique [35]. Camera parameters were iteratively refined until the reprojection error converged. In our experiments the RMS reprojection error converged to 0.2 pixels. The quality of calibration and stereo is measured using triangulation uncertainty and reconstruction error for a known geometry. The stereo triangulation and plane fitting error for different poses of a plane in the calibrated range is show in Table 1. The triangulation error is the smallest distance between the rays from the two cameras. It must be ideally zero, but due to finite precision in pixel matching, the rays will not intersect. The plane fitting is done using RANSAC such that more than 90% of the points are inliers. All distances are in millimeters(mm). The calibration and plane fitting results clearly show that high accuracy reconstructions can be obtained by calibrating the proposed setup. Some results from face video reconstruction are shown in Figure 7.

4.3 More Results

Disparity maps for other scenes are shown in Figure 8. We obtain approximately 150 levels of disparity for the outdoor tree image and 90 levels for the indoor setup. Figure 9 shows disparity maps for an action sequence, with 100 levels

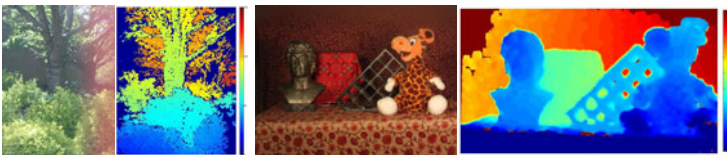


Fig. 8. Disparity maps for general scenes. Left: An outdoor scene and corresponding disparity map, Right: An indoor scene and depth map.

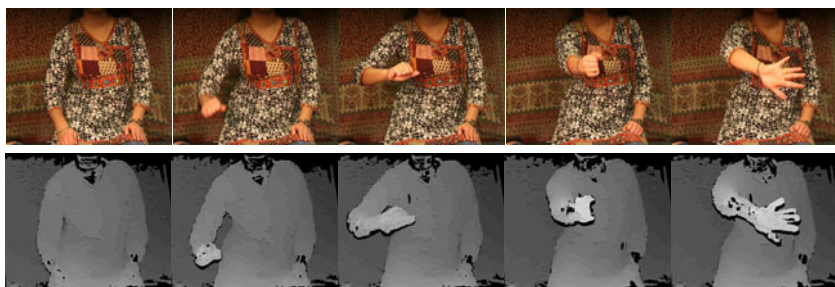


Fig. 9. Disparity maps for an action sequence. Top: Sample captured stereo images, Bottom: Corresponding disparity maps.

obtained between the tip of hand (last image) and background. The quality of results demonstrate that our setup can be used in applications such as autonomous navigation in robots [36], scene understanding [5] and action recognition [33]. For robotic applications, our system can be more effective than using multiple cameras due to limited space and resources on a mobile platform. Our setup also offers flexibility in terms of baseline and vergence, which allows tuning the depth sensitivity to match the application needs. It can also be used in other application such as gesture recognition [37,38] for human-machine interface and 3D immersive technology [32], where the use of a single camera in lieu of multiple cameras greatly simplifies the setup.

5 Conclusion

In this paper, we proposed a setup with prism and mirrors, which allows a single camera to be converted to stereo. The proposed setup was shown to have various key advantages over previous systems. Our setup can be easily attached in front of any camera. Since the prism is placed close to the lens, the portability of the system is not affected. Existing single camera setups can be easily converted to stereo, since the viewing direction of the camera remains unaltered, unlike many of the earlier works. The cost and complexity of construction is much less compared to some commercially available stereo solutions. We demonstrated the use of the system for depth superresolution of stereo video sequences of dynamic scenes. We also showed that high quality 3D reconstructions can be obtained by calibrating the camera. Disparity maps for general indoor and outdoors scenes were provided to show applicability in other areas such as scene understanding, human-machine interface and action recognition.

References

1. Heseltine, T., Pears, N., Austin, J.: Three-dimensional face recognition using combinations of surface feature map subspace components. *Image Vision Comput.* 26, 382–396 (2008)

2. Wang, J., Yin, L., Wei, X., Sun, Y.: 3d facial expression recognition based on primitive surface feature distribution. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 1399–1406 (2006)
3. Pellegrini, S., Iocchi, L.: Human posture tracking and classification through stereo vision and 3d model matching. *J. Image Video Process.* 2008, 1–12 (2008)
4. Sogo, T., Ishiguro, H., Trivedi, M.M.: Real-time target localization and tracking by n-ocular stereo. In: OMNIVIS 2000: Proceedings of the IEEE Workshop on Omnidirectional Vision, p. 153 (2000)
5. Somanath, G., Rohith, M.V., Metaxas, D., Kambhamettu, C.: D - clutter: Building object model library from unsupervised segmentation of cluttered scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009 (2009)
6. Gluckman, J., Nayar, S.K.: Planar catadioptric stereo: Geometry and calibration. *Computer Vision and Pattern Recognition*. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 1022 (1999)
7. Gluckman, J., Nayar, S.K.: Catadioptric stereo using planar mirrors. *Int. J. Comput. Vision* 44, 65–79 (2001)
8. Chaen, A., Yamazawa, K., Yokoya, N., Takemura, H.: Omnidirectional stereo vision using hyperomni vision. In: Technical Report 96-122, IEICE (1997) (in Japanese)
9. Gluckman, J., Nayar, S.K., Thoresz, K.J.: Real-time omnidirectional and panoramic stereo. In: Proceedings of the 1998 DARPA Image Understanding Workshop, pp. 299–303. Morgan Kaufmann, San Francisco (1998)
10. Nene, S.A., Nayar, S.K.: Stereo with mirrors. In: ICCV 1998: Proceedings of the Sixth International Conference on Computer Vision, Washington, DC, USA, p. 1087. IEEE Computer Society, Los Alamitos (1998)
11. Mitsumoto, H., Tamura, S., Okazaki, K., Kajimi, N., Fukui, Y.: 3-d reconstruction using mirror images based on a plane symmetry recovering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 941–946 (1992)
12. Zhang, Z.Y., Tsui, H.T.: 3d reconstruction from a single view of an object and its image in a plane mirror. In: International Conference on Pattern Recognition, vol. 2, p. 1174 (1998)
13. Kawanishi, T., Yamazawa, K., Iwasa, H., Takemura, H., Yokoya, N.: Generation of high-resolution stereo panoramic images by omnidirectional imaging sensor using hexagonal pyramidal mirrors. *International Conference on Pattern Recognition* 1, 485 (1998)
14. Gluckman, J., Nayar, S.K.: Rectified catadioptric stereo sensors. *Computer Vision and Pattern Recognition*. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 2380 (2000)
15. Gluckman, J., Nayar, S.K.: Rectified catadioptric stereo sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 224–236 (2002)
16. Gao, C., Ahuja, N.: Single camera stereo using planar parallel plate. In: 17th International Conference on Pattern Recognition, ICPR 2004, vol. 4, pp. 108–111 (2004)
17. Nishimoto, Y., Shirai, Y.: A feature-based stereo model using small disparities. In: Proc. Computer Vision and Pattern Recognition, pp. 192–196 (1987)
18. Lee, D.H., Kweon, I.S., Cipolla, R.: A biprism-stereo camera system. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 1082 (1999)
19. Xiao, Y., Lim, K.B.: A prism-based single-lens stereovision system: From trinocular to multi-ocular. *Image Vision Computing* 25, 1725–1736 (2007)
20. Duvioubourg, L., Ambellouis, S., Cabestaing, F.: Single-camera stereovision setup with orientable optical axes. In: International Conference Computer Vision and Graphics (ICCVG), 173–178 (2004)

21. Lu, T., Chao, T.H.: A single-camera system captures high-resolution 3d images in one shot. *SPIE Newsroom* (2006), doi: 10.1117/2.1200611.0303
22. Kuhnert, K.D., Stommel, M.: Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4780–4785 (2006)
23. Yang, Q., Yang, R., Davis, J., Nister, D.: Spatial-depth super resolution for range images. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
24. Zhu, J., Wang, L., Yang, R., Davis, J.: Fusion of time-of-flight depth and stereo for high accuracy depth maps. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pp. 1–8 (2008)
25. Schuon, S., Theobalt, C., Davis, J., Thrun, S.: Lidarboost: Depth superresolution for tof 3d shape scanning. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 343–350 (2009)
26. Zhu, J., Wang, L., Gao, J., Yang, R.: Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 899–909 (2010)
27. Sawhney, H.S., Guo, Y., Hanna, K., Kumar, R., Adkins, S., Zhou, S.: Hybrid stereo camera: an ibr approach for synthesis of very high resolution stereoscopic image sequences. In: *SIGGRAPH 2001: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001)
28. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. In: *SIGGRAPH 2007: ACM SIGGRAPH 2007 Papers*, p. 96 (2007)
29. Li, F., Yu, J., Chai, J.: A hybrid camera for motion deblurring and depth map super-resolution. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pp. 1–8 (2008)
30. Fusiello, A., Irsara, L.: Quasi-euclidean uncalibrated epipolar rectification. In: *International Conference on Pattern Recognition, ICPR* (2008)
31. Čech, J., Šára, R.: Efficient sampling of disparity space for fast and accurate matching. In: *BenCOS 2007: CVPR Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*, IEEE, Los Alamitos (2007)
32. Towles, H., Chen, W.C., Yang, R., Kum, S.U., Fuchs, H., Kelshikar, N., Mulligan, J., Daniilidis, K., Holden, L., Seleznik, B., Sadagic, A., Lanier, J.: 3d tele-collaboration over internet2. In: *International Workshop on Immersive Telepresence, ITP* (2002)
33. Roh, M.C., Shin, H.K., Lee, S.W.: View-independent human action recognition with volume motion template on single stereo camera. *Pattern Recognition Letters* 31, 639–647 (2010)
34. Bitelli, G., Girelli, V.A., Remondino, F., Vittuari, L.: The potential of 3d techniques for cultural heritage object documentation. *Videometrics IX* 6491 (2007)
35. Strobl, K.H., Sepp, W., Fuchs, S., Paredes, C., Arbter, K.: DLR CalDe and DLR CalLab
36. Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. *Autonomous Robots* 8, 161–171 (2000)
37. Agarwal, A., Izadi, S., Chandraker, M., Blake, A.: High precision multi-touch sensing on surfaces using overhead cameras. In: *International Workshop on Horizontal Interactive Human-Computer Systems*, pp. 197–200 (2007)
38. Shimizu, M., Yoshizuka, T., Miyamoto, H.: A gesture recognition system using stereo vision and arm model fitting. *International Congress Series*, vol. 1301, pp. 89–92 (2007)

A Region-Based Randomized Voting Scheme for Stereo Matching

Guillaume Gales¹, Alain Crouzil¹, and Sylvie Chambon²

¹ IRIT, Institut de Recherche en Informatique de Toulouse, France
{gales,crouzil}@irit.fr

² LCPC, Laboratoire Central de Ponts et Chaussées, Nantes, France
chambon@lcpc.fr

Abstract. This paper presents a region-based stereo matching algorithm which uses a new method to select the final disparity: a random process computes for each pixel different approximations of its disparity relying on a surface model with different image segmentations and each found disparity gets a vote. At last, the final disparity is selected by estimating the mode of a density function built from these votes. We also advise how to choose the different parameters. Finally, an evaluation shows that the proposed method is efficient even at sub-pixel accuracy and is competitive with the state of the art.

1 Introduction

Pixel matching of epipolar rectified image pair consists in finding for each pixel $\mathbf{p}_{i,j}^l$ at the i^{th} row and j^{th} column in the left image the disparity value d such that the pixel $\mathbf{p}_{i,j+d}^r$ in the right image is the projection of the same scene element as $\mathbf{p}_{i,j}^l$. They are two major families of pixel matching methods: global, based on mathematical optimization techniques to minimise the errors made by the disparity d , and local, based on correlation measures between the neighbourhood of the considered pixel in the left image and the neighbourhood of each candidate in the right image within a predefined search area [1]. Both techniques are usually combined in *region-based* methods which are the best methods in the Middlebury [2] evaluation. They consist in computing a region map of the input image with a color segmentation algorithm, then estimating the parameters of a surface model for every region assuming that it corresponds to a same object. At last, a final step involving global optimization is performed to refine the result. Such an approach helps to compute disparities for homogeneous and occluded areas (which are difficult to match with a correlation-based approach). Nevertheless, the computed disparities are based on the chosen surface model which may not always give the best representation, for instance, if a plane model is chosen to fit a conic surface. Thus, we propose a region-based algorithm which uses a new method to select the final disparity: for each pixel, different disparity approximations are computed based on a surface model with different image

¹ vision.middlebury.edu/stereo/

segmentations and different random seeds. Each found disparity gets a vote. Ultimately, the final disparity is selected by estimating the mode of a density function built from these votes.

2 Previous Work

The principle of region-based methods [2,3,4,5,6,7,8,9] relies on the following hypothesis: Pixels within a homogeneous color region of an image belongs to the same object. The use of the segmentation presents the following advantages: it can help to match homogeneous areas. It can also help to avoid too many boundary artefacts at depth discontinuities. The regions can be approximated by a surface model usually a plane or a B-spline. The plane model is simple and gives good approximations especially with small regions but may not be sufficient when dealing with non-polyhedral surface. The B-spline model can give better results than the plane model when regions do not correspond to flat objects [2,3,4]. However, it is harder to configure (the order of the B-spline and the knot vector must be chosen) and big oscillations in the estimated disparities can appear [5,10].

Region-based methods go through the following steps : (1) Color segmentation of the image. (2) Computation of an initial disparity map. (3) Initial estimation of the chosen model parameters. (4) Refinement of the results.

For the first step, the mean-shift algorithm [11] is commonly used [2,3,4,5,9,7]. An over-segmentation is usually performed especially when the plane model is chosen. To obtain different disparity candidates with both plane and B-spline models, the authors of [5] use different segmentations. For the second step, the initial disparity map is computed. A local correlation-based method with correlation measures is usually employed to find the initial disparities. Different methods such as an adaptive correlation window technique [2,4] or a specific correlation measure [3] can be applied to obtain less disparity errors in the initial map. Then, in order to calculate the model parameters from the initial disparities, the weighted least-squared method is employed iteratively by [9] where the weights are given by a confidence measure based on the difference between current disparities and previous estimated ones. Robust estimation of the parameters can be performed by the RANSAC algorithm [4,8]. This algorithm relies on a stochastic process and the authors of [2,3] propose to calculate the parameters in a non-stochastic robust manner by finding the best value for each parameter one after the other. Finally, two different kinds of refinements can be distinguished:

- *Plane assignment*– The different disparity planes are extracted and grouped according to the disparity estimations and each pixel is assigned to its most probable plane [9,5,12,3]. This assignation is made by global optimization (graph cut [9,5], belief propagation [3], greedy algorithm [12]).
- *Parameter refinement* – The parameters of the model can also be refined using global optimization (cooperative optimization [2], belief propagation [8]).

3 Proposed Algorithm

First of all, we propose to use a robust correlation measure near occluded areas in step 2 to reduce the number of errors in these areas (see §2). Then, the basic idea of our algorithm, is to replace steps 3 and 4 by using for each pixel a randomized voting scheme for different disparity approximations leading to a density function where the mode, i.e. the most occurring disparity, is taken as the final disparity. This approach is simple: it is easy to implement, it does not require any global optimization that relies on a cost function which usually requires to define constraints to obtain good results. The more the number of constraints is, the best the results are. However, the computation time is increased, so does the number of tuning parameters. Besides, optimization methods are usually iterative while our randomized process can be achieved in parallel.

3.1 Initial Disparities

The initial disparities are the data used to compute the different approximations (based on a plane model in our case). Thus, the initial disparity map does not need to be dense. However, a good ratio of correct disparities is needed within each region to be able to find good approximations.

We use a correlation-based matching method (squared window) with a winner-takes-all strategy. Next, we apply a left-right consistency check to get rid of unreliable matches, especially in occluded areas, see figure 1. As errors are most likely to occur near occluded and depth-discontinuity areas, a robust correlation measure which is designed to reduce errors in these areas is used. According to our early experiments, the Smooth Median Powered Deviation SMPD measure proposed in [13] leads to good results near occluded areas. The idea is to discard grey-level differences too far from the median value assuming that these high differences are induced by pixels with different depths:

$$\text{SMPD}(\mathbf{p}^l, \mathbf{p}^r) = \sum_{k=0}^{\frac{N}{2}-1} (|\mathbf{p}^l - \mathbf{p}^r - \text{med}(\mathbf{p}^l - \mathbf{p}^r)|^2)_{k:N-1} \quad (1)$$

where \mathbf{p} is a vector containing the N grey-level values of the studied pixel and those of its neighbours ; $k : N - 1$ indicates that the values of the vector inside the parentheses are sorted in ascending order.

3.2 Multi-segmentations

Region-based methods usually use over-segmentation. Indeed, the smallest a region is, the better the plane model fits the real disparities. Since our method relies on an initial disparity map, if we use an over-segmentation, we may not have enough initial disparities in some regions, especially in occluded and discontinuity areas. On the other hand, if we use an under-segmentation, we can

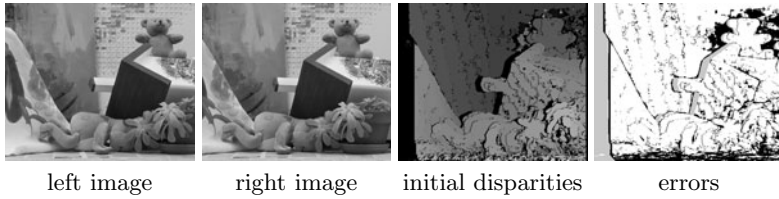


Fig. 1. Left and right images for the pair *teddy*, one example of initial disparity map obtained with SMPD (rejected disparities by the left-right consistency check are shown in black) and error map (errors are shown in black and occluded areas in grey)

find enough initial disparities for a given region but the plane model does not fit the disparities as well as with smaller regions. Therefore, we propose to use different segmentations. This idea is introduced in [5] for B-spline fitting. In our case, under-segmentation is used to estimate disparities in occluded areas hoping that they are inside regions with non-occluded pixels for which initial disparities are computed. Then, finer segmentations are used to have better approximations when possible (mainly in non-occluded and non depth-discontinuity areas).

The mean-shift algorithm is used to perform the segmentations based on color [11]. There are two parameters, the spatial window size h_s and the range size h_r . To set these parameters, we can give the following advices based on our experiments: it seems helpful to select segmentations with different boundaries especially when ambiguities occur (for instance, two neighbour objects with different depth but of the same color) to limit the effects of the segmentation boundary errors by not reproducing the same error too many times on the different segmentations. This can be achieved with the mean-shift algorithm by changing simultaneously the h_r and the h_s parameters. Also it is important to have enough initial matches inside the smallest regions to avoid too many approximation errors for pixels within these regions. An example of different segmentations is given in figure 2.

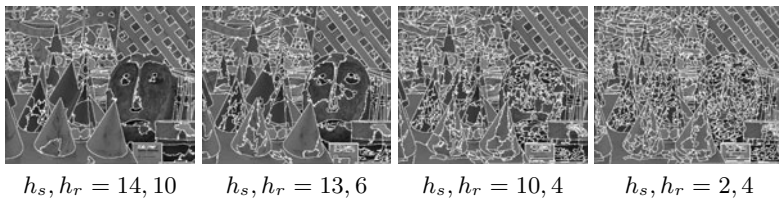


Fig. 2. Example of different segmentations with the left image of the pair *cones*. The boundaries of the regions are shown in white.

3.3 Randomized Voting Scheme

Estimation of the disparity approximations. For each pixel, the proposed algorithm finds different approximations of its disparity value and builds a disparity

density function to determine the most occurring one. These different disparity approximations are given by a plane model computed for the different regions by taking three different initial disparities within these regions. The plane equation is given by: $ai + bj + d = c$ where a, b, c are the parameters and i, j, d are the coordinates of the studied pixel in the disparity space. For each region, we select randomly three points (i, j, d) , named disparity triplet, from the initial disparity map. Then, the parameters are computed by solving a system of three linear equations.

Estimation of the disparity density function. For each region, we consider many possible disparity planes, each plane being given by a different disparity triplet. These planes obtained for each segmentation induce as many disparity approximations for each pixel. Each approximation represents a vote and the final disparity is the most voted one. In order to obtain a sub-pixel estimation, we consider for each pixel the underlying disparity voting distribution. Since the distribution model for the approximations is unknown, we use a kernel density estimation method to compute this disparity voting distribution and we estimate its mode [14]. Thus, a disparity density function \hat{f} based on the n different approximations x_i ($1 \leq i \leq n$) is computed, see figure 4. The value of \hat{f} for $0 \leq x \leq d_{\max}$, where d_{\max} is the maximum disparity value, is given by:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_E(x - x_i) \quad (2)$$

where K_E is the given kernel. According to our early tests, satisfactory results can be obtained with the Epanechnikov kernel:

$$K_E(\mathbf{x}) = \begin{cases} \frac{3}{4}(1 - \|\mathbf{x}\|^2) & \|\mathbf{x}\| \leq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

4 Results

Data set and evaluation. The pairs used in our evaluation are the ones provided for the Middlebury² [1] evaluation: *cones*, *teddy*, *tsukuba* and *venus*. The error rate is the percentage of bad pixels:

$$BP = \frac{1}{N} \sum_{i,j} (|d(i, j) - d_{th}(i, j)| > t) \quad (4)$$

where N is the number of evaluated pixels, d the computed disparity function, d_{th} the theoretical disparity function and t the disparity error threshold. This value is computed over all the pixels (*all*), the non-occluded pixels (*nonocc*) and the depth discontinuity areas (*disc*).

² vision.middlebury.edu/stereo/

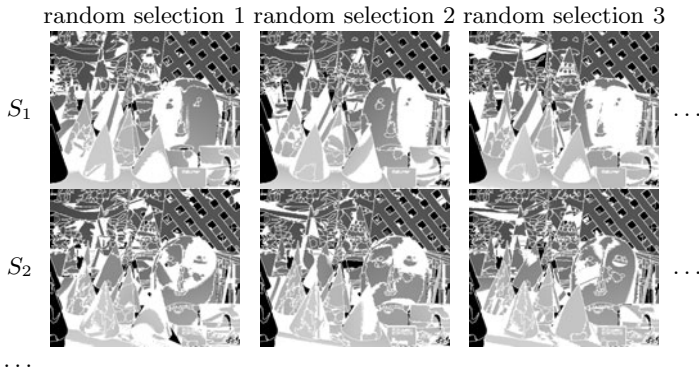


Fig. 3. Disparity maps computed by our algorithm with different random selections and segmentations (S_{1-2}). Region boundaries as well as errors are shown in white. Each map presents different errors. However, these errors are approximations of the true disparities. Let's take a closer look at the cone in front of the mask. The plane model does not fit the whole cone, nevertheless, at each random selection, a different part of the cone obtains correct disparities. As a final result, we take the mode of the disparity density function computed from all the approximations.

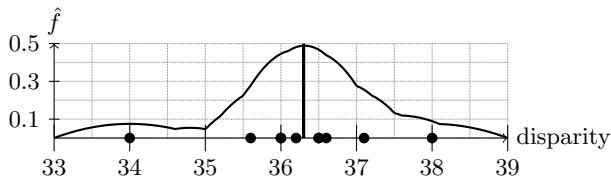


Fig. 4. Density function built from the different estimated disparities (black circles) for a given pixel. The mode (36.3 here) is shown with a bold vertical line.

Error rate vs. number of segmentations. We evaluate the result of our algorithm with different number of segmentations. According to our early experiments, 4 different segmentations seem to be enough to obtain good results. Thus, we use up to 4 different segmentations for this evaluation. We compute the error rate with 1, then 2, then 3 and finally 4 different segmentations. When using only 1 segmentation, early tests showed that the best scores are obtained with the finer segmentation rather than the coarser ones, thus we use this segmentation in our protocol. In the same manner, we determine that when using 2 segmentations, it is better to use the coarser and the finer ones. With 3 segmentations, the best configuration is obtained with the coarser segmentation, the finer segmentation and one between the two others. For each test, the total number of random selections is fixed to 100 and since our algorithm is based on a stochastic process, the experiment is repeated 30 times which is enough to compute a representative average error rate and a standard deviation. The results are computed with different error thresholds t . Since the behaviour is the same for all the t , we present the results for $t > 1$ in figure 5 and table 1. We can see that the use of at

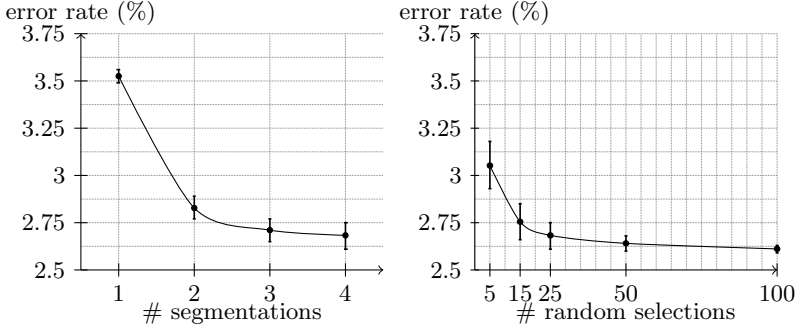


Fig. 5. Error rate for the images *cones* with $t > 1$ over the non-occluded pixels versus the number of segmentations (left) and versus the number of random selections (right). The mean values are given by the black circles and the standard deviation are given by the vertical lines.

least 2 different segmentations significantly improves the results. The best results are obtained with the four segmentations. The only exception is for the pair *tsukuba* which is hard to segment, especially in the background where they are different objects (i.e. with different disparities) whereas they have low intensity differences and low contrast. We can notice that the standard deviation does not decrease much with the different number of segmentations. In fact, it depends on the number of random selections, see § 4. Therefore, if the aimed application requires a low error rate, the choice of at least 2 segmentations is recommended. A better precision can still be achieved with more segmentations.

Table 1. Error rate and standard deviation for the 4 stereo pairs ($t > 1$) versus the number of segmentations (1–4). The total number of random selections is set to 100. The evaluation is performed over the non-occluded (*nonocc*) pixels, all the pixels (*all*) and the pixels within the depth-discontinuity areas (*disc*). The best result of each row is written in bold.

		1	2	3	4
<i>tsukuba</i>	<i>nonocc</i>	9.72 ± 0.47	5.79 ± 0.81	6.76 ± 0.53	6.74 ± 0.34
	<i>all</i>	10.59 ± 0.47	6.46 ± 0.84	7.45 ± 0.52	7.49 ± 0.35
	<i>disc</i>	19.63 ± 0.64	19.83 ± 1.44	18.08 ± 0.96	17.45 ± 0.75
<i>venus</i>	<i>nonocc</i>	0.31 ± 0.03	0.19 ± 0.03	0.20 ± 0.03	0.17 ± 0.02
	<i>all</i>	0.71 ± 0.05	0.53 ± 0.05	0.58 ± 0.06	0.50 ± 0.04
	<i>disc</i>	3.83 ± 0.25	2.59 ± 0.33	2.66 ± 0.39	2.22 ± 0.27
<i>teddy</i>	<i>nonocc</i>	6.84 ± 0.18	6.34 ± 0.23	6.22 ± 0.25	5.95 ± 0.21
	<i>all</i>	11.97 ± 0.36	10.14 ± 0.41	10.24 ± 0.39	10.13 ± 0.28
	<i>disc</i>	17.93 ± 0.38	16.82 ± 0.68	16.00 ± 0.50	15.62 ± 0.42
<i>cones</i>	<i>nonocc</i>	3.53 ± 0.03	2.83 ± 0.06	2.71 ± 0.06	2.68 ± 0.07
	<i>all</i>	10.60 ± 0.09	8.60 ± 0.19	8.37 ± 0.14	8.12 ± 0.12
	<i>disc</i>	9.51 ± 0.08	8.10 ± 0.16	7.79 ± 0.16	7.71 ± 0.18

Table 2. Error rate and standard deviation for the 4 stereo pairs ($t > 1$) versus the number of random selections (5–100). The evaluation is performed over the non-occluded pixels (*nonocc*), all the pixels (*all*) and the pixels within the depth-discontinuity areas (*disc*) with 4 segmentations. The best result of each row is written in bold.

		5	15	25	50	100
<i>tsukuba</i>	<i>nonocc</i>	9.89 ± 0.94	7.33 ± 0.71	6.74 ± 0.34	5.91 ± 0.28	5.47 ± 0.18
	<i>all</i>	10.66 ± 0.93	8.05 ± 0.68	7.49 ± 0.35	6.65 ± 0.27	6.22 ± 0.18
	<i>disc</i>	19.88 ± 1.24	17.58 ± 0.82	17.45 ± 0.75	16.45 ± 0.63	16.22 ± 0.46
<i>venus</i>	<i>nonocc</i>	0.49 ± 0.31	0.20 ± 0.03	0.17 ± 0.02	0.16 ± 0.01	0.15 ± 0.01
	<i>all</i>	0.89 ± 0.34	0.55 ± 0.06	0.50 ± 0.04	0.48 ± 0.02	0.48 ± 0.02
	<i>disc</i>	3.21 ± 0.82	2.56 ± 0.37	2.22 ± 0.27	2.13 ± 0.18	2.09 ± 0.16
<i>teddy</i>	<i>nonocc</i>	7.17 ± 1.03	6.04 ± 0.25	5.95 ± 0.21	5.74 ± 0.15	5.65 ± 0.12
	<i>all</i>	11.56 ± 1.30	10.26 ± 0.38	10.13 ± 0.28	9.77 ± 0.28	9.80 ± 0.26
	<i>disc</i>	17.38 ± 1.20	15.70 ± 0.52	15.62 ± 0.42	15.29 ± 0.39	15.19 ± 0.27
<i>cones</i>	<i>nonocc</i>	3.05 ± 0.12	2.75 ± 0.09	2.68 ± 0.07	2.64 ± 0.04	2.61 ± 0.02
	<i>all</i>	8.88 ± 0.22	8.30 ± 0.18	8.12 ± 0.12	8.02 ± 0.08	7.94 ± 0.04
	<i>disc</i>	8.58 ± 0.29	7.88 ± 0.23	7.71 ± 0.18	7.60 ± 0.10	7.52 ± 0.06

Error rate vs. number of random selections. We apply our algorithm with different numbers of random selections (5, 15, 25, 50 and 100) to find the best solution. As explained in § 4, for each tested number of random selections, we repeat the test 30 times and we compute the mean and the standard deviation given by the error rates, the results are also computed with different error thresholds t but only the results for $t > 1$ are presented, see figure 5 and table 2. We can notice that after 25 random selections, the mean value of the error rate decreases slowly. However, the standard deviation is still high. It tends to decrease with the number of random selections. At 100 random selections, we can see that the standard deviation is low (i.e. the probability to obtain a better solution increases). Therefore, if the aimed application requires a low error rate, the number of random selections must be set between 25 and 100. The higher this number is, the lower the standard deviation is.

Ranking. Among all the results, we take for each pair the one that gives the best results and we submit them to the Middlebury website. According to the Middlebury ranking, see table 3 and figure 6, the proposed method gives very competitive results on the images *teddy* and *cones*. Our method is ranked in the top 5 for these images at sub-pixel level ($t > 0.5$). These images show a lot of

Table 3. Middlebury error rates and ranking into parentheses with different thresholds t : $t_1 = 0.5$, $t_2 = 0.75$ and $t_3 = 1$. The best ranking for each t is shown in bold.

t	<i>tsukuba</i>			<i>venus</i>			<i>teddy</i>			<i>cones</i>		
	<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>
t_1	16(34)	16.7(34)	27.1(73)	2.48(10)	2.95(8)	8.13(8)	8.73(3)	13.9(4)	22.1(3)	4.42(1)	10.2(1)	11.5(2)
t_2	9.83(27)	10.6(26)	23.8(64)	0.39(6)	0.78(9)	3.97(13)	6.41(7)	11.1(6)	17.1(9)	2.95(2)	8.40(4)	8.45(3)
t_3	4.85(77)	5.54(70)	17.7(74)	0.13(6)	0.45(13)	1.86(8)	5.40(12)	9.54(12)	14.8(13)	2.62(4)	7.93 (7)	7.54(6)

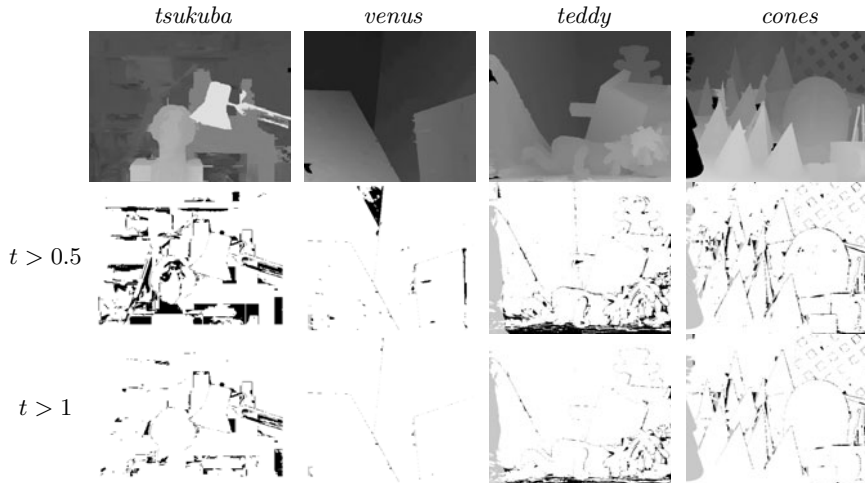


Fig. 6. The first row shows the final disparity maps computed by our algorithm. The second and third rows show the error maps with $t > 0.5$ and $t > 1$ (black = errors, grey = errors in occluded areas).

conic or spheric areas and show the robustness of the method to this kind of areas. Nevertheless, errors are high with the image pair *tsukuba*. In fact, these images show dark objects in the background with different disparities which are difficult to segment. According to our observations, we can say that errors are mainly due to two reasons:

- *Bad segmentations* – We can see for instance on *tsukuba* that the bad segmentations (occurring too many times in the different maps), mixing the top of the camera and the background, introduces errors in the final result.
- *Bad initial disparities* – For instance, we can notice on *teddy* that the errors are mainly located on the floor. These errors come from the fact that almost all the initial disparities are false, for the majority, in this area, see figure 11. This is due to the fact that slanted areas are difficult to match by using a squared-window local correlation algorithm.

Computation time. The total computation time of our method depends on the time of the different steps. The mean-shift algorithm can be time consuming depending on the number of shifts to be performed, another segmentation algorithm can be used instead if time is an issue. The time of the correlation-based matching depends on the complexity of the correlation measure and the constraints employed but this step is easily and highly parallelizable since the processing is the same for each pixel. Finally, according to our experiments, the voting scheme takes about 1 s. to compute one random selection map on a laptop³. However, our code was not optimized but since these random

³ MacBook Intel 2GHz.

selections are independent they can be computed in parallel. Besides, each selection processes regions independently thus parallelism can also be achieved.

5 Conclusion

We proposed a region-based stereo matching algorithm which uses different segmentations and a new method to select the final disparity based on a randomized voting scheme. Our algorithm gives good results at sub-pixel accuracy even with non polyhedral objects. It is easy to implement yet very effective, giving competitive results in the Middlebury evaluation protocol. Nevertheless, we believe that we can still improve our method by using a confidence measure to have a weighted random selection of the disparity triplet to be able to reduce the influence of incorrect disparities.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJVC* 47, 7–42 (2002)
2. Wang, Z.F., Zheng, Z.G.: A region based stereo matching algorithm using cooperative optimization. In: *CVPR* (2008)
3. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: *ICPR*, vol. 3, pp. 15–18 (2006)
4. Yang, Q., Wang, L., Yang, R., Stewénius, H., Nistér, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *PAMI* 31, 492–504 (2009)
5. Bleyer, M., Rother, C., Kohli, P.: Surface stereo with soft segmentation. In: *CVPR* (2010)
6. Sun, J., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: *CVPR*, vol. 2, pp. 399–406 (2005)
7. Taguchi, Y., Wiburn, B., Zitnick, C.L.: Stereo reconstruction with mixed pixels using adaptive over-segmentation. In: *CVPR* (2008)
8. Yang, Q., Engels, C., Akbarzadeh, A.: Near real-time stereo for weakly-textured scenes. In: *BMVC*, vol. 1, pp. 924–931 (2008)
9. Hong, L., Chen, G.: Segment-based stereo matching using graph cuts. In: *CVPR*, vol. 1, pp. 74–81 (2004)
10. Lin, M.H., Tomasi, C.: Surfaces with occlusions from layered stereo. *PAMI* 26, 1073–1078 (2004)
11. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: color image segmentation. *CVPR*, 750–755 (1997)
12. Bleyer, M., Gelautz, M.: A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS* 59, 128–150 (2005)
13. Chambon, S., Crouzil, A.: Dense matching using correlation: new measures that are robust near occlusions. In: *BMVC*, vol. 1, pp. 143–152 (2003)
14. Chen, H., Meer, P.: Robust computer vision through kernel density estimation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 236–250. Springer, Heidelberg (2002)

Adaptive Neighbor Pairing for Smoothed Particle Hydrodynamics

Brandon Pelfrey and Donald House

Clemson University

Abstract. We present a technique for accelerating Smoothed Particle Hydrodynamics (SPH) by adaptively constructing and reusing particle pairing information. Based on a small calculation performed on each particle pair, we can determine whether or not pairing information needs to be recomputed at each simulation frame. We present simulations that show that for numbers of particles above 8,000, total simulation computation time is less than one-third of that of the non-adaptive algorithm. In addition, our simulations demonstrate that our algorithm produces simulation results that are visually very similar to the unoptimized version and that visual error does not accumulate as the simulation progresses.

1 Introduction

Smoothed Particle Hydrodynamics (SPH) has become a popular method, in the computer graphics community, for simulating free surface flow. Being a *Lagrangian* technique, SPH tracks the fluid using a set of particles, each of which represents a chunk of material with a constant mass. SPH has the advantage, over grid-based Eulerian methods, of being able to easily track complex surface features without the use of a level set and allows for complex splashing and surface tension effects.

The structure of the SPH algorithm requires that fluid particles exert forces upon their neighbors at every time step. Because of this, the time complexity of a naive implementation of the algorithm is $O(n^2)$ in the number of particles, which is too poor even for a moderately-sized system. Therefore, the representation of fluid forces in SPH typically uses weighting kernels of compact support, so that it is only necessary for a single particle to consider neighbors lying within the radius of its weighting kernel. Then if local particle density is bounded it is possible to use uniform grids to turn the system into an $O(n)$ algorithm for a fixed density. Unfortunately, in this case, the algorithm becomes $O(d^2)$, where d is the particle density, which is directly proportional to the maximum number of particles in a grid cell. Thus, for SPH simulations, finding neighboring particles, a process which we call neighbor pairing, is often the most compute intensive component of the SPH algorithm [2]. In this paper we present a technique that alleviates the need to compute this information for each particle at each frame and instead computes lists of particle pairs which are then adaptively updated when needed.

2 Previous Work

SPH was developed simultaneously by Lucy [6] and Gingold and Monaghan [3] for astrophysical simulation of interacting star systems and was introduced, as a fluid simulation method, to computer graphics by Desbrun and Gascuel [2]. SPH has since been extended in many ways in order to simulate exotic types of flows and to counteract numerical instabilities. Premroze et. al. [9] introduced the Moving Particle Semi-Implicit (MPS) technique to computer graphics as a mesh free method for computing multiple interacting fluid flows. Müller et al. [7] introduced interactive real-time simulation of SPH on commodity hardware utilizing a variety of weighting functions, each for different force calculations. Plastic and elastic flows were modeled in work by Clavet et al. [1] using an impulse-based solver capable of dynamic splashing behaviors in real-time, and introduced a two-part pressure force scheme based on double-density relaxation with two polynomial weighting functions. Fast solvers based on parallel algorithms for the GPU were introduced by Takahiro et. al. [4] and brought large scale systems to commodity workstations.

Since the traditional SPH codes do not directly enforce fluid incompressibility, the fluids simulated with SPH have a tendency to compress, giving them a springy, unnatural look that is not volume preserving. This is because pressure forces are calculated as a function of local particle density, and thus act like tiny springs responding to the distance between particles. A number of techniques have been developed to reduce these compression artifacts. Solenthaler and Pajarola [10] introduced a predictive-corrective scheme for incompressible flows, iteratively arriving at a divergence-free result. Recently, Fabiano et. al. [8] used the SPH paradigm to construct a meshless form of the Helmholtz-Hodge decomposition used in the grid-based semi-Lagrangian method [11], resulting in a sparse linear system directly solvable for exactly incompressible flow.

While SPH was originally devised for astrophysical simulation, the method presented here is principally motivated by observations of the computational molecular dynamics experiments of Loup Verlet [12], where the dynamics of argon atoms were simulated through particle-particle interactions. In that paper, instead of computing neighboring particles lists each frame, they are computed once, reused many times, and then recomputed on a regular interval. The number of frames to reuse information was chosen experimentally and was selected so that momentum was approximately conserved. In this paper we expand on this idea for the particular case of incompressible fluid simulation and provide an adaptive method to choose which neighboring particle lists should be recomputed.

3 Overview of SPH

Smoothed Particle Hydrodynamics is a Lagrangian, particle-based, formulation of the Navier-Stokes equations

$$\rho(\dot{\mathbf{u}} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot (\mu \nabla \mathbf{u}) + \mathbf{F}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where \mathbf{u} is the velocity field, ρ is the fluid density, p is pressure, μ is the fluid viscosity, and \mathbf{F} is the sum of external or “body” forces exerted on the fluid, accounting for effects such as gravitational acceleration. Equation 1 is the force equation, relating fluid acceleration to the various forces acting on the fluid, and Equation 2 is a statement of incompressibility; i.e. that the fluid velocity field should be divergence free. As a numerical technique, SPH approximates a field locally by a weighted sum of values from nearby particles. Any particular field quantity A , evaluated at position \mathbf{x} is expressed as

$$A(\mathbf{x}) = \sum_i \frac{m_i}{\rho_i} A_i W_h(\|\mathbf{x} - \mathbf{x}_i\|), \quad (3)$$

with A_i , m_i , and ρ_i being the field value, mass, and density at particle i . W_h is typically a smooth radial basis function with finite support, often approximating a Gaussian function. These weighting functions take the distance between two points as an argument, and the subscript h is used to denote the radius of support, thus the summation need be taken over only those particles falling within radius h of position \mathbf{x} . Weighting functions are chosen to maximize accuracy and provide stability in numerical simulations. A discussion of weighting function properties is given by Müller et. al. [7].

The use of weighting functions simplifies the treatment of differential operators, since differentiation in the form of gradients and Laplacians is carried onto the weighting functions via linearity, so that

$$\begin{aligned} \nabla A(\mathbf{x}) &= \nabla \left(\sum_i \frac{m_i}{\rho_i} A_i W_h(\|\mathbf{x} - \mathbf{x}_i\|) \right) \\ &= \sum_i \frac{m_i}{\rho_i} A_i \nabla W_h(\|\mathbf{x} - \mathbf{x}_i\|). \end{aligned}$$

This formulation makes the assumption that density ρ_i is locally constant. This simplifies the computation of forces and provides negligible error if the density varies significantly only at a distance beyond the weighting function’s radius. During simulation, density is needed for essentially every calculation. Using equation 3 we can formulate density at a particle i as

$$\rho_i \equiv \rho(\mathbf{x}_i) = \sum_j m_j W_h(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad (4)$$

where parameters are often chosen to simplify computation, e.g. $m = 1$. Pressure is typically calculated through the equation of state and is approximated by

$$p_i = k(\rho_i - \rho_0),$$

where k is a stiffness constant related to the speed of sound in the fluid and ρ_0 is the rest density of the fluid. The pressure and viscous forces presented by Müller et. al. [7] are given by

$$\mathbf{f}_i^{\text{pressure}} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_h(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad (5)$$

$$\mathbf{f}_i^{\text{viscous}} = -\mu_i \sum_j m_j \frac{\mathbf{v}_j + \mathbf{v}_i}{\rho_j} \nabla^2 W_h(\|\mathbf{x}_i - \mathbf{x}_j\|). \quad (6)$$

With these quantities defined, the typical SPH algorithm proceeds as in Algorithm 1.

Algorithm 1. The standard SPH algorithm

```

while simulating do
  for each particle i do
    | Compute Densities per Eqn. (4)
  end
  for each particle i do
    | Compute  $\mathbf{f}_i^{\text{pressure}}$  per Eqn. (5)
    | Compute  $\mathbf{f}_i^{\text{viscous}}$  per Eqn. (6)
  end
  for each particle i do
    |  $\mathbf{F}_i = \mathbf{f}_i^{\text{pressure}} + \mathbf{f}_i^{\text{viscous}}$ 
    | Integrate( $\mathbf{x}_i, \dot{\mathbf{x}}_i, \mathbf{F}_i$ )
  end
end

```

4 Neighbor List Construction

While the above constitutes the bulk of a typical SPH implementation, without a mechanism to manage the time complexity of computing interparticle interactions, the simulation will scale quadratically with the number of particles. Below we explain our method for mitigating this scaling problem.

Calculating the full n^2 set of inter-particle forces is both inefficient and unnecessary. Since particle pairs that are farther than kernel radius h have weighting function values of 0, they can be discarded. Taking advantage of this requires an extra step in the SPH algorithm, where a list of interacting particle pairs is created. Particles are linked with a position in a spatial data structure such as a uniform grid [74]. Usually, data structures such as KD trees have been employed for these types of search problems, however, KD trees in particular have a super-linear time complexity for radius-based searches. In the computer graphics literature, there have been numerous implementations based on spatial hashing and uniform grids, which have roughly the same performance [74,5]. We present here the method for utilizing a uniform grid.

Letting m be the number of dimensions in the simulation, a particle's position is passed to a function, $\mathbf{I}(\mathbf{x}_i) : \mathbb{R}^m \rightarrow \mathbb{N}_0^m$, which provides the m integer

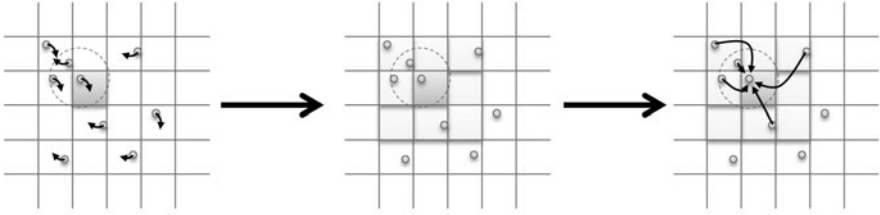


Fig. 1. Using a uniform grid to create particle-interaction lists. (left) All particles are inserted into the grid. (middle) Each particle queries neighboring grid cells and creates a list of neighboring particles. (right) This list is used throughout computation.

coordinates of the grid cell in which the particle lies. If we choose grid cell side lengths to be twice the smoothing radius h we can guarantee that, for any given position, only $2m$ grid cells need to be explored. At each grid cell, a list holds indices to particles whose centers lie inside of that grid cell. The process is shown visually in Figure 1. After this algorithm has been performed, it is no longer necessary to query all of the n^2 pairings. Instead, each particle will have a list of indices to neighboring particles.

5 Adaptive Neighbor List Algorithm

We now present our method for accelerating the algorithm given in the last section, by updating the neighbor lists only when necessary. We note that particle pairs will continue to interact so long as they lie within distance h of each other. Since the SPH algorithm gives us ready access to current particle velocity information, we can make a prediction about the positions of particle pairs in the next frame. For each particle pair, we estimate the inter-particle distance in the next frame by simple Euler integration using position and velocity from the current frame using the update equation

$$\begin{aligned}
 D_{ij}^{t+\Delta t} &= \|\mathbf{x}_i^{t+\Delta t} - \mathbf{x}_j^{t+\Delta t}\| \\
 &\approx \|(\mathbf{x}_i^t + \Delta t\mathbf{u}_i^t) - (\mathbf{x}_j^t + \Delta t\mathbf{u}_j^t)\| \\
 &= \|\mathbf{x}_i - \mathbf{x}_j + \Delta t(\mathbf{u}_i - \mathbf{u}_j)\|
 \end{aligned}
 \tag{7}$$

Thus, in our algorithm, for each current pair of particles (i, j) if the predicted inter-particle distance $D_{ij}^{t+\Delta t}$ is less than h then the pairing information is kept. If a pair violates this condition then we conclude that the two particles will be out of range of each other in the next frame and flag both particles, indicating that their neighbor lists need to be recalculated. The resulting algorithm is shown in algorithm 2.

Algorithm 2. SPH Algorithm using Adaptive Neighbor Lists

```

forall  $f_i = true$ 
Clear Particle Neighbor Lists
while simulating do
  Clear Grid Nodes
  Insert Particles into Grid
  for each particle  $i$  do
    if  $f_i$  or  $frame \% 200 == 0$  then
      Form Particle Neighbor List
       $f_i = false$ 
    end
  end
  for each particle  $i$  do
    Compute Density
    for each neighbor particle  $j$  in  $list_i$  do
      Compute distance  $(D_{ij}^{t+\Delta t})^2$  per Eqn. 7
      if  $(D_{ij}^{t+\Delta t})^2 > h^2$  then
         $f_i = f_j = true$ 
      end
    end
  end
  for each particle  $i$  do
    for each neighbor particle  $j$  in  $list_i$  do
      Compute Forces  $F_{ij}$ 
    end
    Integrate Particle States
  end
end

```

We have found our simulations to be more reliable if we also periodically recompute every particle list using the traditional method. Our experiments have shown that recomputing all pairing information after a period of 200 frames, regardless of predictions made by this algorithm, is adequate. We also explored additional conditions for re-computation, such as recomputing if the number of neighbors was small, but found that this did not visibly enhance performance.

Since the estimated positions at the future time step are based on linear approximations to the true integrated paths, the error introduced by this method is bounded by terms on the order of $O(\Delta t^2 \|\partial_t \mathbf{f}\|)$. For shorter time steps or forces that vary slowly with respect to successive time steps, this method can be expected to give a good estimation to the particle locations in the following time step, and thus will be stable. For this reason, we use spring penalty forces at object faces rather than simply projecting the particles to the surface, since projection would invalidate our linear estimation.

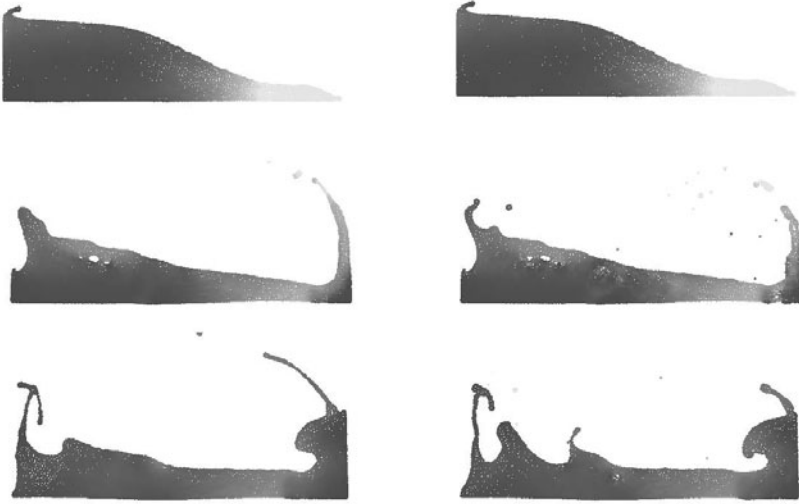


Fig. 2. Three frames taken from identical 2D simulations of a column of water under the force of gravity comparing the visual consistency of standard SPH (left) and our adaptive pairing algorithm (right). Color is used to encode kinetic energy, showing that waves travel similarly through the two simulations.

6 Results and Discussion

We conducted several simulations in two and three dimensions to compare the visual appearance and performance of fluids computed with and without the adaptive neighbor pairing algorithm. Figure 2 shows side-by-side comparisons of three frames from two 2D simulations, one using the standard SPH algorithm, and the other using our approach. Although there are small differences in the fine details, the color coding, which portrays kinetic energy, clearly shows that the overall progression of the two simulations is nearly identical.

Figure 3 gives a more fine-grained demonstration that simulations performed with and without adaptive neighbor pairing maintain excellent visual agreement. This figure shows three frames from a 3D simulation of 4,000 particles, with and without adaptive neighbor pairing. Each particle is individually represented as a ball, whose color coding represents the particle's spatial position at the start of the simulation. It can be readily seen that the positions of individual particles are quite similar, even near the end of the simulation.

Figure 4 compares timing results for a 3D simulation of water being flooded into a cubic tank, with and without our adaptive pairing algorithm. Timing results were obtained with an Intel Core i7 930 with 6 GB DDR3 1600 memory on Ubuntu 10.4 (Linux Kernel 2.6.32). Both simulations use the same uniform grid to accelerate neighbor finding. In this example, the total number of

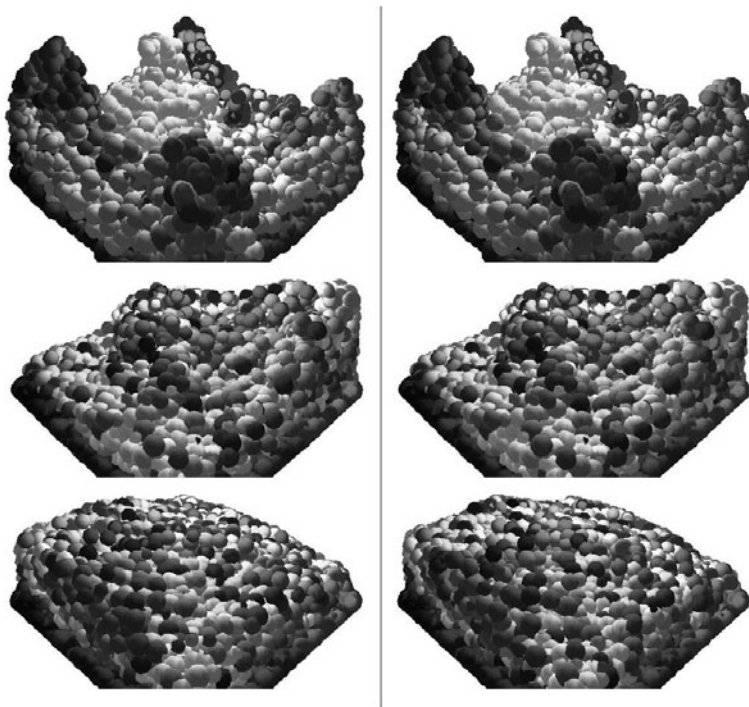


Fig. 3. Three frames from two identical simulations comparing the visual consistency between standard SPH (left), and adaptive pairing (right). Balls representing particles are color coded by their initial position so that differences are obvious. The close similarity between the two animations means that faster simulations can be produced without sacrificing control of the simulation.

particles active in the simulation increases linearly with time, such that the number of particles at the end of the simulation reaches 8,000. These results highlight the fact that for large numbers of particles, adaptive neighbor pairing becomes increasingly attractive, outperforming the standard implementation and decreasing the total time required to simulate a fluid. As can be read from the graph, with a set of only 8,000 particles the average simulation time step is computed in only a third of the time required by the standard SPH algorithm. Thus, we are confident that our method enables SPH simulations to be calculated more efficiently, without sacrificing the control of the fluid animation, at the expense of only minor surface differences.

It is important to note that while this technique is much faster than the standard SPH implementation, it does not lower the asymptotic time complexity of the algorithm since, for stability reasons, we are still required to periodically recompute neighbor lists for all particles.

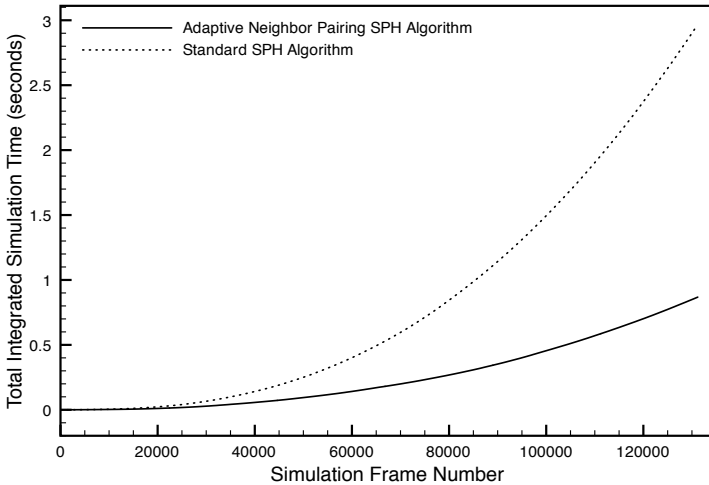


Fig. 4. Integrated simulation time for a simulation where fluid fills a cubic tank. Particles are added to the simulation every 16 frames. The total number of particles at the end of the simulation was 8,000.

We point out that this method can be additionally augmented with information from the scene such as information about collision with rigid bodies or other fluid types. For instance, since fluids interacting with fast rigid bodies or highly deformable models may present a special challenge for this algorithm, it would be possible to flag particles that interact with these objects regardless of other neighboring actions.

Our simulations were intentionally chosen to include high-pressure regions, such as where fluid collides with the floor of a tank. These types of scenarios are typically unstable because of the spring-like formulation of pressure in SPH. In our simulations, we found that the new technique does not introduce any additional instability.

Current research efforts into SPH techniques for GPU see better performance through 3D texture splatting and other techniques that circumvent the need for explicit neighbor-neighbor pairing information. As such, this technique will not be useful for GPU-based techniques.

7 Conclusion

In conclusion, we have presented a new method for speeding up the simulation of SPH fluids, based on maintenance of particle neighbor lists. The algorithm presented can be easily introduced into an existing SPH framework and provides an efficient adaptive method for lowering the computation time of particle-based fluid simulation for computer graphics and animation.

References

1. Clavet, S., Beaudoin, P., Poulin, P.: Particle-based viscoelastic fluid simulation. In: Symposium on Computer Animation 2005, pp. 219–228 (July 2005)
2. Desbrun, M., Gascuel, M.P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In: Computer Animation and Simulation 1996 (Proceedings of EG Workshop on Animation and Simulation), pp. 61–76. Springer, Heidelberg (1996)
3. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics. Theory and application to non-spherical stars 181, 375–389 (1977)
4. Harada, T., Tanaka, M., Koshizuka, S., Kawaguchi, Y.: Real-time particle-based simulation on gpus. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 posters, p. 52. ACM, New York (2007)
5. Krog, O., Elster, A.C.: Fast gpu-based fluid simulations using sph. In: Para 2010 - State of the Art in Scientific and Parallel Computing (2010)
6. Lucy, L.B.: A numerical approach to the testing of the fission hypothesis. *Astronomical Journal* 82, 1013–1024 (1977)
7. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: SCA 2003: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Aire-la-Ville, Switzerland, pp. 154–159. Eurographics Association (2003)
8. Petronetto, F., Paiva, A., Lage, M., Tavares, G., Lopes, H., Lewiner, T.: Meshless helmholtz-hodge decomposition. *IEEE Transactions on Visualization and Computer Graphics* 99(RapidPosts), 338–349 (2009)
9. Premroze, S., Tasdizen, T., Bigler, J., Lefohn, A., Whitaker, R.T.: Particle-based simulation of fluids. *Computer Graphics Forum* 22(3), 401–410 (2003)
10. Solenthaler, B., Pajarola, R.: Predictive-corrective incompressible sph. *ACM Transactions on Graphics, SIGGRAPH* (2009)
11. Stam, J.: Stable fluids. In: SIGGRAPH 1999: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, pp. 121–128. ACM Press/Addison-Wesley Publishing Co. (1999)
12. Verlet, L.: Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.* 159(1), 98 (1967)

System Structures for Efficient Rendering in Virtual Worlds and Virtual Testbeds

Jürgen Rossmann and Nico Hempe

Institute for Man-Machine Interaction
RWTH Aachen University
{rossmann,hempe}@mmi.rwth-aachen.de

Abstract. To date, it is still a challenge to develop a comprehensive simulation system which has the capabilities to serve as a Virtual Testbed. In this paper, we present an efficient system structure, based on a novel 3D simulation database approach, which has been designed according to testbed requirements collect from space robotics to virtual production requirements. The simulation framework VEROSIM[®] which is the basis of our current developments, has been specifically designed to be able to handle a comprehensive class of process simulations in one modular framework. By combining different capabilities - like advanced rendering techniques, rigid body dynamics, terra mechanics, etc. - in a single framework, it is now possible to make the step from virtual worlds to virtual testbeds. This paper focuses on the framework's simulation database, system structure and render engine. It also shows how the render engine extracts a view onto the simulation database in order to work performantly. Furthermore, the simulation system is able to produce high quality real-time renderings of complex scenes on standard consumer PC hardware as well as functional, intuitive graphics on low-end hardware like ruggedized notebooks designed for outdoor usage. Last but not least, examples of different application areas for virtual testbeds are shown and discussed.

1 Introduction

Simulation systems are used in almost any area of research and development. Many systems try to handle a large number of simulation tasks. Each one is developed and specialized for clearly defined goals and applications. In contrast to these specific simulation systems, the goal of the development currently done at our institute in cooperation with our industrial partners is the realization of a comprehensive and performant simulation framework based on an efficient system structure. This system structure is based on a novel 3D simulation database approach, which has been designed according to testbed requirements collect from space robotics to virtual production requirements. The presented simulation framework is able to handle widespread tasks of technical, but also biological or geological processes and present the results in attractive virtual worlds. Furthermore, the flexible and modular system structure allows for the combination

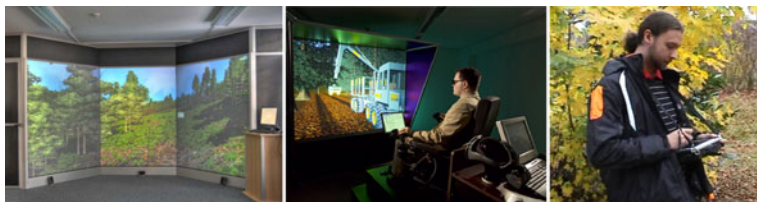


Fig. 1. Left: 3D multi screen visualisation. Middle: Forest machine simulation. Right: Ruggedized notebook for outdoor usage.

of different capabilities - like advanced rendering techniques, rigid body dynamics, terra mechanics, etc. - that can produce such realistic results, that they also can be used as a development basis for engineers. In this case, a virtual world becomes a virtual testbed and real prototypes could become superfluous. In this paper, we will focus on the system and data structure, the render engine and its view onto the simulation database. For the render engine it is a different part to handle different kinds of geometry and furthermore process and render this geometry in a performant and realistic way. Additionally, the render quality should fit to the used hardware configuration. In figure 1 different usecases of our simulation system are shown. On the one hand, additional graphical effects should be applied on high-end systems to further improve the visual quality, on the other hand, the simulation should run in real-time even on low-end hardware like ruggedized notebooks used for outdoor tasks. Essential areas of application and realisations of some different simulation tasks will show the flexibility and quality of the simulation system and its render engine. Finally, we will conclude and discuss the results.

2 Basic Data Structure

The simulation framework is build in an modular way. It is composed of a few core components and many different optional modules that handle additional tasks. In this section, the simulation system's database is introduced. It represents the geometrical information as well as the process-specific information in an object oriented form. The applications shown in section 5 are only a few examples of virtual testbeds currently simulated with our framework. Due to the fact, that a scenegraph-centric view usually being used for VR-Systems is not practical for applications that simulate more than one process in a model, we developed an own object-oriented data structure. The benefits of this data structure on the programmers side, it is possible to keep the model representation in one single data base. Especially for the administration of large models it is a huge benefit to keep them in (distributed) data bases instead of single files. Figure 2 shows the model representation of a simple city environment with a car. Process methods - like RigidBody, Sound, etc. - can be attached to every node for simulation purposes. This is the central view of the database, it contains all

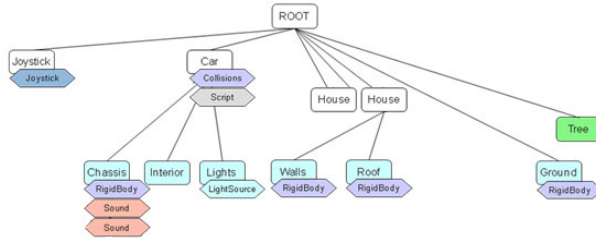


Fig. 2. Database for a city environment with a car. Process methods (like RigidBody, Sound, etc.) can be assigned to every node.

components necessary for the different simulation modules. Because not every module needs to know the whole database and all its components, every module can have its own view of the database. The modules for dynamic simulation, terra mechanics, particles, fire, water and other simulation components have different views of the database, but they all refer to the same single database. Due to the fact, that the render engine is a central part of almost every VR-Simulation System, the following sections will focus on the render engine’s point of view of the model representation. We will show how rendering techniques can benefit from the new structure and how they are used in the simulation framework.

2.1 Node Types

Now we will present an extract of node types available in the simulation system. Beside these node types, the simulation system contains many more for other process simulation purposes. In this paper, we will focus on the rendering structure and the render engine’s view of the model database. For other modules, the basic approaches are similar to the presented ones. Figure 3 shows a simple model with two different boxes. The used node types and their relation are explained now.

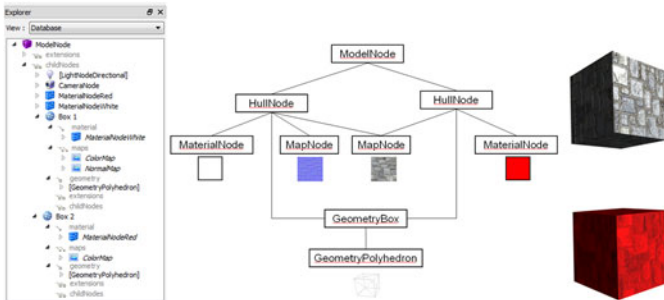


Fig. 3. Left: Explorer view of the database for the simple model. Middle: Illustration of the node types and their connections. Right: The rendered model.

ModelNode: This is the root node of the scene's data tree. It contains global parameters for the scene. A model can only have one ModelNode.

HullNode: Every HullNode contains a matrix that describes the position of the subjacent nodes in the world. In a traverse step, these matrices were accumulated to place the attached objects at the correct position.

MaterialNode: This node type contains material information for objects attached to the corresponding HullNodes. The material properties are essential for lighting calculations.

MapNode: These nodes contain information about textures and an URL to an image file. Every MapNode refers to a single texture but every HullNode can refer to more than one MapNode. This way, more than one texture can be attached to an object. By declaring a map type - like ColorMap, NormalMap, GlossMap, etc. - effects like Normal-Mapping can be applied to the rendered object and improve its visual quality.

GeometryNode: GeometryNodes are the connector between abstract geometry and defined geometry. A GeometryNode contains information that describe a geometry in an abstract way. For example, the GeometryBox contains a parameter 'side length' that describes the size of the box. The GeometryBox itself do not contain any more information about the geometry but it can create a GeometryPolyhedron from it's parameters.

GeometryPolyhedron: In contrast to the GeometryNode, the GeometryPolyhedron contains well defined geometry - like vertices, colors, normals and texture coordinates - which are necessary for rendering purposes.

One of the boxes in the shown scene contains two MapNodes, one with a ColorMap and NormalMap type. Hence, the box is rendered by using the Normal-Mapping technique. The second box only refers to the MapNode with the color map, therefore it is only rendered with a texture and it's corresponding MaterialNode with red material properties. The explorer-view of the scene on the left shows that lights and cameras are also defined as nodes. This way, they can be attached to the desired positions in the tree structure to allow the cameras or lights moving with an desired object. In this example, they are attached to the ModelNode (root).

2.2 Node Extensions

Node extensions can be attached to every node in the data structure (as seen in figure 2). These extensions contain additional information necessary for specific simulation framework modules. There are two extension types: extensions itself and internals. Extensions are saved with the database while internals are created at runtime and deleted after the model has been closed. For the render engine, both extension types are used as well. A ModelNodeExtension is attached to

the ModelNode which contains global rendering parameters like fog, culling, etc., while internals are used for HullNodes, MapNodes and GeometryPolyhedron and contain OpenGL-specific parameters like texture-IDs or Vertex Buffer Objects that become invalid after the model has been closed.

3 Render Engine Basics

The simulation system's render engine is composed of a core component and a various count of render extensions that can be registered at the core component. The rendering process is layer-driven. A defined count of layers - like background, main, overlay, etc. - is rendered sequentially in the given order. Every render extension must register to a specific layer, so the core component can call the render extension at the correct time. In the following subsections, we describe the render pass preparations and the render pass itself. Following, we will show some render extension examples and their application.

3.1 Render Preparations

Before a scene can be rendered, a render-friendly data structure needs to be build for performance reasons. Therefore, the scene is sorted by materials and textures. Furthermor, the scene's geometry needs to be arranged in Vertex Buffer Objects. In figure 4 the used render-friendly data structure is illustrated.

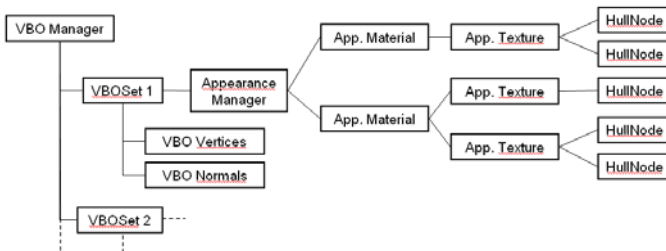


Fig. 4. Data structure for the render pass

Every kind of geometry - vertices, normals, colors and texture coordinates - is arranged in an own Vertex Buffer Object. This way, the simulation system can handle all possible combinations of geometry types, like vertices and normals or vertices, normals and texture coordinates and so on. For every combination, a VBOSet with the corresponding count of Vertex Buffer Objects is defined. All HullNodes found in the data base are traversed by the VBO-Manager and refered to the correct VBOSet according to the geometry types found in the GeometryPolyhedron. The data blocks of the GeometryPolyhedron are written to the Vertex Buffer Objects and a HullNodeInternal is attached, which contains the HullNode's corresponding VBOSet, the VBO start offset and the render

sequence (i.e. 10 quads, 5 triangles, 6 polygons). Subsequently, HullNodes are sorted by materials and textures. The AppearanceManager, which is part of every VBOSet, assigns the HullNodes to AppearanceMaterials and AppearanceTextures according to their referred MaterialNodes and MapNodes. Finally, image files of MapNodes are loaded into textures, the corresponding texture-IDs are written to a MapNodeInternal and attached to the MapNodes. Now, the data structure is ready for rendering.

3.2 Render Pass

In the render pass, the VBOSets are activated in rotation. Firstly, the material parameters are set according to the AppearanceMaterial, secondly, the textures of the corresponding AppearanceTexture are bound. As the render engine is shader-driven, the matching shaders need to be applied before rendering. Depending on the Geometry- and MapNode types - ColorMap, NormalMap, etc. - the matching shader is chosen and activated. If MapNodes with the types 'NormalMap' and 'ColorMap' are found, a Normal-Mapping shader is activated for example. Finally, the ModelViewMatrix of the corresponding HullNode is set and the geometry attached to the activated appearance is rendered. These steps are repeated until every VBOSet and its Appearances are rendered.

4 Render Techniques and Usage

In this section, an abstract of some render techniques currently integrated to the simulation system is presented. Due to the modular database and framework structure, most of these techniques are implemented in single plugins that can be attached as render extensions. We will show how these techniques are applied to the model's database and how they are handled by the render engine.

4.1 Shadows

Shadows are an essential part of virtual worlds. They greatly improve the immersion and virtual worlds become more authentic. Moreover, they greatly help to locate an object in a virtual scene. Especially for virtual testbeds, this is an important fact. Shadow rendering in our simulation system is based on Shadow Mapping as there is no need to modify the geometry. Moreover, it can be applied to almost every type of scene. Activating shadows for any scene in our simulation system is very easy. Only a single extension named ExtensionShadow needs to be applied to the desired light source. This extension also contains all shadow rendering options like shadow penumbra size, map size or offset necessary for shadow rendering and quality adjustments. In the render pass, the render engine automatically performs an additional depth render pass and adjusts the scene shaders to display shadows.

4.2 Skinning

With skinning, an object animated by a skeleton structure can be displayed very realistic [1][2]. The object itself is covered with a flexible 'skin' that is animated with the skeleton. This way, the different object's parts fit to each other with smooth transitions. Usually, there are two different ways to apply skinning. The first is the modification of the geometry in the basic data structure. The drawback of this method is a great loss of performance, because the geometry needs to be updated and copied every frame. The second method uses Vertex Shaders. In this case, the geometry only contains the base pose of the object and stays untouched, the skinning animation is calculated directly on the GPU. This results in a much better performance but every shader of the render engine needs to be modified. To avoid this modification, we use the benefits resulting from the flexible structure of our render engine. Skinning is done by a render extension independent from the core render module which is registered in an layer preliminary to the main render layer. Methods provided by modern graphics hardware are used to update the geometry directly on the GPU using Geometry Shaders and written back to the GPU memory using transform feedback. This results in an excellent performance without modify any scene shaders. The skinning technique can easily be applied by adding a SkinningExtension to a HullNode containing skinning matrices and parameters.

4.3 Particles

Particle systems can be used in many ways to improve the visual appearance of virtual worlds. The particle system of our simulation system is based on point particles and geometry shaders. The geometry shaders add recent vertices to each point transforming them to textured quads or more complex objects. Texture coordinates are calculated and the particles are moved over time. Beside additional per-point input like normals, colors, etc, more information can be applied by using TextureBufferObjects and assigning the particles to specific texels by using their vertex input id. Currently, two sorts of particle systems are implemented. The first one is a simple ParticleEmitterExtension that can be attached to a node of the database. This particle emitter spreads out textured quads in a defined angle and direction. This kind of particle system can be used to visualize welding spark, saw dust, weather effects like rain or other situations that require a large amount of particles [3]. The second particle system is used for shrub visualisation like grass or small vegetation. It mostly works like the first particle approach, but in addition, it is attached to a ground object that contains the ground geometry and a vegetation map. Every color channel of this texture represents a defined sort of vegetation - like grass, flowers, bushes or shrubs - and the color channel's value represents it's density. Using these information, corresponding points are generated at run-time. Depending on the distance to the viewer, single or multiple textured quads are generated by the geometry shader for each input point. This way, it is easy to add shrub to any kind of ground geometry without adding vegetation objects to the scene by hand which is very time-consuming.

4.4 Post-Processing Effects

Render extensions that are registered for layers called after the main layer apply post-processing effects. They use the rendered scene's depth- and colormap and apply additional effects that increase the visual appearance. To prevent a performance loss resulting from copying the frame and depth buffer to a texture, the render engine uses Frame Buffer Objects to render the scene directly into textures. These textures are modified by post-processing effects and finally drawn to the screen. These post processing effects do not need any specific node added to the database as they are globally applied after scene rendering. Currently, we have implemented two post processing effects, Bloom and depth of field. The bloom effect analyzes the scene texture and expands and blurs bright areas. This results in a glow effect that simulates high dynamic range images. Depth of field simulates a camera effect. In reality, a camera can only display a specific focus area very clear. Depending on the camera's focus, very close or far objects appear blurred. To simulate this effect, the whole scene is blurred and merged with the original scene's texture depending on the depth of each single pixel. The bigger the distance between the camera's focus and the pixel's depth, the bigger the percentage of the blurred texture in the result.

5 Applications

In this section some simulation scenarios and virtual testbeds are presented, which currently use the framework. All shown scenarios and screenshots are taken from our multi-purpose simulation framework VEROSIM® and are based on the model representation described in the past sections. The examples will demonstrate the flexibility and expandability of the simulation system. All shown scenarios run on standard PC hardware. Many different output devices like multi screen stereo projection, different 3D displays or head mounted displays are currently supported by the simulation system, which can easily be applied to any simulated scenario.

5.1 Virtual Production

The simulation of virtual factories starts with the modeling of production lines. Transportation elements, sensors, programmable controls and robots were chosen



Fig. 5. Screenshots taken from the Virtual Factory testbed

from a library. After the production line modeling, robots can be programmed using their individual programming language like in a real factory. The whole production process can be simulated in this virtual testbed including worker interaction without the need of real prototypes. Figure 5 shows screenshots of the Virtual Factory. For a better visualisation, shadows, skinning for the workers and particles for welding sparks are simulated.

5.2 Space Robotics

In the field of space robotics virtual testbeds not only deliver additional benefits, they are indispensable. In contrast to other simulation areas it is sometimes impossible to build a real prototype for space robotics. Figure 6 shows screenshots of some of those virtual testbeds. On the left side screenshots of the simulation of the International Space Station ISS are shown. The model is composed of over two million vertices and high resolution textures. In this environment, the presented post-processing effects like Bloom and Depth of Field lead to a very good visual quality. The bloom effect emphasizes the shining characteristics of the ISS's structure with incoming or reflective sunlight and with the depth of field simulation the observers view can be focused to desired areas. On the upper right part of figure 6 you can see a screenshot of the 'SpaceClimber' by the DFKI 4. The SpaceClimber is a small six-legged robot to explore the moon in the near future. A virtual testbed that simulates this robot and the environment is currently under development. It uses dynamic simulation, terra mechanics, etc., for mission planning. The lower right part shows a screenshot of the DEOS project. Here, a satellite should be developed, that can grasp other satellites for repairs or course correction.

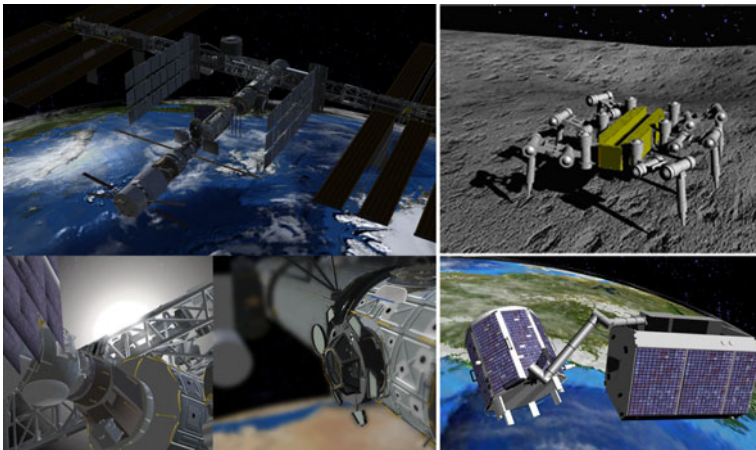


Fig. 6. Left: Screenshots of the virtual International Space Station ISS. Upper right: The 'Space Climber'. Lower right: The DEOS satellite.

5.3 The Virtual Forest

The Virtual Forest is a general term for a collection of virtual testbeds for the forestry industry within one single simulation system [5]. Beside high quality 3D rendering it also includes geo information system (GIS) functionalities, virtual testbeds for work machine simulations as well as remote sensing simulation and database visualisation. The simulation system has to produce high quality renderings for presentation or demonstration reasons on high-end hardware, but also it must be capable to run on low-end hardware like ruggedized notebooks for outdoor usage. On the left side of figure 7 a screenshot from the high quality forest database visualisation is shown. The raw data like ground geometry, tree positions and textures are loaded from the database, while effects like the grass, bloom and weather effects are calculated at runtime. In the middle part, a virtual testbed for work machines is shown, which we not only used for vehicle driver training but also for the simulation of the work machines itself [6]. As mentioned before, the dynamic simulation handles the crane interpretation while another plugins - like earth mechanics - supports the interpretation of the chassis and the training in difficult terrain. On the right part of figure 7 a screenshot taken from a low-end device like a ruggedized notebook is illustrated. It uses exactly the same database as the high-quality screenshot on the left, most rendering effects are disabled. Nevertheless, the information important for the user are identically.



Fig. 7. Left: Screenshot of the virtual forest. Middle: Forest machine simulation. Right: Low-end rendering as used on ruggedized devices.

6 Conclusion

New applications for virtual testbeds spur the demand for high performance VR-Systems. The desired ability to simulate different processes within a single framework and within the same model representation requires new software structures to provide a systematical implementation. The structures presented in this paper are flexible, expandable and match the requirements, as depicted by the examples of different application areas. Furthermore, the system is optimized to run on standard PC hardware. Besides the integration of a physically correct dynamics simulation the visualization subsystem, based on a performant

render engine, is a key component of such systems. The object-oriented modelling structure is realized as a simulation database, which serves as the basic information source for all process simulation, as well as for the rendering components. Thus the VR systems structure is no longer 'built around' a scene graph, but the scene graph is derived as just one 'view' onto the database. As depicted by the presented applications, the modular render framework then produces high quality results in a very performant manner, but can also handle low-end devices. The underlying structure furthermore supports the easy adding of new rendering techniques and advanced methods made available through new graphics hardware - and still provides further ideas for future projects.

References

1. Kry, P., James, D., Pai, D.: Real time large deformation character skinning in hardware. In: ACM Siggraph Symposium on Computer Animation, pp. 153–160 (2001)
2. Akenine-Moeller, T., Haines, E., Hoffman, N.: Real-Time Rendering, 3rd edn. A K Peters Ltd., Wellesley (2008)
3. Hempe, N., Rossmann, J.: A particle based real-time rain simulation on the gpu for generic scenes. In: Proceedings of the 11th IASTED International Conference Computer Graphics and Imaging CGIM (2010)
4. Yoo, Y.H., Ahmed, M., Roemmermann, M., Kirchner, F.: A simulation-based design of extraterrestrial six-legged robot system. In: Proceedings of the 35th Annual Conference of the IEEE Industrial Electronics Society IECON 2009 (2009)
5. Rossmann, J., Schluse, M., Schlette, C.: The virtual forest: Robotics and simulation technology as the basis for new approaches to the biological and the technical production in the forest. In: Proceedings of the 13th World Multi-Conference Conference on Systems, Cybernetics and Informatics, WMSCI (2009)
6. Rossmann, J., Schluse, M., Jung, T., Rast, M.: Close to reality simulation of bulk solids using a kind of 3d cellular automaton. In: Proceedings of the ASME 2009 International Design Engineering Technical Conferences Computers and Information in Engineering Conference IDETC/CIE (2009)
7. Wright, R., Sweet, M.: OpenGL Super Bible. Comprehensive Tutorial and Reference, 4th edn. Addison-Wesley Professional, Reading (2007)
8. Rost, R.: OpenGL Shading Language, 3rd edn. Addison-Wesley Professional, Reading (2009)
9. Scott, J.: Game Programming Gems, vol. 7. Course Technology CENGAGE Learning, Florence (2008)

Prismfields: A Framework for Interactive Modeling of Three Dimensional Caves

Matt Boggus and Roger Crawfis

The Ohio State University

boggus@cse.ohio-state.edu, crawfis@cse.ohio-state.edu

Abstract. A framework for interactive modeling of three dimensional caves is presented. It is based on a new spatial data structure that extends existing terrain rendering methods. The cave is represented as a set of slabs, each encoding a portion of the model along an axis. We describe methods for a user to modify this model locally and procedural methods for global alteration. We wish to allow cave modeling as easily as existing terrain editing programs that restrict the model to a single two dimensional manifold. In this paper, we discuss existing cave visualization programs, including their limitations, as well as how terrain editing and rendering methods can be used in the process of modeling caves.

1 Introduction

Caves have long been a staple environment in fantasy and science fiction settings. This trend has continued in many popular video game series such as Halo, Metroid, and Fallout. As the popularity of open world games rises, so does the demand for expansive virtual environments. To ease this cost, many tools have been developed to procedurally create terrains and allow local editing by artists. The same is not true for caves. The goal of this paper is to establish a framework to allow modeling of caves at an interactive rate. This includes the data structure used to represent the cave, methods to interact with the data structure, and rendering methods to display it. The rest of the paper is organized as follows. In the next section, existing approaches for cave modeling and visualization are discussed. Sections 3 and 4 introduce our framework and describe an implementation of it. This is followed by results in Section 5. We end with directions for future research.

2 Related Work

The majority of caves on Earth are solution caves, formed by acidic water dissolving and removing soluble material over time [4]. Caves that are created by the same process have structural similarities. Regions, or passages, within a solution cave fall into three categories. Phreatic passages are horizontal and tube-like. They form in regions where the cave is fully saturated with water and dissolution occurs roughly evenly. Uneven dissolution results in vadose passages that have side walls shaped like those of a canyon. Combination passages have both phreatic and vadose characteristics.

There has been some work on simulating erosion of terrain [1] and [7], but not in the context of modeling caves.

Datasets for caves have yet to be standardized and survey data is often at very low resolution. Still, there are several visualization programs that display cave survey data including WinKarst [11], The Survex Project [8], and Therion [9]. These programs are limited to creating 3D line drawings and coarse polygonal meshes. These models are intended to visualize the structure of the cave, typically for the purpose of map-making. The programs are not designed to edit or create new passages in the cave, they simply visualize cave data. Recent work has shown the benefit of these types of visualizations within a virtual reality CAVE [6], but these models were still static during runtime.

More recently, procedural methods of creating three dimensional cave models have been introduced. Boggus and Crawfis introduce a method of creating solution caves based on approximating water transport, but the models are limited to two planar surfaces: the cave's floor and ceiling [2]. Our proposed framework supports caves with any geometry. Peytavie describes procedural editing operations on run length encoded terrain, including cave-like models [5]. However, no local editing operations are provided. In contrast, our framework allows direct manipulation of individual vertices in the cave and has an efficient explicit surface representation for rendering. Procedural heightmap synthesis and editing algorithms can also be incorporated in the proposed framework. Lastly, the fine details found on the walls of solution cave passages, speleothems [10] and cave scallops [4], are more easily incorporated using a model with an explicit surface representation.

3 Our Approach

In this paper, we ignore conservation of mass. Material can be created or destroyed spontaneously. We refer to adding mass as constructive and removing it as destructive. The editing environment consists of the model and another geometric object which we call an editing tool. The shape of the object is used to select a region in three dimensional space in which we would like to add or remove material. We use tools with convex shapes in order to simplify the task of finding the lowest and highest points on the tool at a given location (x, z) . Tools performing editing operations are referred to as active. Before we introduce our framework, we discuss heightfields, a data structure used in terrain rendering, upon which we base our approach.

3.1 Deformable Heightfields

A heightfield is a two dimensional grid (x, z) containing points that store a single height value (y) . A vertex in a deformable heightfield has only two operations. Increasing its value corresponds to construction of terrain whereas decreasing its value results in destruction of terrain. When a heightfield vertex is contained within an active destructive tool, its height value is lowered to the lowest y value of the tool at (x, z) . Conversely, when a heightfield vertex is contained within an active constructive tool, its height value is raised to the greatest y value of the tool at (x, z) . An example of a deformable heightfield is shown in Figure 1. The editing tool in (b) contains three vertices, which are raised by a constructive tool in (c) and lowered by a destructive tool in (d).

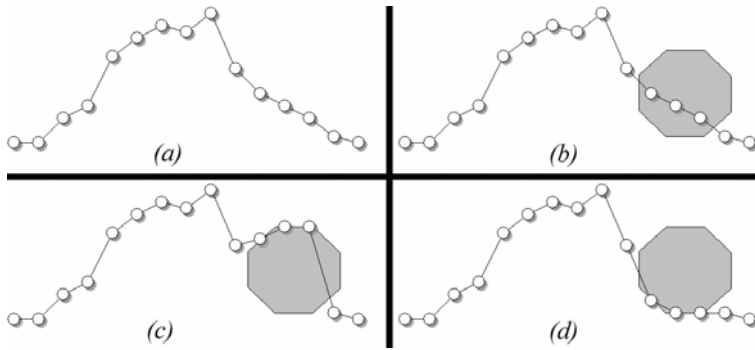


Fig. 1. A heightfield (a) and an editing tool (b) are shown. The three points within the boundary of the tool are adjusted by a constructive tool (c) and a destructive tool (d).

3.2 Prismfields and Vertical Drilling

Since solution caves form by removal of material over time, our main focus is the use of a destructive tool to tunnel or drill through material. In order to support modeling of caves we need a data structure that represents solids rather than surfaces. However, we would like to retain an explicit surface representation so that we can easily distribute speleothems like stalagmite and stalactites and clamp them to the floors and ceilings of cave passages. Since heightfields represent the surface between ground and air, extruding them downwards forms a corresponding three dimensional solid shape. Instead of a grid of triangle or quadrilaterals, we now have a grid of prisms. We call this data structure a prismfield. A simplified example of a prismfield is given in Figure 2. We refer to the original surface as the top or upper surface and the surface at the end of the downward extrusion as the lower or bottom surface.

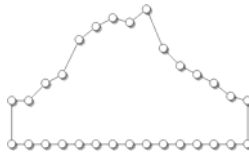


Fig. 2. An example of a prismfield is shown, with interior edges omitted. The result of deformation actions on it are depicted in Figures 3 and 4.

Updating a deformable prismfield is similar to updating a heightfield. We can determine all vertices within the interior of the editing tool. Vertices in the top surface are updated exactly the same as deformable heightfields. The logic of the bottom surface is reversed. A constructive tool should further extrude the object (move vertices down) whereas a destructive tool should lessen the extrusion (move vertices up). In some cases, vertices on the bottom surface may rise above the ones on the top surface, which is invalid. When this occurs the material should no longer exist at this particular grid point. Each vertex has an associated flag that can be used to mark it as inactive. This allows vertical drilling through a prismfield as shown in Figure 3. A

destructive tool removes a large portion of material, leaving a hole in the middle where a vertex became inactive. Since a single prismfield stores only two height values, horizontal drilling is not possible. To incorporate this effect, we introduce a refinement to our framework in the next section.

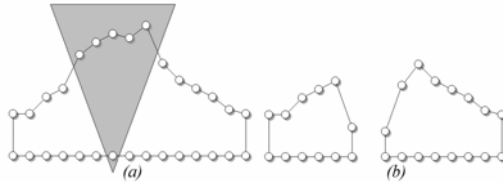


Fig. 3. An example of vertical tunneling is shown. Before updating (a), the tool contains five vertices from the top surface and one from the bottom. Afterwards (b), four of the top vertices are lowered and one becomes inactive.

3.3 Prismfield Framework and Horizontal Drilling

The solution to allowing horizontal tunneling using prismfields is to split one into two, as shown in Figure 4. After the initial update of prismfield vertices for vertical drilling, we check each prism edge against the destructive tool. If the tool contains an edge but neither of its endpoints, we split the prismfield. We label these edges as splitting. Each corresponds to a grid point in the prismfield. Logically, splitting will occur when part of the tool is completely contained within the prismfield. Consider Figure 4, three edges must be split since the tool contains three prism edges but not their endpoints. The result is two prismfields occupying the same space as the old one without the region where the tool was.

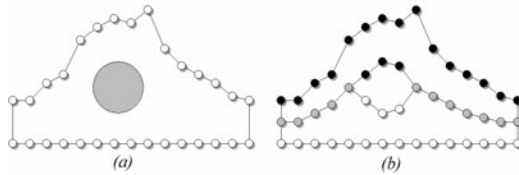


Fig. 4. An example of horizontal tunneling is shown. Before updating (a), the tool is occupying the space between three upper and lower vertices. After updating (b), the prismfield has been split into two where the black vertices belong to the first prismfield, white belong to the second prismfield, and gray vertices are white and black vertices overlapping.

When a prismfield is split into two, one prismfield retains the old bottom surface while the other stores the old upper surface. The vertices for the other surfaces are placed according to the splitting property previously described. The regions corresponding to splitting edges should not contain material as they have been affected by the editing tool. The vertices for these edges are placed at the top and bottom of the tool. Other edges should still represent solid material as they have not been affected

by the tool, so they need to be placed at the same location to avoid creating a gap between the two prismfields. Further horizontal digging can require additional splitting. Thus the final framework is an array of prismfields, each encoding a region, or level, of the cave higher than the previous one. This complicates the first update step since consecutive prismfields can intersect. To handle this, we clamp height values when an upper surface vertex reaches the lower surface of the prismfield above it and in the case where a lower surface vertex it reaches the upper surface of the prismfield beneath it.

4 Implementation

We implemented the prismfield framework and editing operations using C++ and OpenGL. For comparison, we also implemented a comparable application that uses binary voxels. The voxel resolution (64x64x64) was the same as the prismfield resolution on the x and z axes (64x64). Conversion from prismfields and voxels is done by sampling at regular intervals, activating voxels whose centers are contained within a prismfield. All performance tests were done on a PC with an Intel Pentium 4 CPU clocked at 3.20GHz, with 3GB RAM, using an Nvidia GeForce 6800GT.

View independent rendering a prismfield of resolution $n \times n$ requires drawing $O(n^2)$ prisms (each with 6 faces: top, bottom, and four sides), as some prisms may contain inactive vertices and need not be drawn. However, this method introduces redundant work. Only the surface, or boundary faces, between solid material and air must be rendered. In addition to the brute force rendering method we also implemented an optimized approach that only rendered boundary faces of prisms. Rendering of voxels is analogous to prisms, brute force draws $O(n^3)$ voxels (also with 6 faces) but neighboring voxels share faces, so rendering can be optimized by only drawing voxel faces on the boundary between solid material and air. We refer to non-boundary polygons and faces as interior, as they are logically contained within solid material.

A face of an active binary voxel is interior if and only if the voxel neighboring that side is active. Likewise for a prismfield, the side of an active prism is interior if and only if the adjacent prism is active. For binary voxels this is a single Boolean condition. For prisms the status of two nearby vertices must be checked. Figure 5 shows example cases when rendering a prism, labeled pcenter, using quads. One inactive vertex eliminates an entire prism, so to determine if a side face of pcenter is interior, both of the other vertices in the corresponding adjacent prism must be checked. If both are active, the adjacent prism is active, which makes the face the prisms share interior.

Prismfields are capable of representing multiple types of cave passages. Phreatic cave passages, horizontal and tube-like can be accurately modeled using two prismfields, one above and one below the passage. Vadose cave passages, more vertical and shaped like canyons, could be represented by specifying more geometric detail on the side of prisms. Alternatively, vadose passages could be modeled by rotating the prismfields so that splitting occurs for drilling in the vertical direction rather than in the horizontal.

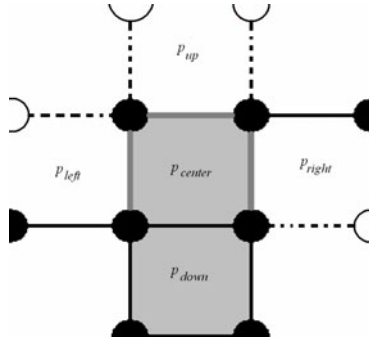


Fig. 5. The top view of rendering a prism pcenter is shown. Active vertices are black, inactive vertices are white, and sides of pcenter that must be rendered (i.e. not interior) are marked with dark gray lines. The prisms pup, pleft, and pright are not present due to at least one vertex being inactive, so the sides pcenter shares with these prisms are exterior, indicating the boundary between solid material and space. The side shared between prisms pdown and pcenter is interior since it is between two solid prisms.

5 Results

Four data sets were created to compare performance between the prismfield framework and voxels. Screenshots of the models are shown in Appendix A. The first (a) was a heightfield formed by smoothing white noise values. The second (b) was made by scaling and reflecting a Digital Terrain Elevation Data file. The third (c) was a cave system composed of three horizontal passages stacked vertically, created using the editing operations described in this paper. The last (d) was created by starting with a solid cube, and iteratively removing material according to a randomly placed sphere within the cube. Thirteen prismfields were present in the final result. Frame rates during rendering of the datasets with and without interior polygon removal as described in Section 4 are given in Table 1.

Table 1. Average performance (in frames per second) for rendering

	a	b	c	d
prismfields (brute force)	37.52	19.26	10.22	9.71
prismfields (interior removed)	58.24	29.88	15.7	13.33
voxels (brute force)	0.48	0.19	0.16	0.62
voxels (interior removed)	15	4.68	4.23	6.26

In these test cases, prismfields outperform voxels. The speed of the prismfield framework is inversely proportional to the number of prismfields needed to represent the environment. The frame rate decreases with the addition of each new prismfield, from one (a) to two (b) to four (c). However, when substantial gaps are present the trend does not continue, as data set (d) used thirteen prismfields yet had performance comparable to (c).

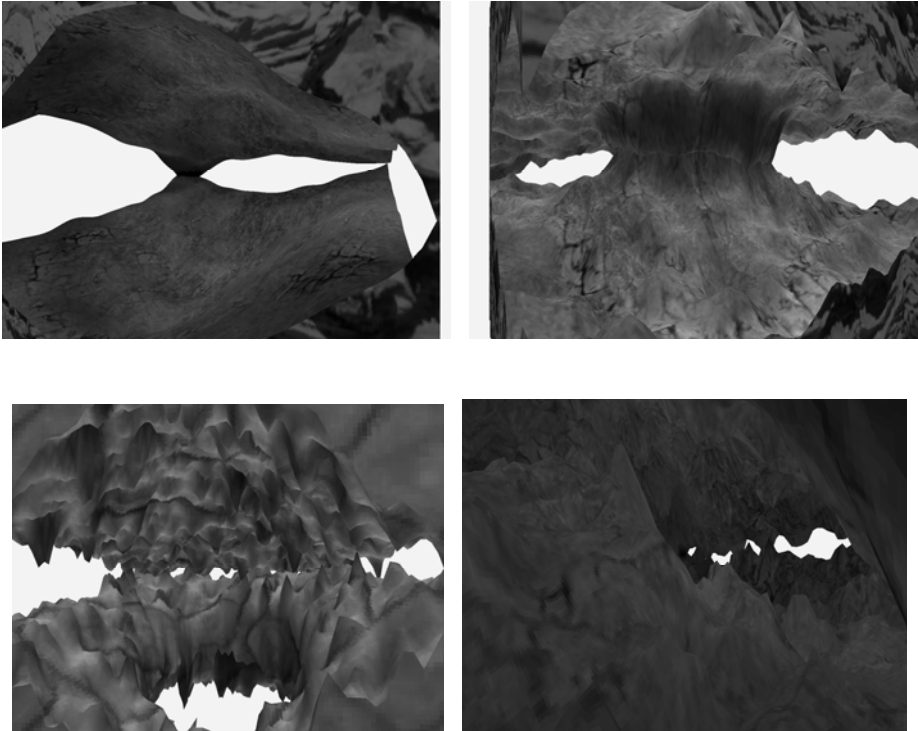
Table 2. Average performance (in frames per second) for rendering a single prismfield at different resolutions

	64 x 64	128 x 128	256 x 256
prismfield (brute force)	37.52	9.89	2.33
prismfield (interior removed)	58.24	15.13	5.49

We performed additional testing to determine the scaling capabilities of prismfields. A single prismfield was rendered at various resolutions. The results are shown in Table 2. Logically, rendering a prismfield is similar to rendering a connected pair of heightfields, so its performance at higher resolutions scales better than voxels, but not as well as heightfields.

6 Conclusion

In this paper we have introduced a framework for modeling and editing of three dimensional caves. The proposed framework consists of a new data structure, the prismfield, and operations to update and maintain an array of prismfields that encodes a three dimensional model. A set of prismfields can represent more varied geometry

**Fig. 6.** Four examples of caves modeled using prismfields are shown

than a heightfield but maintains an explicit surface representation unlike voxels. Since prismfields are based on heightfields, procedural methods to generate and edit terrain can also be used with the new framework in addition to the local editing operations described in this paper. Figure 6 shows several examples of cave environments represented with prismfields. The CPU based implementation of the prismfield framework meets frame rate requirements for an interactive application, making it a viable representation for three dimensional caves. Cave scallops can be included by using bump displacement mapping on the prismfield surfaces. Speleothems can be procedurally placed according to which prisms are active, then clamped to surface of those prisms.

In future work, we plan to explore the use of existing heightfield level of detail algorithms in order to work with higher resolution prismfields. We are also investigating the possibility of adapting specialized heightfield rendering techniques to prismfields. Further optimization of prismfield editing is another topic open to further study. In order to integrate a prismfield framework into computer games, common problems like navigation, path-finding, and procedural generation should also be addressed. Lastly, it may be possible to use heuristics to treat run length encoded terrain as prismfields, which would eliminate the need to store active flags and height values for inactive points.

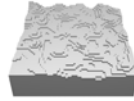
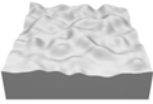
References

1. Benes, B., Forsbach, R.: Layered Data Representation for Visual Simulation of Terrain Erosion. In: Proc. Spring Conference on Computer Graphics, pp. 80–86 (2001)
2. Boggus, M., Crawfis, R.: Procedural Creation of 3D Solution Cave Models. In: Proc. Modelling and Simulation, pp. 180–186 (2009)
3. Houston, B., Nielsen, M.B., Batty, C., Nilsson, O., Museth, K.: Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Trans. Graph.* 25(1), 151–175 (2006)
4. Palmer, A.N.: *Cave Geology*. Cave Books (2007)
5. Peytavie, A., Galin, E., Grosjean, J., Merillou, S.: Arches: a Framework for Modeling Complex Terrains. *Computer Graphics Forum* 28(2), 457–467 (2009)
6. Schuchardt, P., Bowman, D.A.: The benefits of immersion for spatial understanding of complex underground cave systems. In: Proc. Symposium on Virtual Reality Software and Technology, pp. 121–124 (2007)
7. Stava, O., Krivanek, J., Benes, B., Brisbin, M.: Interactive Terrain Modeling Using Hydraulic Erosion. In: Proc. Symposium on Computer Animation, pp. 201–210 (2008)
8. The Survex Project, survex.com
9. Therion, therion.speleo.sk
10. Tortelli, D.M., Walter, M.: Modeling and Rendering the Growth of Speleothems in Real-time. In: Proc. International Conference on Computer Graphics Theory and Applications, pp. 27–35 (2009)
11. WinKarst, <http://www.resurgentsoftware.com/winkarst.html>

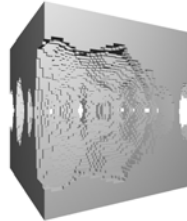
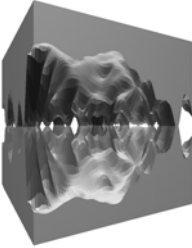
Appendix A: Screenshots of the Four Test Data Sets

Prismfields

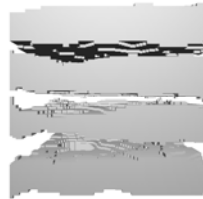
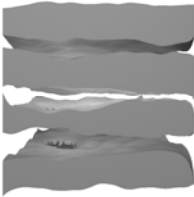
Voxels



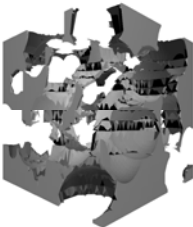
Dataset (a)



Dataset (b)



Dataset (c)



Dataset (d)

Efficient Marker Matching Using Pair-Wise Constraints in Physical Therapy

Gregory Johnson, Nianhua Xie, Jill Slaboda, Y. Justin Shi,
Emily Keshner, and Haibin Ling

Temple University, Philadelphia, USA

Abstract. In this paper, we report a robust, efficient, and automatic method for matching infrared tracked markers for human motion analysis in computer-aided physical therapy applications. The challenges of this task stem from non-rigid marker motion, occlusion, and timing requirements. To overcome these difficulties, we use pair-wise distance constraints for marker identification. To meet the timing requirements, we first reduce the candidate marker labels by proximity constraints before enforcing the pair-wise constraints. Experiments with 38 real motion sequences, our method has shown superior accuracy and significant speedup over a semi-automatic proprietary method and the Iterative Closest Point (ICP) approach.

1 Introduction

Infrared tracked marker analysis is widely used for human motion analysis in computer-aided physical therapy [8] and related applications. In his paper, Klaus Dorfmueller-Uhaas uses a Kalman Filter to perform optical motion tracking [4]. Alexander Hornung discusses a method that automatically estimates all parameters on the fly [6]. Kazuutaka Kurihaha proposed an optical motion capture system with pan-tilt camera tracking, which expanded capturing range [9]. Victor B. Zordan used a physical model to map optical motion capture data to corresponding skeletal motion [15]. L. Herda performed studies for capturing skeletal motion as well [5]. Greg Welch et. al. introduced the Hi-Ball Tracking System [13]. This enabled Virtual Reality applications by generating over 200,000 head-pose estimates per second with very little noise and latency. Hirokazu Kato studied marker analysis for an application in augmented reality conferencing [7].

The key challenges are fast marker detection and fast processing. In this paper we first focus on marker identification problem, which is the first step of further marker sequence analysis. One way to model the marker identification problem is through marker tracking. A large amount of research effort has been conducted on this topic [14] in computer vision. The motion of all markers as a set is non-rigid since it articulates human motion. While rigid transformation approximates well for very small human movement, such as swaying, it is not suitable for the motion patterns in our task. Therefore, the non-rigidity brings difficulties to many marker tracking approaches such as the Kalman filter [2,4]. Another solution is to track each marker independently and use essentially the local proximity to decide marker correspondences. This works fine when marker motions are small and reliable, but this condition is often violated in our study. More relevant technologies can be found in [14], such as [3,20,16].

Our goal is to provide a reliable and efficient solution to the marker sequence tracking/identification problem. Given the labeling of an initial frame, we render the problem as a point set corresponding problem and apply pair-wise distance constraints for markers from rigid parts of human body. We further improve the speed by restricting search of three nearest neighbors when forming candidate label sets. The proposed approach was tested on 38 sequences in comparison with the current semi-automatic proprietary system and the Iterative Closest Point (ICP) approach [1,22]. The results show that our method not only significantly improves previous used semi-automatic system, but also runs much faster. Our method requires no manual labor except the labeling of the initial framework, which largely reduces the tedious human work. We also show the by showing our method provides more accurate results than an application of the ICP method.

2 Problem Formulation

We define a marker motion sequence as $S = \{I_t\}_{t=1}^n$, which contains n frames of marker positions. In a reference frame I_r , there are N identified marker positions and corresponding labels $l_1 = \{1, \dots, N\}$ that are either manually or automatically labeled. In the rest of frames, markers are all un-identified. Let the t -th frame be $I_t = \{p_{t,i}\}_{i=1}^{N_t}$, $t \neq 1$, where N_t is the number of markers in I_t , $p_{t,i}$ is the position of the i -th marker. The identification task is to find a mapping $\pi : \{1, \dots, N_t\} \rightarrow \{0, 1, \dots, N\}$, such that $p_{t,i}$ and $p_{1,\pi(i)}$ are from the same marker if $\pi(i) > 0$. Our task has the following challenges:

- **Missing markers.** There are often some markers missing due to occlusion or system errors.
- **Ghost markers.** Spurious markers sometimes appear, which do not correspond to any marker labels. These markers are called *ghost markers* due to their unpredicted spatial and time appearance. The ghost markers are caused by signal detection errors and the dropping of markers during human motion.
- **Efficiency.** Currently, using a propriety semi-automatic procedure with a commercial system, it takes a post doctoral researcher several hours to annotate a two-minute captured marker sequence.

We now use a toy example (shown in Fig. 1) to illustrate the matching process and our solution. We let $P = \{p_1, p_2, \dots, p_N\}$ be the positions of N identified makers at time $t - 1$ and let $Q = \{q_1, q_2, \dots, q_N\}$ be the positions of N un-identified makers at time t . Examples with $N = 7$ are shown in Fig. 1(a) and (b).

Many previous systems used in physical therapy study build the mapping $\pi(\cdot)$ through nearest neighbor matching. In other words, the solution intends to reduce the following cost:

$$C(\pi) = \sum_{i=1}^{N_t} |p_{t,i} - p_{r,\pi(i)}|^2. \quad (1)$$

Such a solution, while efficient, is problematic with the presence of missing and ghost markers. We developed a second order cost method to improve the matching.

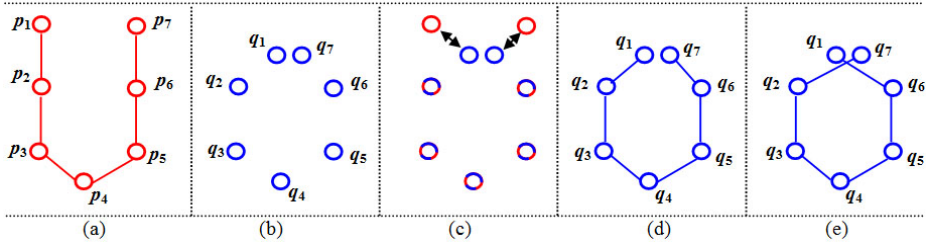


Fig. 1. Pair-wise distance constraints for marker identification. (a) An identified marker set P at frame $t-1$. The links between points show the skeleton of two arms viewed from the top of the skeleton, such that p_1 to p_3 are markers on the left arm and where p_5 to p_7 are markers on the right arm. (b) The marker set Q at frame t , which is to be matched to P . (c) Matching results without consider pair-wise distance. (d) Resulted skeleton from (c). (e) Resulted skeleton from the proposed matching.

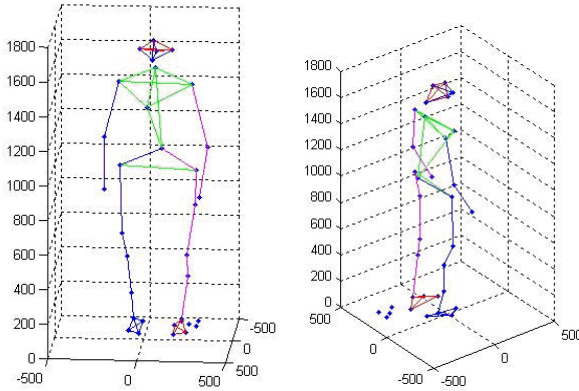


Fig. 2. An example frame with manual annotation. Each line represents a pair-wise constraint.

Intuitively, we can include in \mathbb{I} the deformation constraints from all pairs of markers. Such a direct solution is very expensive and practically un-necessary. It is natural to restrict the constraints in selected pairs of markers that reflect human motion structures, e.g., the links between a hand and an elbow, but not the link between the head and an ankle. Denote $E = \{(i, j)\}$ as the set of such links, we now extend \mathbb{I} as following:

$$C(\pi) = \sum_{i=1}^{N_t} |p_{t,i} - p_{r,\pi(i)}|^2 + \lambda \sum_{(i,j) \in E} c(p_{t,i}, p_{t,j}; p_{r,\pi(i)}, p_{r,\pi(j)}), \quad (2)$$

where λ is the regularization weight, $c(\cdot)$ is the cost of matching segment $(p_{t,i}, p_{t,j})$ to $(p_{r,\pi(i)}, p_{r,\pi(j)})$. Intuitively, the first term on the right hand side models “proximity constraints” and the second term models “geometric constraints”, which is defined by

the deformation between pairs of markers. A natural selection of $c(\cdot)$ is the absolute difference between Euclidean distances $|p_{t,i}p_{t,j}|$ and $|p_{r,\pi(i)}p_{r,\pi(j)}|$, that is:

$$c(p_{t,i}, p_{t,j}; p_{r,\pi(i)}, p_{r,\pi(j)}) = \left| |p_{t,i}p_{t,j}| - |p_{r,\pi(i)}p_{r,\pi(j)}| \right|. \quad (3)$$

For efficiency, we pre-compute the pairwise distance matrix D from the reference frame as $D_{i,j} = |p_{r,i} - p_{r,j}|$. An example reference frame with annotation is shown in Fig. 2.

Directly minimizing the cost function is very expensive. Instead, we turn to find a heuristic solution by taking into account first the proximity constraints and then the geometric constraints.

3 Algorithm

Our task is to find the matching $\pi(\cdot)$ from current frame I_t to the reference frame I_r . Motivated by the above discussion, we propose a two-stage solution for finding the matching $\pi(\cdot)$, followed a step to update the reference frame p_r .

The first stage addresses the proximity constraint, i.e., $\sum_{i=1,\dots,N_t} |p_{t,i} - p_{r,\pi(i)}|^2$. For this purpose, we build *candidate labels* $C_i \subset \{0, 1, \dots, N\}$ for each point $p_{t,i} \in I_t$. Specifically, for each $p_{t,i}$, we first choosing the three markers from I_r that are closest to it. Then further check these three candidates based on their absolute distances from $p_{t,i}$ and how they related to each other. After this step, the complexity of our matching task is largely reduced to picking the best candidate from C_i , which contains at most three candidates. The candidate set C_i may contain 0–3 candidate labels for point $p_{t,i}$. Our problem then contains two tasks: reducing the ambiguity for $|C_i| > 1$ and solving matching conflicts. Both tasks are dealt in the second stage.

The second stage addresses the geometric constraint, i.e., $\sum_{(i,j) \in E} c(p_{t,i}, p_{t,j}; p_{r,\pi(i)}, p_{r,\pi(j)})$. We use the pairwise constraints for this purpose. In particular, for each candidate label $i' \in C_i$, the distances from $p_{r,i'}$ to its linked neighbors should be similar to the distances from $p_{t,i}$ and its potential neighbors. This heuristic solution is effective in the physical therapy application, since high frame rate is used and the variation are either very small (for true correspondence) or are fairly large (for ghost points).

After these two stages, we also need to update the reference frame. This is based on fusing the previous reference frame with the new matching result. One challenge here is caused by missing markers, especially continuous missing markers. Fortunately, the pairwise constraints provide again reliable way to recover them. The basic idea is that, for a missing marker, the positions of its neighbors can provide a strong constraint for its position. The details of the whole algorithm is summarized in Algorithm 1. In the algorithm, the thresholds τ_k , $k = 1, 2, \dots, N$ are used for determine ghost points. We use $\tau_k = 0.25$ for all labels except for $\tau_6 = 0.4$, $\tau_9, \tau_{21} = 0.35$. The three exceptions are for elbows and T-1 bone's positions, which are usually unstable due to large human motion and system errors.

4 Experimental Results

4.1 Database

We use a human motion database including 38 sequences, 27 of which come from non-patients and the rest come from patients. The sequences consist of the subjects taking a

Algorithm 1. Automatic Marker Matching

```

1: Input: The reference frame  $I_r$  and the frame  $I_t$ .
2:  $\mathcal{U} \leftarrow \emptyset$ .
3: for  $i = 1..N_t$  do
4:   For  $p_{t,i}$ , find its three nearest markers in  $I_r$ , denote their labels as  $l_1, l_2, l_3$ .
5:   Calculate their distances  $d_1, d_2, d_3$  to  $p_{t,i}$ . Without loss of generality,  $d_1 \leq d_2 \leq d_3$ .
6:   if  $d_1 < d_2/4$  then
7:      $C_i \leftarrow \{l_1\}; \mathcal{U} \leftarrow \mathcal{U} \cup \{l_1\}$ 
8:   else
9:     if  $d_1 \geq d_2/4$  and  $d_1 < d_3/4$  then
10:       $C_i \leftarrow \{l_1, l_2\}$ 
11:     else
12:       $C_i \leftarrow \{l_1, l_2, l_3\}$ 
13:     end if
14:   end if
15: end for
16: Remove duplicates in  $\mathcal{U}$ 
17: for  $i = 1..N_t$  do
18:    $\pi(i) \leftarrow 0$ , /*default, ghost point*/
19:   if  $|C_i| == 1$  then
20:      $\pi(i) \leftarrow C_i(1)$ 
21:   else
22:     for  $k \in C_i$  do
23:       for  $p_{t,j}$  whose label  $j \in \mathcal{U} \cap E_k$ , where  $E_k = \{m : (m, k) \in E\}$  do
24:          $e_{k,j} = ||p - p_{t,j}| - D_{k,j}|/D_{k,j}$ 
25:       end for
26:        $e_k = \max_j \{e_{k,j}\}$ 
27:     end for
28:      $e = \min_k \{e_k\}$ 
29:      $l = \operatorname{argmin}_k \{e_k\}$ 
30:     if  $e < \tau_l$  then
31:        $\pi(i) \leftarrow l$ 
32:     end if
33:   end if
34: end for
35: Update the reference frame  $I_r$  accordingly.

```

few steps forward or backward, as well as sitting down or standing up. Each sequence contains approximately 7800 frames. All sequences are 120 frames per second. Each marker is labeled manually by an expert using semi-automatic software (Motion Analysis from Santa Rosa, CA) for comparison. As shown in Fig 2, the ground truth number of markers is 34 for all sequences, but because of missing or ghost markers, the number of markers in a frame varies from 28 to 36. Our algorithm only uses one frame as the manually labeled frame, which may contain ghost markers.

The sequences from non-patients usually have rare missing markers but may contain ghost markers, while the patient sequences often contain lots of missing and ghost markers. Therefore, the patient sequences are more difficult and interesting.

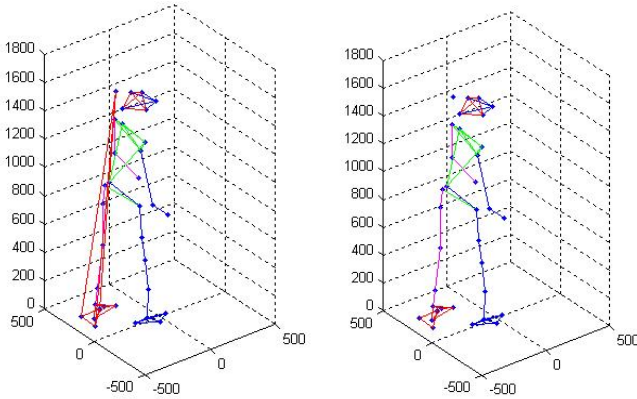


Fig. 3. A frame with different identifications. Left: results from semi-automatic labeling. Right: results of our method.

4.2 Comparison with Manual Marker Labeling

We compared the results of our algorithm with those from manual labeling using a semi-automatic commercial software. We compare the results frame-by-frame. We are mainly interested in frames where the two methods generate different labels. For the 27 non-patient videos, all the identification results are the same. This means that our method works as well as Motion Analysis. Using our algorithm eliminates the human intervention and is much faster (a few minutes as opposed to a few hours of manual labeling).

For the 11 patient videos, which are more pertinent within the context of physical therapy applications, our method generates different label configurations from the semi-automatic system in 2333 frames. By analyzing these frames one-by-one carefully, we find that whenever our method disagrees with the previous solutions, it is either due to the incorrect labeling of the previous system or due to the ambiguity in marker positions. Specifically, among the 2333 frames, our predictions are correct in 2259 frames, and the remaining 74 frames have ambiguous labels. One example is shown in Fig. 3. For example, in some frames, two markers may be very close in proximity to each other. Also, a marker may be largely disturbed in one axis for several frames.

In summary, the experimental results show that our method outperformed the commercial system with manual marker labeling. In addition, our method runs about 70 seconds per sequence with current Matlab implementation. This is extremely fast compared with previous solutions that usually take an expert five to six hours.

4.3 Comparison without Pairwise Constraints

We also compared our method with a method without pairwise constraints. The proposed method certainly outperforms nearest principal method when the frame is far away from human-labeled frame. This means that the inner distance constraints play an important role in matching.

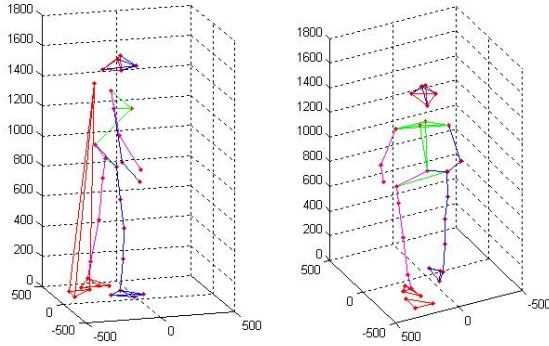


Fig. 4. A frame with different identifications. The marker near T-1 bone should be labeled 6(T-1 bone) as right figure showed, however, it's incorrectly labeled 34 by human.

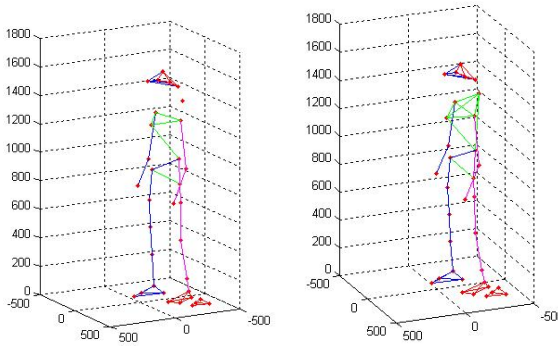


Fig. 5. A frame with different identifications. The marker near T-1 bone should be labeled 6 as shown in the right figure, however, it is incorrectly labeled as a ghost marker.

4.4 Comparison with Iterative Closest Point Algorithm

The Iterative Closest Point Algorithm [1122] is used to minimize the difference between two point clouds. The algorithm takes in two groups of points and outputs the transformation parameters between them. The first step is to associate the points between the groups by nearest neighbor criteria. Next, the transformation parameters are estimated using a mean square cost function. The points are then transformed using the estimated parameters, and the process is repeated a number of times based on a predetermined stopping criteria.

We can use this method as a comparison tool to our algorithm. Using ICP, we can go frame by frame and determine the point in the previous frame that corresponds to each point in the current frame. Using an initially labelled frame, we can find the corresponding points in the next frame, and then label these points with the same labels as in the previous frame. We repeat these process with the entire sequence, using the most recently labeled frame as the new ground truth each time. After this process, we will

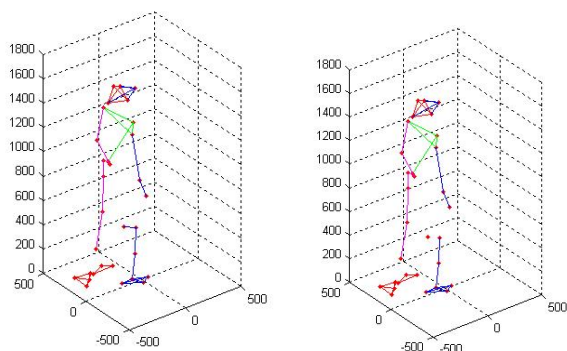


Fig. 6. A frame with different identifications. The marker near right leg should be a ghost marker as right figure showed, however, it is incorrectly labeled 13—right thigh.

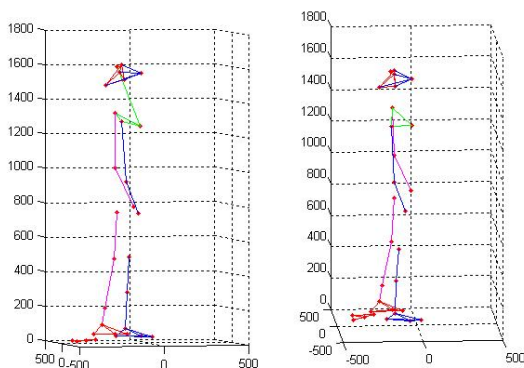


Fig. 7. A frame with different identifications. The marker near head should be a ghost marker as right figure showed, however, it is incorrectly labeled as 11—L4/L5.

have an array of frame label predictions in a similar format as the results produced in our algorithm.

Using fourteen different sequences, we used both the ICP algorithm and our algorithm to predict labels for all frames, given one initially labeled frame for each sequence. We then compared the labelling results for each frame from the two algorithms. If the results of a frame differed, a diagram of both algorithms' labels for that frame was saved as an image file. After going through all sequences, we went through each of the images created to see which algorithm produced a better set of labels for that frame. We have a correctly labeled initial frame for each sequence, so we know the basic structure to look for when analyzing these files. Figure 8 is an example of these image files.

Of the 19,620 frames found, the Automatic Marker Matching algorithm performed better in 17,842 of them (90.94%), while the ICP algorithm performed better in 689 frames (3.51%). In 1,089 frames (5.55%), the visual difference was too small to declare one algorithm more effective than the other.

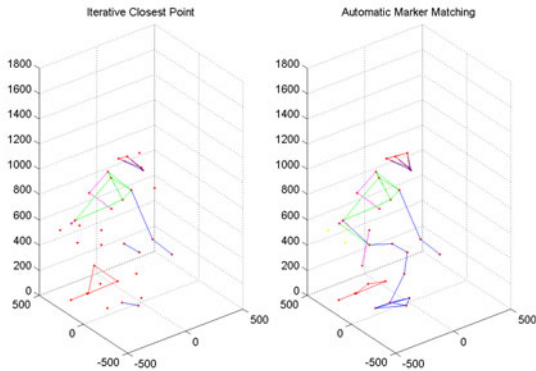


Fig. 8. An example comparison between the Iterative Closest Point algorithm and the Automatic Marker Matching algorithm. In this frame, the Automatic Marker Matching algorithm is more effective.

5 Conclusion

In this paper we reported a fast and robust algorithm for automatic infrared tracked marker identification for physical therapy applications. Our method uses both proximity and pairwise constraints. Experiments showed that our method not only generates better accuracy than the current commercial system, but also runs much faster. We expect to apply the reported method to production runs and exploit other potential motion analysis applications.

Acknowledgment. This work is supported in part by NSF Grant IIS-1049032, NIA-NIH grant AG26470. Gregory Johnson is supported by a fellowship supported by NSF Grant DGE-0841377.

References

1. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. *PAMI* 14, 239–256 (1992)
2. Brodia, T.J., Chellappa, R.: Estimation of Object Motion Parameters from a Sequence of Noisy Images. *PAMI* 8, 90–99 (1986)
3. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. *IJCV* 29, 5–28 (1998)
4. Dorfmueller-Ulhaas, K.: Robust optical user motion tracking using a kalman filter. In: *ACM VRST* (2003)
5. Herda, L., Fua, P., Plankers, R., Boulic, R., Thalmann, D.: Skeleton-based motion capture for robust reconstruction of human motion. In: *Proc. Computer Animation* (2000)
6. Hornung, A., Sar-Dessai, S., Kobbelt, L.: Self-calibrating optical motion tracking for articulated bodies. *IEEE Virtual Reality*, 75–82 (2005)

7. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *Int'l Wshp on Augmented Reality*, pp. 85–94 (1999)
8. Keshner, E.A., Kenyon, R.V.: Using immersive technology for postural research and rehabilitation. *Assist Technol. Summer* 16(1), 54–62 (2004)
9. Kurihara, K., Hoshino, S., Yamane, K., Nakamura, Y.: Optical motion capture system with pan-tilt camera tracking and real time data processing. In: *ICRA*, vol. 2 (2002)
10. van Liere, R., van Rhijn, A.: Search space reduction in optical tracking. In: *Proceedings of the Workshop on Virtual Environments*, pp. 207–214 (2003)
11. Ringer, M., Lasenby, J.: A procedure for automatically estimating model parameters in optical motion capture. *Image and Vision Computing* 22, 843–850 (2004)
12. Tolani, D., Goswami, A., Badler, N.I.: Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models* 62, 353–388 (1999)
13. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K.: The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In: *ACM VRST* (1999)
14. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. *ACM Computing Surveys* 38(4) (2006)
15. Zordan, V.B., Van Der Horst, N.C.: Mapping optical motion capture data to skeletal motion using a physical model. In: *ACM symp. on Computer Animation*, pp. 245–250 (2003)
16. Salari, V., Sethi, I.K.: Feature point correspondence in the presence of occlusion. *PAMI* 12(1), 87–91 (1990)
17. Sethi, I., Jain, R.: Finding trajectories of feature points in a monocular image sequence. *PAMI* 9(1), 56–73 (1987)
18. Rangarajan, K., Shah, M.: Establishing motion correspondence. In: *Conference Vision Graphics Image Process*, vol. 54(1), pp. 56–73 (1991)
19. Intille, S., Davis, J., Bobick, A.: Real-time closed-world tracking. In: *CVPR*, pp. 697–703 (1997)
20. Veenman, C., Reinders, M., Backer, E.: Resolving motion correspondence for densely moving points. *PAMI* 23(1), 54–72 (2001)
21. Shafique, K., Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. In: *ICCV*, pp. 110–115 (2003)
22. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *IJCV* 13, 119–152 (1994)

Learning and Prediction of Soft Object Deformation Using Visual Analysis of Robot Interactions

Ana-Maria Cretu, Pierre Payeur, and Emil M. Petriu

School of Information Technology and Engineering, University of Ottawa,
Ottawa, ON, Canada

Abstract. The paper discusses an innovative approach to acquire and learn deformable objects' properties to allow the incorporation of soft objects in virtualized reality applications or the control of dexterous manipulators. Contours of deformable objects are tracked in a sequence of images collected from a camera and correlated to the interaction measurements gathered at the fingers of a robotic hand using a combination of unsupervised and supervised neural network architectures. The advantage of the proposed methodology is that it not only automatically and implicitly captures the real elastic behavior of an object regardless of its material, but it is also able to predict the shape of its contour for previously unseen interactions. The results obtained show that the proposed approach is fast, insensitive to slight changes in contrast and lighting, and able to model accurately and predict severe contour deformations.

1 Introduction

Unlike virtual reality environments that are typically made of simplistic CAD models, a virtualized environment contains models that constitute conformal representations to real world objects [1]. Such representations have to preserve the visible details of the described real-world object and to accurately capture its properties. While several approaches exist to the modeling of the behavior of rigid objects, virtualized reality still needs to introduce accurate representations of deformable objects in order to fully reach its usability and functionality. Such representations are highly desirable in applications such as computer gaming and interactive virtual environments for training, and critical for other applications such as robotic assembly and medical robotics.

Inspired by the human experience with object manipulation where the ability of vision interacts with the servo-muscular and touch sensory systems for every day manipulation tasks, the work in this paper uses visual information and interaction parameters measured at the level of the fingers of a robotic hand for the acquisition and mapping of properties characterizing soft deformable objects. The solution extends a previously proposed algorithm that tracks deformable object contours in image sequences captured by a camera while the object deforms under the forces imposed by the fingers of a robotic hand [2]. Each contour is associated to the corresponding measured interaction parameters to characterize the object's shape deformation and implicitly describe its elastic behavior without knowledge on the material that the object is made of. Due to the choice of neuro-inspired approach used for

modeling, the solution not only captures the dynamics of the deformation, but is also capable to predict in real-time the behavior of an object under previously unrecorded interactions. Such a description enhances the accuracy of the models obtained and represents a significant advantage over existing deformable object models.

2 Related Work

As the latest research proves [3, 4], much of the current research on deformable models is still based on computer-generated models. The classical mass-spring models and finite-element methods still constitute the standard for virtual reality applications. In spite of their simplicity and their real-time simulation ability, mass-spring models are application-dependent, their behavior varies dramatically according to the choice of spring constants and their configuration. Also the models obtained have in general low accuracy. Finite-element methods can obtain more accurate models, but they are more complex and the very high computational time incurred as the object deforms (force vectors, mass and stiffness matrices are re-evaluated each time the object deforms) is a serious obstacle for their use in real-time applications.

An alternative solution to ensure better conformance to the reality of the deformable object model without requiring the pre-selection of elastic parameters or material properties (as in the case of mass-spring models) or requiring increased complexity and excessive computation times (as in the case of finite-element methods), is to interact with the objects in a controlled manner, observe and then try to mimic as accurately as possible the displayed object behavior. Neural networks are well-fitted for such tasks, as they are computationally simple, have the ability to map complex and non-linear data relationships and have the ability to learn and then predict in real-time the displayed behavior. This explains the interest of researchers from both the deformable object modeling [5, 6] and the grasping and manipulation research fields [7-11] into such techniques.

In the area of deformable object models, object deformation is formulated as a dynamic cellular network that propagates the energy generated by an external force among an object's mass points following Poisson equation in [5]. Greminger *et al.* [6] learn the behavior of an elastic object subject to an applied force, by means of a neural network which has as inputs the coordinates of a point over a non-deformed body (obtained by a computer vision tracking algorithm based on boundary-element method that builds on the equations of the elasticity) and the applied load on the body, and as outputs the coordinates of the same point in the deformed body.

In the area of robotic grasping and manipulation, neural networks have been employed to learn the complex functions that characterize the grasping and manipulation operations [7-11] and to achieve real-time interaction after training [10]. A neural network is used in [7] to approximate the dynamic system that describes the grasping force-optimization problem of multi-fingered robotic hands (the set of contact forces such that the object is held at the desired position and external forces are compensated). Pedreno-Molina *et al.* [8] integrate neural models to control the movement of a finger in a robotic manipulator based on information from force sensors. Howard and

Bekey [9] represent the viscoelastic behavior of a deformable object according to the Kelvin model and train a neural network for extracting the minimum force required for lifting it. A hierarchical self-organizing neural network to select proper grasping points in 2D is proposed in [10]. Chella *et al.* [11] use a neuro-genetic approach for solving the problem of three-finger grasp synthesis of planar objects.

Neural network architectures are chosen in the context of this work for reasons similar to those mentioned above, namely their capability to store (offline) and predict (online) the complex relationship between the deformation of the object and the interaction parameters at each robotic finger. However, unlike the other neural network approaches encountered in the literature, the proposed approach neither imposes certain equations to model the elastic behavior [5, 6] or certain dynamic models at the points of contact [7], nor requires a certain representation of the deformable object [9, 11]. The proposed solution combines in an original manner neural architectures to identify an object of interest, to track its contour in visual data and to associate and predict its shape under interactions exercised with a robotic hand.

3 Proposed Solution

In order to map the properties of an elastic object, its controlled interaction with a robotic hand is monitored by means of visual data collected with a camera, while additional measurements are collected using sensors installed in the three fingers of a Barrett robotic hand. The elastic properties are then mapped and learned as a complex relationship between the deformed contour of an object obtained from the visual data and the corresponding interaction parameters, as illustrated in Fig. 1 that summarizes the proposed approach. Neural network approaches are used both to segment and track the object contour in the image sequence [2] and to capture implicitly the complex relationship between the object's contour deformation and the force exercised on the object through the robotic fingers at defined finger positions. The choice to use a supervised (feedforward neural network) architecture for mapping the relationship between the interaction parameters and the corresponding contours is justified by the capability of the neural network to eliminate the need for predefined elastic parameters of the object or predefined object models. This aspect is essential in the proposed application, as most of the objects used for experimentations are made of soft, highly deformable material whose elastic behavior is very difficult to be described in terms of standard elastic parameters. The choice of a neural-network approach also ensures the ability of the application to estimate the contour of an object for previously unseen combinations of interaction parameters.

It is important to mention that the experimentation takes place in a relatively controlled environment, as data is collected separately for each modeled object. The solution does not have to deal with multiple moving objects and severe changes in the environment, but rather focuses on accurately tracking severe contour deformations that describe the object's behavior. For a proper identification of the object's elastic properties, it is considered that the object has already been grasped by the three robot fingers. Balanced grasping forces are initially applied to maintain the object within

VISUAL ANALYSIS OF INTERACTIONS

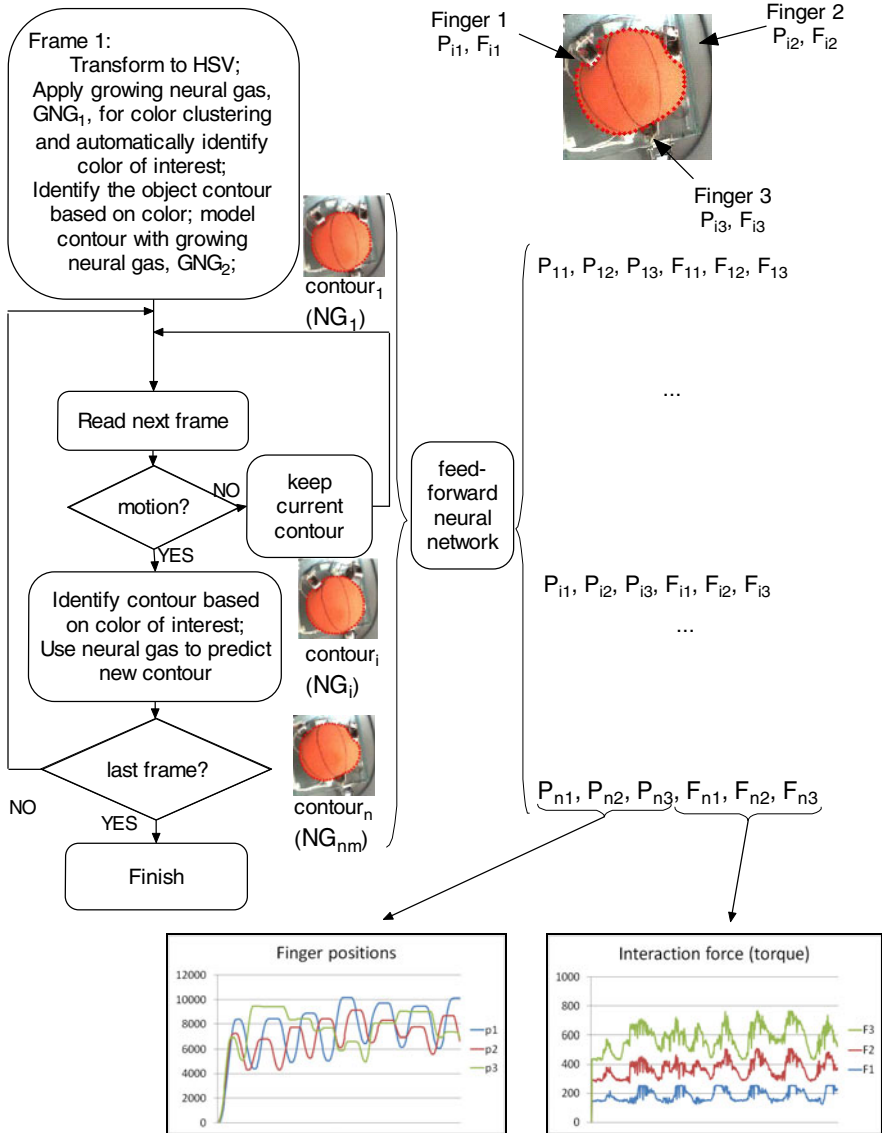


Fig. 1. Proposed framework to acquire, capture and predict the object contour shape based on the position of the robotic fingers and force measurements at level of the robotic fingers

the hand grip without slippage. The hand then compresses the object repetitively by contracting and relaxing its fingers over a period of several sampling periods.

The test is carried out using Barrett hand's real-time mode which allows the input fed to the hand system to generate a periodic movement of the fingers in order to

properly excite the object and extract its dynamic characteristics. The motion of the hand's three fingers is defined to follow a sinusoidal profile defined in encoder pulses. Two interaction parameters are recorded at each finger at every sampling instant, as illustrated in the lower part of Fig. 1. One corresponds to the position, P_{ij} , of each fingertip and is represented by the number of pulses in the encoder that reads the angle of the motor that drives the finger. This measurement is referred to as "position" measurement to simplify the explanations. It is equivalent to the Cartesian coordinates of the fingertip using the Barrett hand's kinematic model. The second parameter is a measure of the interaction force (torque), F_{ij} , applied at each fingertip and obtained via strain gauges embedded in each of the three fingers. It will be called hereon the "force" measurement, as the strain value can be converted to equivalent physical force measurements through proper calibration. These interaction parameters are collected simultaneously with an image sequence of the object's contour deformation as captured by the camera. Measurements are collected on a set of test objects made of soft deformable materials. The force and position measurements are then associated with the tracked contour of the object in the image sequence using an innovative combination of neural network architectures.

3.1 Visual Analysis of Interaction

The segmentation and tracking algorithm applied on visual data is illustrated in the left side of Fig. 1 in form of a flowchart and can be summarized as follows [2]: the initial frame of the sequence of images collected by the camera is used to identify automatically the object of interest by clustering the color (HSV coding) and spatial components (X, Y coordinates) of each pixel in this frame in two categories: object and background. The clustering is based on an unsupervised neural architecture, a growing neural gas, denoted GNG_1 . The reason for choosing an unsupervised network is the fact that, beyond being automated, it results in lower error rates when compared with a standard segmentation technique based on mean HSV values computed in a user selected frame that samples the object color. The color of interest is then automatically computed as the mean for all HSV values within the cluster representing the object of interest. The identified HSV color code is subsequently searched only over frames in the sequence where movement occurs to speed up the processing. The motion is detected based on intensity difference between the grayscale representations of the current and previous frames. The contour of the object is identified after straightforward image processing with a Sobel edge detector.

A second growing neural gas, GNG_2 , is used to map and represent the position of each point over the contour. Its main purpose is to detect the optimum number of points, c_n , on the contour that accurately represent its geometry. This compact growing neural gas description of the contour is then used as an initial configuration for a sequence of neural gas networks, NG_i , whose purpose is to track the contour over each frame in the image sequence in which motion occurs. A new neural gas network, initialized with the contour of the object in the previous frame, is used to predict and adjust the position of its neurons to fit the new contour. This new contour is used iteratively to initialize the next neural gas network in the sequence. The procedure is repeated until the last frame of the sequence, as illustrated in the flowchart of Fig. 1, resulting in n_m separate neural gas networks, as determined by the number of frames

exhibiting motion. The full description of the object segmentation and contour tracking algorithm is presented in [2]. The n_m contours representing each neural gas network are further associated to the measured interaction parameters for a comprehensive description of the object's deformation.

3.2 Mapping of Contours with Interaction Parameters

The n_m contours extracted from the sequence of images, as obtained in Section 3.1 are mapped with the interaction parameters using a feedforward neural network. The network capturing the behavior of each deformable object has six input neurons associated with the interaction parameters, namely the position of the three fingers (P_{i1} , P_{i2} , P_{i3}) and the force measurements (F_{i1} , F_{i2} , F_{i3}) at each fingertip, as shown in the right side of Fig. 1. A number of 30 hidden neurons was experimentally selected to ensure a good compromise between the length of training and the accuracy of modeling for all the objects used in the experimentation. The output vector is the set of coordinates for the points on the contour as obtained by tracking in the image sequence. It contains concatenated vectors of X and Y coordinates for each point in the contour and therefore its size is the double of the number of points, c_n , in the contour. This is also the number of nodes in the second growing neural gas network, GNG_2 , which defines those contours, and in the series of neural gas networks, NG_i . The input vectors (sets of P_{ij} and F_{ij} values) are normalized prior to training, and three quarters of the data available is used for training and a quarter for testing. Each network is trained using the batch version of scaled conjugate gradient backpropagation algorithm with the learning rate of 0.1 for 150,000 epochs. Once trained, the network takes as inputs the interaction parameters (P_{i1} , P_{i2} , P_{i3} , F_{i1} , F_{i2} , F_{i3}) and outputs the corresponding contour that the object should exhibit under the current configuration of interaction parameters.

4 Experimental Results

The proposed method has been applied on a set of deformable objects with different shapes and colors, of which a limited set is presented here, namely an orange foam ball and a green rectangular foam sponge. Fig. 2 illustrates the object segmentation and tracking algorithm for the ball, as described in Section 3.2 and illustrated on the left side of Fig. 1. The procedure starts with the automated identification of the color of interest (Fig. 2b) by clustering with GNG_1 the initial frame (Fig. 2a). A tolerance level (defined as the maximum distance for each component of the HSV coding from the mean HSV code that identifies the color of interest) is allowed to eliminate the effect of non-uniform illumination and shadow effects, e.g. color reflected by the shining fingers (Fig. 2c). The contour is then identified using Sobel edge detector (Fig. 2d and 2e) and modeled with an optimal number of nodes, c_n , using the second growing neural gas, GNG_2 (Fig. 2f). Finally the object is tracked over the series of frames using a sequence of neural gas networks. The computation time required to track the objects is low, on average 0.35s per frame on the Matlab platform. The average error, measured as the Hausdorff distance between the points in the contour

obtained with the Sobel edge detector and the modeled neural gas points is of the order 0.215. The Hausdorff distance is chosen because it allows the comparison of curves with different number of points, as it is the case between the neural gas model and the Sobel contour.

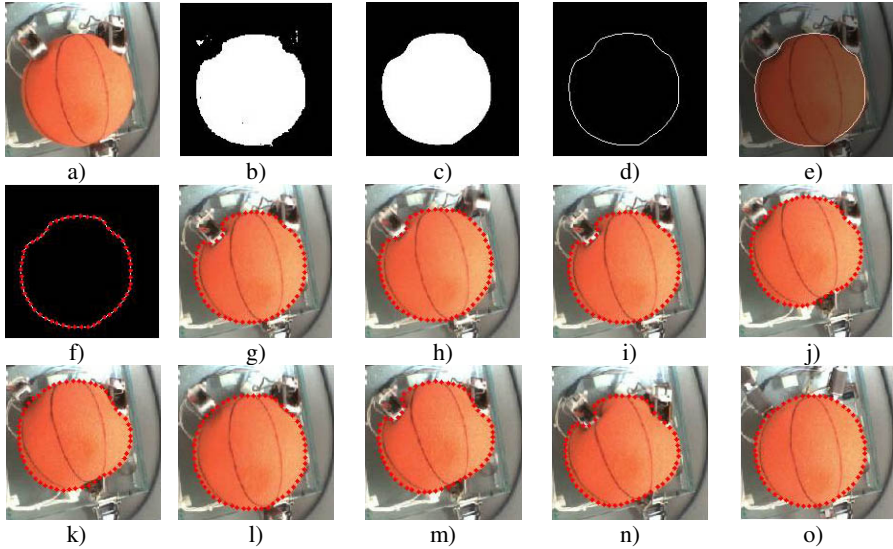


Fig. 2. Object segmentation and tracking based on unsupervised architectures: a) initial frame, b) identification of the color of interest, c) results when a tolerance level is accepted, d) contour identification, e) contour overlapped on initial frame, f-g) growing neural gas model of contour in the initial frame and g) - o) contour tracking using a series of neural gas networks

Table 1 illustrates the average computation time per frame (in seconds) as well as the error incurred during tracking for each of the objects under study. The error is slightly higher for objects that deform rapidly from one frame to the other or roll during probing, as in the case of the ball illustrated in Fig. 2. It can be observed that the tracking procedure is fast and follows closely the contours of the objects. A typical characteristic of the proposed tracking algorithm is the fact that the nodes in the contour retain their correspondence throughout the deformation. This behavior is due to the choice of a fixed number of nodes in the neural gas network, NG_i , and to the proposed learning mechanism. Fig. 3a details a part of the trajectory (marked with arrows) that the points in the neural gas model of the ball follow as a result of the interaction with the robotic hand.

Table 1. Average error and average time per frame for the tracking algorithm

Object	Average error	Average time per frame [s]
Round ball	0.269	0.44
Rectangular sponge	0.189	0.306
Yellow curved sponge	0.187	0.371

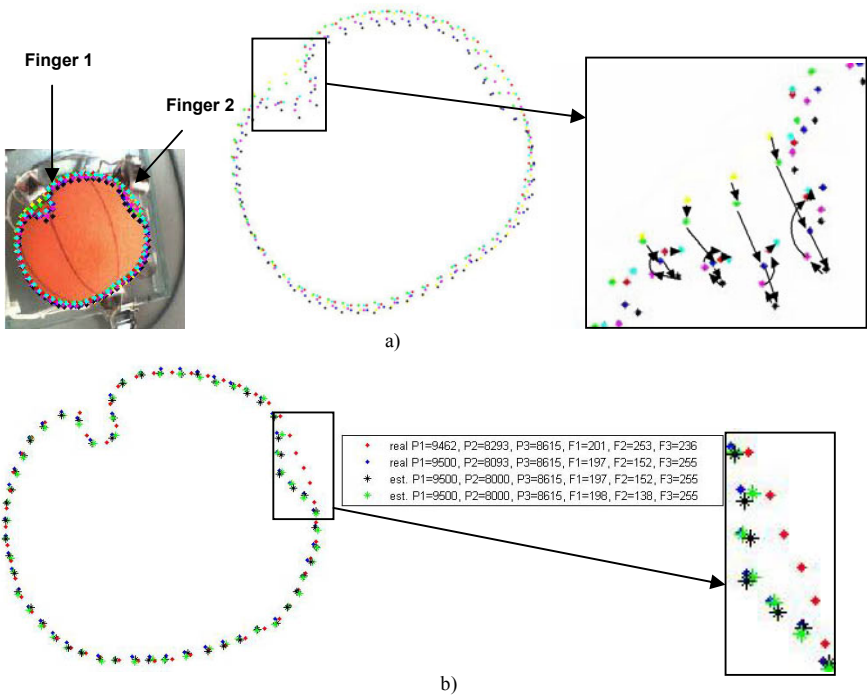


Fig. 3. a) Trajectory of nodes during compression and slight depression of ball and b) real (*dot*) and estimated (*star*) contour points and detail for the ball

It can be seen in the detailed image, that there is a one-to-one correspondence of the points in each contour. This correspondence ensures a unified description of the contour deformation throughout the entire sequence of images and at the same time avoids the mismatch of points during deformation. Such a property usually requires sophisticated feature descriptors to be defined in most tracking algorithms. The sequence of contours is finally mapped with the corresponding interaction parameters at the level of the robotic hand by means of a feedforward neural network as described in Section 3.2. The performance of the neural network solution is illustrated in Figs. 3b and 4. The training/learning error for the neural networks corresponding to these two objects is of the order 5×10^{-5} , illustrating the capability of the network to accurately map the interaction parameters to the corresponding deformed contour. The testing error is of the order 4×10^{-3} . To validate the prediction abilities of the network, tests are conducted on two datasets that were not part of the training or the testing sets. In the first example depicted in Fig. 3b, the position of the Finger 2 is moved slightly lower (from $P_{2 \text{ blue}}=8093$ to $P_2=8000$) while the other input parameters are kept unchanged with respect to the blue contour (parameters P_1 , P_3 , F_1 , F_2 , F_3 are the same as those in the blue-dot profile extracted from real measurements). The estimated contour marked with black stars passes slightly under the one at $P_{2 \text{ blue}}=8093$ marked with blue dots, with the peak slightly lower, as it is expected because the finger was moved lower. In the second example, also shown in Fig. 3b, the force at

the first finger is slightly increased ($F_1=198$ from $F_{1blue}=197$), while the one applied at the second finger is slightly decreased ($F_2=138$ from $F_{2blue}=152$) from the values in the measured blue dot contour. As a result of the increased forced at first finger, the estimated contour denoted by green stars goes slightly deeper than the contours marked in blue and black (as they both have the same position and force applied at this finger). As well, as it is expected, under the slightly reduced force at second finger, the estimated contour goes below (more to the right) both the blue dot and black star contour and gets slightly closer to the red dot one, as it can be observed in the detailed image. Another example is presented for the rectangular sponge in Fig. 4.

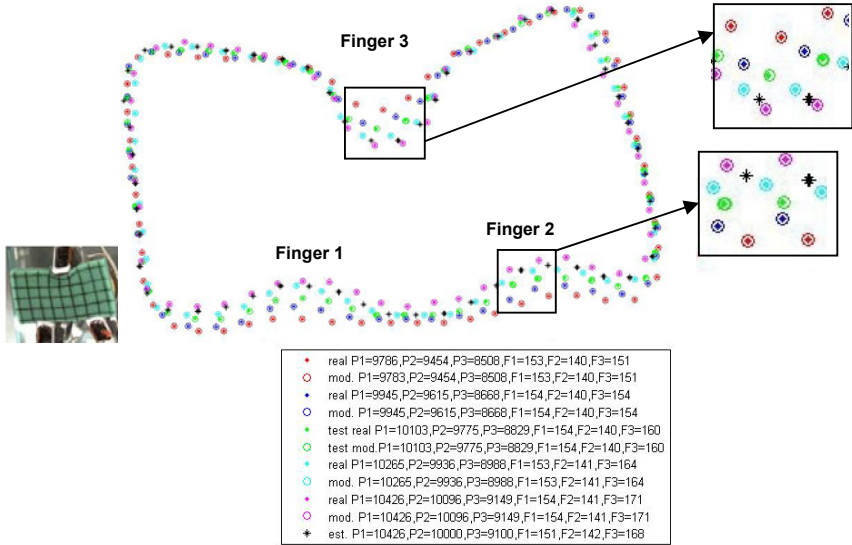


Fig. 4. Real (dot), modeled (circle) and estimated (star) contour points for the green sponge

In this example, the network provides an estimate for a value of the force at the third finger ($F_3=168$) that is in between the one of the cyan contour ($F_{3cyan}=164$) and the magenta contour ($F_{3magenta}=171$), associated as well with a slight movement in the position of the fingers. Such a testing scenario where the force value is close to modeled values is chosen as an example to allow the observation and quantification of the results in spite of the complexity of the object behavior. One can notice multiple interactions and changes in the contour that occur due to increased forces at different fingers. For example, while the forces at Fingers 1 and 2 in Fig. 4 are almost unchanged, an increased force at Finger 3 creates a bending in the object that appears to be due to the forces applied at Fingers 1 and 2. A careful observation leads to the conclusion that the estimate is still correctly placed around the extra fingers under these conditions. The proposed approach is therefore able not only to capture the contour, but also to predict its shape in spite of the coupling between multiple finger interactions.

5 Conclusion

This paper discusses an original approach to merge visual and force measurements to achieve models of deformable objects relying on experimental manipulation of actual objects. It demonstrates the advantage of using neural networks solutions to map and predict the contour shape of soft deformable objects from real measurements collected by a robotic hand and a camera. The neural network inspired algorithm for segmentation and tracking runs fast, with low errors and guarantees the continuity of points in the tracked contours unlike any classical tracking solutions. The neural network approach for the modeling and prediction of contour shapes based on force measurements and the position of the robotic fingers ensures that the application handles properly previously unseen situations on which the system was not trained. The study will be expanded in future work for different orientations of the robot fingers and for additional parameters in order to achieve a more extensive description of the interaction.

References

1. Petriu, E.M.: Neural Networks for Measurement and Instrumentation in Virtual Environments. In: Ablameyko, S., Goras, L., Gori, M., Piuri, V. (eds.) *Neural Networks for Instrumentation, Measurement and Related Industrial Applications*. NATO Science Series III: Computer and System Sciences, vol. 185, pp. 273–290. IOS Press, Amsterdam (2003)
2. Cretu, A.-M., Payeur, P., Petriu, E.M., Khalil, F.: Deformable Object Segmentation and Contour Tracking in Image Sequences Using Unsupervised Networks. In: *Canadian Conference Computer and Robot Vision*, pp. 277–284. IEEE Press, Ottawa (2010)
3. Wang, H., Wang, Y., Esen, H.: Modeling of Deformable Objects in Haptic Rendering System for Virtual Reality. In: *IEEE Conference on Mechatronics and Automation*, pp. 90–94. IEEE Press, Changchun (2009)
4. Luo, Q., Xiao, J.: Modeling and Rendering Contact Torques and Twisting Effects on Deformable Objects in Haptic Interaction. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 2095–2100. IEEE Press, San Diego (2007)
5. Zhong, Y., Shirinzadeh, B., Alici, G., Smith, J.: Cellular Neural Network Based Deformation Simulation with Haptic Force Feedback. In: *Workshop Advanced Motion Control*, pp. 380–385. IEEE Press, Turkey (2006)
6. Greminger, M., Nelson, B.J.: Modeling Elastic Objects with Neural Networks for Vision-Based Force Measurement. In: *International Conference on Intelligent Robots and Systems*, pp. 1278–1283. IEEE Press, Las Vegas (2003)
7. Xia, Y., Wang, J., Fok, L.M.: Grasping-Force Optimization for Multifingered Robotic Hands Using Recurrent Neural Network. *IEEE Trans. Robotics Automation* 26(9), 549–554 (2004)
8. Pedreno-Molina, J., González, A.G., Moran, J.C., Gorce, P.: A Neural Tactile Architecture Applied to Real-time Stiffness Estimation for a Large Scale of Robotic Grasping Systems. *Intelligent Robot Systems* 49(4), 311–323 (2007)
9. Howard, A.H., Bekey, G.: Intelligent Learning for Deformable Object Manipulation. *Autonomous Robots* 9(1), 51–58 (2000)
10. Foresti, G.L., Pellegrino, F.A.: Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation. *IEEE Trans. Systems, Man Cybernetics* 34(3), 325–333 (2004)
11. Chella, A., Dindo, H., Matraxia, F., Pirrone, R.: Real-Time Visual Grasp Synthesis Using Genetic Algorithms and Neural Networks. In: Basili, R., Paziienza, M.T. (eds.) *AI*IA 2007*. LNCS (LNAI), vol. 4733, pp. 567–578. Springer, Heidelberg (2007)

A Novel Consistency Regularizer for Meshless Nonrigid Image Registration

Wei Liu and Eraldo Ribeiro

Computer Vision and Bio-Inspired Computing Laboratory
Florida Institute of Technology
Melbourne, FL 32901, U.S.A.

Abstract. In nonrigid motion analysis, deformation fields are often modeled using splines defined on a control-point grid. Inspired by recent development of meshfree methods, we propose a novel motion model that does not use control-point grids, nor use explicit node connections. We also propose a regularizer for the deformation field and the minimization algorithm. The method has promising features such as the handling of irregular regions, adaptive accuracy, the multi-scale modeling, and the potential for integrating physical properties into the registration process. Promising results were obtained on both synthetic and real imagery.

Keywords: nonrigid meshless image registration, meshfree methods.

1 Introduction

Spline-based free-form deformation models have been successfully applied in nonrigid image registration methods [1,2]. However, they suffer from a number of problems including the reliance on explicitly connected control points, the conflict between grid resolution and computational efficiency, and the difficulty to handle topological changes. Recent efforts using adaptive irregular grids [2] still rely heavily on explicit control-point connections or meshes, and require nontrivial handling of topology information. In this paper, we propose a novel mesh-free approach for nonrigid image registration that eliminates the need for explicitly handling topology information in the registration process.

Spline-based models' limitation of relying on control-point grids is fundamental, and is shared by applications such as computational mathematics [3], mechanical engineering [4], and computer graphics [5]. Recently, many meshless models (i.e., no explicit control-point connections) have been proposed in different areas [4,5,3,6,7]. In this paper, we will focus ourselves on the problem of nonrigid image registration while also drawing from the results in other areas.

For nonrigid image registration, early meshless models are based on radial basis functions (RBFs). For example, the thin-plate spline model [8] represents nonrigid deformations by a linear combination of locally supported RBFs that are scattered over the computation domain. The local basis functions are blended together without explicitly connecting the pieces, and thus avoiding the need for maintaining a control-point grid. Recently, Rohde et al. [8] extended this model

to adaptively select the number and position of the basis functions according to registration residues. RBFs-based meshless models are also used for data interpolation [3], and for solving partial differential equations (PDEs) [4]. However, RBFs-based models have two main problems. First, as Ruecket et al. [1] pointed out, due to prohibitive computational complexity of the thin-plate spline warps, the registration is restricted to a very limited number of degrees of freedom. Secondly, RBFs are unable to represent polynomial functions exactly (i.e., lack of polynomial reproducibility [4]). For example, RBFs cannot exactly represent a constant deformation $f(\mathbf{x}) = c$, $c \neq 0$. It has been shown that RBFs are less accurate than polynomial-based models in finite-element methods [4]. Here, we propose a polynomial-based meshless model for nonrigid image registration.

Some existing works represent nonrigid deformation using polynomial-based meshless models. For example, polynomial-based moving least-squares (MLS) method has been used for heart-motion analysis [7], and prostate image registration [9]. Most notably, Makram-Ebeid et al. [6] proposed a nonrigid image registration method based on a partition-of-unity finite-element method (PUFE). However, these approaches inherit heavily from meshless methods developed for mechanical engineering, and essentially solve mechanical PDEs using image evidence as boundary conditions. A major difference with respect to previous related work, is that our method does not rely on prior segmentation. Here, we directly integrate the meshless deformation model into the nonrigid image registration formulation, leading to a simplified registration framework.

Our main contributions are as follows. First, we propose a novel image registration method that represents deformation fields by blending together locally supported polynomial models using partition-of-unity (PU) (Section 2). The local deformation models are defined at scattered nodes, and their supporting domains are restricted by radial weighting functions. Secondly, we introduce a new functional to penalize for inconsistencies between local deformation fields (Section 3). This regularizer greatly simplifies the registration process compared to the classic regularizer based on Sobolev norm [1], or to the non-conformity measure proposed in [6]. Additionally, our regularizer is not biased towards certain lower-order deformations. The proposed functional can be minimized hierarchically at varying scales. Unlike previous methods, where a coarse-scale deformation field is only used as an intermediate result [1,6], our method is able to regularize the deformation field by combining image evidence at different scales *simultaneously*. Finally, these contributions are supported by a number of experiments performed on both synthetic and real images (Section 4).

2 Meshless Deformation Model

We commence by defining the nonrigid image-registration problem as that of deforming a source image $I(\mathbf{x})$ to “best” match a target image $I'(\mathbf{x})$ with respect to a given similarity measure [8]. Formally, we seek for a warping field $\mathbf{u}(\mathbf{x})$ that satisfies the following equation $\arg \max_{\mathbf{x}'} F(I'(\mathbf{x}'), I(\mathbf{x}), \mathbf{x}')$, $\mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x})$, where \mathbf{x} is a coordinate vector, and F is the intensity-based similarity measure.

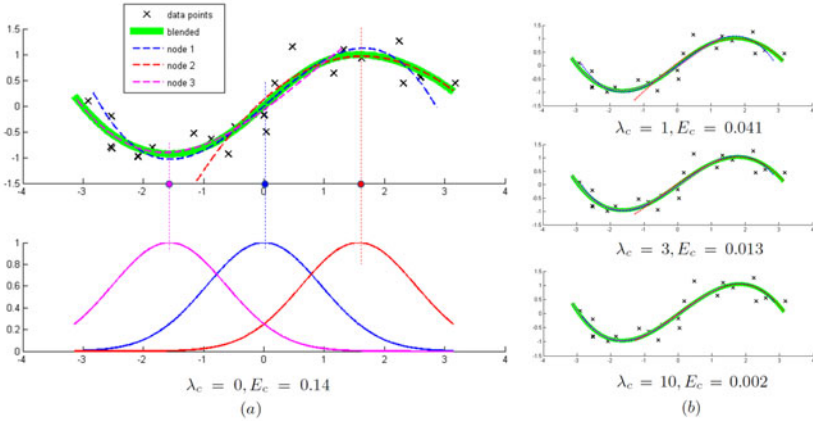


Fig. 1. Meshless scattered-data approximation using our consistency regularizer. (a) The scattered data-points are locally approximated around each node using polynomials (dashed curves), and then blended into a global approximation (green solid curve) based on their weighting functions (bottom). (b) The consistency regularizer penalizes inconsistencies between local models. Increasing the regularizer coefficient λ_c leads to more consistent local approximations with lower inconsistency energy E_c .

Robustness and efficiency are often obtained by representing \mathbf{x} using a sparse set of control points interpolated by splines [11]. To avoid explicit handling of neighborhood information, we approximate the global deformation field $\mathbf{u}(\mathbf{x})$ by blending together a set of particle-based local approximations of the deformation. Thus, no explicit connections between control points are needed. Figure 1 illustrates the concept of our approximation model and consistency regularizer, using an example problem of scattered-data approximation. Here, a set of nodes is distributed across the deformation field domain. Around each node, a local polynomial approximation of the deformation field $\mathbf{u}(\mathbf{x})$ is obtained. As polynomials do not have compact support, we restrict the approximation to the node’s vicinity, a region called the node’s *influence domain*. The influence domain will also be used to limit the interaction range between neighboring nodes.

The node’s influence domain. Let us define the influence domain D_p around a node p as a disk of radius r_p (ball in 3-D¹). D_p can be modeled by a weighting function $w_p(\mathbf{x})$ with local support [4]. We define $w_p(\mathbf{x})$ as:

$$w_p(\mathbf{x}) = \begin{cases} \alpha_p \exp\left(-\kappa \frac{\|\mathbf{p}-\mathbf{x}\|^2}{\sigma_p^2}\right), & \mathbf{x} \in D_p \\ 0 & \mathbf{x} \notin D_p \end{cases}, \quad (1)$$

where \mathbf{p} is the coordinate vector of node p , $\kappa = \frac{1}{9}$, and $\alpha_p \in (0, 1]$ is the node’s predefined influence factor in the final global blending. Thus, a node p is defined by its spatial position, \mathbf{p} , the radius (scale) of its influence domain, σ_p , and its

¹ 3-D extension is straightforward.

influence factor, α_p . It is worth pointing out that although the weighting function in (II) is a radial function, it is different from previous RBF-based models such as thin-plate splines [8]. Here, the radial basis functions are used for blending the local polynomial models, instead of directly representing the image deformation.

Local approximation model around a node. We write the deformation field $\mathbf{u}_p = (u, v)$ around particle p as a linear combination of monomials $\gamma_j(\mathbf{x}) = x^s y^t$:

$$u(\mathbf{x}) = \sum_{j=0}^m \gamma_j(\mathbf{x})a_j \quad \text{and} \quad v(\mathbf{x}) = \sum_{j=0}^m \gamma_j(\mathbf{x})b_j. \quad (2)$$

Thus, the deformation field is a linear combination of monomial basis functions $\phi^\top(\mathbf{x}) = [1, x, y, xy, \dots, x^s y^t, \dots]$, with coefficients $\mathbf{d}_p = [a_0, b_0, \dots, a_m, b_m]^\top$. Monomials in ϕ are ordered in a Pascal triangle manner for numerical stability [4]. Next, local deformations \mathbf{u}_p are combined into a global deformation field.

Blending local models into a single deformation field. Given the parameters of local deformation models, deformation at point \mathbf{x} can be obtained by blending together the nodes that contain \mathbf{x} in their influence domains. These nodes are called the *support domain* [4] of \mathbf{x} , and are formally denoted by $N_{\mathbf{x}} = \{p \mid \mathbf{x} \in D_p\}$. The blended global deformation field is obtained as $\mathbf{U}(\mathbf{x}) = \sum_{p \in N_{\mathbf{x}}} r_p(\mathbf{x})\mathbf{u}_p(\mathbf{x})$, with $r_p(\mathbf{x}) = \frac{w_p(\mathbf{x})}{\sum_{p' \in N_{\mathbf{x}}} w_{p'}(\mathbf{x})}$. Here, functions $r_p(\mathbf{x})$ ensure the partition-of-unity (PU), i.e., the contributions at \mathbf{x} from various nodes add to one. This blending scheme was used in [6] and is equivalent to the one used in the polyaffine model [10]. The scheme allows for arbitrary placement of local models, while the polynomial basis allows for accurate deformation representation.

3 Optical Flow Based Registration

In this section, we integrate the proposed deformation model with squared-sum-of-difference (SSD) similarity measure that assumes the image brightness remains constant during image deformation [1]. Study on integrating other similarity measures (e.g., mutual information) is underway. For improved efficiency, the brightness-constancy condition is often linearized to form the optical-flow constraint $I_x u + I_y v + I_t = 0$ [6,2]. Combining the optical-flow constraint and our meshless deformation model results in an image registration method that is both simple and elegant. It is worth pointing out that the choice of image deformation model is often independent from specific similarity measures, and our meshless deformation model can be easily combined with other similarity measures such as cross-correlation (CC) [8] and mutual-information (MI) [1].

Local intensity constraint. Locally, the weighted sum of squared residues based on optical-flow constraint within a particle's influence domain is given by:

$$E_p^d = \sum_{\mathbf{x} \in D_p} w_p(\mathbf{x}) (I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x}) + I_t(\mathbf{x}))^2, \quad (3)$$

where w_p is the weighting function defined in Equation 1. By plugging the local model defined in (2) into the registration residual measure in (3), we obtain:

$$E_p^d = (\mathbf{R}_p^\top \mathbf{d}_p + \mathbf{T}_p)^\top \mathbf{W}_p^d (\mathbf{R}_p^\top \mathbf{d}_p + \mathbf{T}_p), \tag{4}$$

where \mathbf{W}_p^d is the weighting function written as a diagonal matrix, the columns of \mathbf{R}_p are the vectors $[I_x(\mathbf{x})\gamma_0(\mathbf{x}), I_y(\mathbf{x})\gamma_0(\mathbf{x}), \dots, I_x(\mathbf{x})\gamma_s(\mathbf{x}), I_y(\mathbf{x})\gamma_s(\mathbf{x}), \dots]^\top$ for $\mathbf{x} \in D_p$, \mathbf{T}_p is the vector form of $I_t(\mathbf{x})$, and $\mathbf{d}_p = [a_0, b_0, \dots, a_m, b_m]_p^\top$ is the coefficient vector. Equation 4 summarizes the local image evidence.

Consistency enforcement. In spline-based methods, the global deformation field is consistent across control points, and regularization is obtained by penalizing the deformation’s spatial variation. In our framework, global deformation fluctuations cause inconsistencies among local deformation fields (Figure 1). As a result, rather than penalizing deformation’s spatial variation [6,21], we penalize the variation’s *spatial inconsistency*. Indeed, this produces a regularizer that is not biased towards the deformation field’s lower-order fluctuations, provided the fluctuation itself is spatially consistent. Consistency between two local deformation fields, \mathbf{u}_p and \mathbf{u}_q , can be measured from the coefficient vectors \mathbf{d}_p and \mathbf{d}_q . However, \mathbf{u}_p and \mathbf{u}_q lie on different local coordinate systems, and therefore need to be aligned (shifted). Shifting the basis functions ϕ by $\Delta\mathbf{x} = [\delta x, \delta y]$ leads to:

$$\begin{aligned} \phi(\mathbf{x} + \Delta\mathbf{x}) &= [1, x + \delta x, y + \delta y, (x + \delta x)(y + \delta y), \dots, (y + \delta y)^m]^\top \\ &= \mathbf{S}^\top(\Delta\mathbf{x})\phi(\mathbf{x}), \end{aligned} \tag{5}$$

where $\mathbf{S}^\top(\Delta\mathbf{x})$ is a linear *basis-shifting-operator* that can be written as a matrix. Therefore, shifting the local coordinate system leads to shifted coefficients. The local deformation consistency between two nodes p and q can be defined as:

$$E_{p,q}^c = [\mathbf{S}'(\mathbf{p} - \mathbf{q})\mathbf{d}_q - \mathbf{d}_p]^\top [\mathbf{S}'(\mathbf{p} - \mathbf{q})\mathbf{d}_q - \mathbf{d}_p]. \tag{6}$$

Here, the operator $\mathbf{S}'(\mathbf{p} - \mathbf{q})$ is obtained by simply duplicating and shifting the elements of the basis-shifting-operator $\mathbf{S}(\mathbf{p} - \mathbf{q})$ so that the multiplication holds. We can now combine the local data term and the global consistency term into a single functional-minimization problem as follows:

$$\mathbf{d}_p = \arg \min_{\mathbf{d}_p} \sum_p \left[E_p^d + \lambda_c \sum_{q \in N_p} w_q(\|\mathbf{p} - \mathbf{q}\|) E_{p,q}^c \right]. \tag{7}$$

Here, λ_c defines the consistency regularizer’s relative importance. Instead of using a single parameter to control the trade-off between data terms and regularizers [6,1], Equation 7 allows for different nodes to influence the smoothness regularizer, which suggests a potential regularizer adapted to image content. We implemented a Levenberg-Marquardt algorithm to solve the minimization.

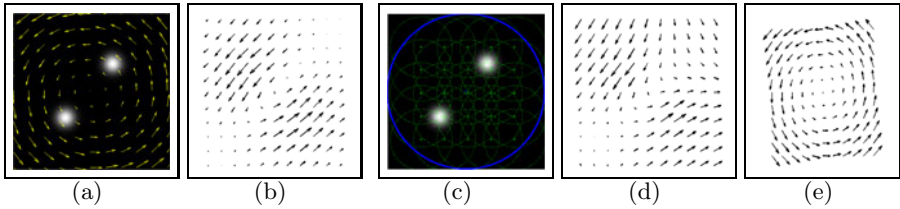


Fig. 2. Nodes' coupling with varying scales. (a) Ground truth. (b) B-spline registration (Rueckert's [1]). (c) A global node (in blue) with regularly distributed smaller nodes (in green). (d) Our method without global node. (e) Our method with the global node.

Hierarchical warping and multi-scale coupling. We adopt a hierarchical warping process to avoid local minima and improve computational efficiency. Convergence is achieved using a coarser-to-fine manner. Rather than downsizing the image [6], we simply use large scale nodes for coarse registration. In this case, less nodes are needed at large scales, leading to computation efficiency without sacrificing image data. Our framework allows for randomly distributed nodes of different scales. More interestingly, both coarse and fine deformation estimations can be considered *simultaneously*. This multi-scale ability leads to interesting results as shown in Figure 2. The synthetic example shows a counter-clockwise rotation about the image center with little data evidence. In this case, many ambiguous solutions exist. For the demons [13] and spline-based [1] methods presented in this example, the lack of higher-level information causes a dominant smoothness term to exaggerate deformations at textureless regions. By integrating a node at global scale (blue circle in Figure 2(c)), the simpler rotation solution becomes favored by our energy functional, even though that node's influence power is very small (less than 1% of the smaller node's weight). While both the exaggerated and the rotation solutions are legitimate, the rotation field is simpler and arguably more consistent with human perception. This large-scale node incorporates into the minimization higher-level information about the deformation field's shape, with little effect on smaller-scale image evidence.

4 Experiments

Experiments were performed on two data sets. We first used synthetic fluid PIV images obtained from the FLUID Project [11] which contains seven sequences varying from simple vortices to flows around obstacles. The database's low-speed fluids suits our evaluation, and registration algorithms produce comparable results to fluid-motion estimation methods. Secondly, we used 2-D cardiac MRI data from [12], and compared our method with Rueckert's B-spline based method [1], and an improved Demons method implemented by [13].

4.1 Parameter Settings

Similarly to both finite-element and meshless methods [4], the accuracy of our representation increases with the density of nodes (the h problem), and on the

order of basis polynomials (the p problem) for a higher computational cost. We experimentally found that second-order polynomials $\{1, x, y, xy, x^2, y^2\}$ provide sufficient representation accuracy at reasonable computational cost. In all experiments, we used second-order polynomials as basis functions. Second-order polynomials are also the empirical choices in many meshless algorithms [4,6]. Both the PIV and MRI images are of the size 256×256 pixels, and we used a grid of $21 \times 21 = 441$ nodes, i.e., the inter-node spacing was roughly $d = 12$ pixels. Finally, in meshless methods, the nodes' average radius r_a controls the average number of interacting neighbors (i.e., bandwidth of the system). Higher bandwidth often increases accuracy and stability at the cost of computation. The average radius r_a can be defined relative to the inter-node spacing d , i.e., $r_a = k \times d$, for some $k > 0$. We experimentally chose $k = 2.5$, so that the interacting neighbors average number was around 20.

For the consistency regularizer, we chose unity influence factor for all nodes, i.e., $\alpha_p = 1$, and then roughly tuned the weighting parameter λ_c on PIV sequences. Experiments show that $\lambda_c = 10^{-2}$ produced the best registration results. For Rueckert's method, we chose the same control-point density as our meshless method, i.e., a grid of $21 \times 21 = 441$ control-points. Again, regularizer parameter λ was tuned to the PIV sequences, and set $\lambda = 10^{-3}$. For the Demon method, we chose the default parameters provided by [13].

4.2 Analytic Fluids

Regularly and randomly distributed nodes. First, a regular node distribution was used. Secondly, nodes' positions were randomly perturbed (standard deviation of $[-8, 8]$), while nodes' radii were randomly generated, $r_p \in [32, 48]$. Finally, for all test cases, we added 15 percent of independent random noise to the input images. Tables 1 and 2 show the results for the different registration methods. Here, we borrowed the Average End-Point Error and the Average

Table 1. Average End-Point Error on Analytic Fluid Sequence

Method	Seq.1	Seq.2	Seq.3	Seq.4	Seq.5	Seq.6	Seq.7
B-Spline	0.04	0.03	0.04	0.13	0.10	0.04	0.25
Demons	0.15	0.13	0.16	0.11	0.12	0.13	0.22
Meshless	0.02	0.02	0.02	0.06	0.06	0.04	0.17
Meshless Random	0.03	0.02	0.02	0.06	0.06	0.14	0.17

Table 2. Average Angular Error on Analytic Fluid Sequence

Method	Seq.1	Seq.2	Seq.3	Seq.4	Seq.5	Seq.6	Seq.7
B-Spline	0.84	1.70	1.04	6.61	5.04	1.35	12.61
Demons	3.79	6.09	4.15	5.59	6.03	4.32	11.11
Meshless	0.53	0.95	0.51	2.55	2.63	1.23	8.40
Meshless Random	0.77	1.18	0.55	2.82	2.90	2.92	8.26

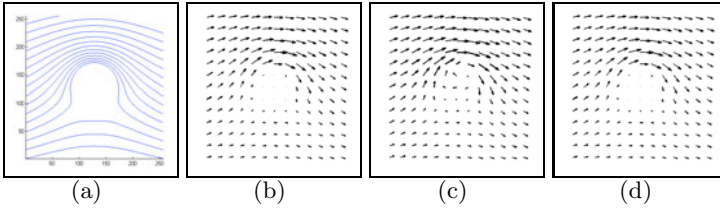


Fig. 3. Fluid around an obstacle. Grid-based methods overestimate the flow. Our method produces accurate results by removing nodes from the obstacle region. (a) streamline, (b) ground truth, (c) regular, and (d) adapted.

Angular Error metrics from the optical-flow literature [2]. The meshless method with regularly distributed nodes produced the best registration results under both metrics. When using randomly perturbed nodes, the nodes’ density and registration accuracy are downgraded in some regions, but our method still produced better results, with good robustness to image noise.

Handling topology challenges. In Figure 3, we show a FLUID sequence (Sequence 6) containing a flow around a cylindrical obstacle. When using our method with regularly-distributed nodes or a control-point grid for the spline-based method, the obstacle region was estimated as part of the flow (Figure 3(c)). In our method, we can handle holes in the blended global deformation by simply removing the nodes from the regions containing the obstacle (Figure 3(d)).

4.3 Cardiac MRI Sequence

Here, we used cardiac MRI slices from a dataset containing of 33 4-D MRI sequences from different patients [12]. Sequences have 20 frames of a 3-D cardiac motion. We performed intra-subject registration of 2-D slices by registering each slice to the next frame in the sequence. As ground truth was not available in this case, we used both Root-Mean Square (RMS) and Cross-Correlation (CC) as error metrics. While RMS and CC only provide visual verification of the registration results, they still provide good indication of registration quality. Evaluation of our meshless method using clinical images with manually labeled ground-truth data is currently underway and will be reported in due time. Two experiments on the MRI data were performed. First, we used regularly-distributed nodes, and then, node distribution was manually adapted to an area of interest.

Regularly distributed nodes. Parameters for the compared algorithms were set to the same values used for the analytic flows experiment. Table 3(right half) shows the error metrics for the methods. Ours underperformed compared methods, but our results were close to Rueckert’s method. Next, we describe how results were improved by concentrating nodes around the heart’s region.

Adapted registration. In Figure 4(a), we show a polygonal mask manually created on the first frame of a heart sequence. This mask labels the area of

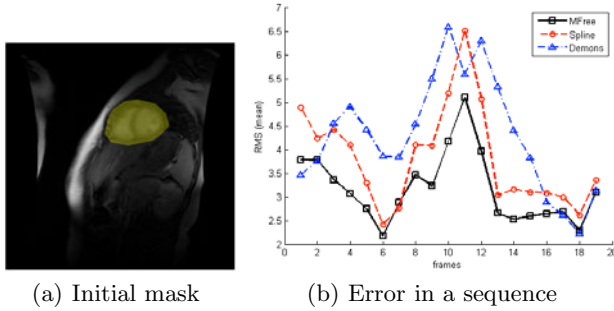


Fig. 4. Adapted computation for cardiac motion. (a) Manual initialization of a computation mask. (b) Registration RMS in the sequence for a patient.

Table 3. Node distribution scheme

Method	Regularly distributed			Adaptively distributed		
	RMS mean	RMS variance	CC mean	RMS mean	RMS variance	CC mean
B-Spline	1.45	1.08	99.75	3.72	4.08	99.75
Demons	1.27	0.83	99.80	4.24	3.42	93.95
Meshless	1.49	1.07	99.73	3.02	2.83	98.15

interest. Also, the deformation field in the unmasked region was not used for error evaluation. The initial mask was propagated to the remaining frames using the estimated deformation field. As the sequence is relatively short, the mask was reliably kept around the heart.

For both Rueckert’s and Demons methods, the evaluation was also restricted to the masked area. For our meshless method, the registration process was adapted by removing the nodes outside the mask. Removing nodes helped reduce computation costs significantly allowing for increased node density, by reducing both the space between them and their scales by half. Furthermore, rather than initializing a regular control-points grid for each frame [1], we simply propagated the nodes to the next frame. Table 3 (left half) shows a comparison of the three methods. Here, our method outperformed the other two on almost all metrics. We also observed that after adapting the evaluation, Rueckert’s method scored slightly higher than Demons. In Figure 4(b), we show the RMS for registration performed by the three methods on the sequence for a patient. The RMS peak was due to the heart’s contraction.

5 Conclusions

We proposed a meshless method for nonrigid image registration in which the deformation field is locally modeled using monomials, and then a global

deformation field is obtained by blending together the local deformation fields. For regularizing the deformation, we designed an energy function that penalizes inconsistencies between local fields. Quantitative and qualitative evaluations were performed on analytic fluid images, and on cardiac MRI images. Future work includes a comprehensive evaluation of the parameters' effects on registration results, and an extensive comparison to other registration methods.

References

1. Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., Hawkes, D.: Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. on Med. Imag.* 18, 712–721 (1999)
2. Hansen, M., Larsen, R., Lyngby, D., Glocker, B., Navab, N., München, G.: Adaptive parametrization of multivariate B-splines for image registration. In: *CVPR*, pp. 1–8 (2008)
3. Buhmann, M.: Radial basis functions: theory and implementations. Cambridge Monographs on Applied and Computational Mathematics, vol. 12 (2003)
4. Liu, G.R.: Mesh free methods: moving beyond the finite element method, 2nd edn. CRC, Boca Raton (2009)
5. Kobbelt, L., Botsch, M.: A survey of point-based techniques in computer graphics. *Computers & Graphics* 28, 801–814 (2004)
6. Makram-Ebeid, S., Somphone, O.: Non-rigid image registration using a hierarchical partition of unity finite element method. *ICCV* 510, 7 (2007)
7. Wang, X., Metaxas, D., Chen, T., Axel, L.: Meshless deformable models for LV motion analysis. In: *CVPR*, pp. 1–8 (2008)
8. Rohde, G., Aldroubi, A., Dawant, B.: The adaptive bases algorithm for intensity-based nonrigid image registration. *IEEE Trans. on Med. Imag.* 22, 1470–1479 (2003)
9. Chen, T., Kim, S., Goyal, S., Jabbour, S., Zhou, J., Rajagopal, G., Haffty, B., Yue, N.: Object-constrained meshless deformable algorithm for high speed 3D nonrigid registration between CT and CBCT. *Medical Physics* 37, 197 (2010)
10. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and simple calculus on tensors in the Log-Euclidean framework. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3749, pp. 115–122. Springer, Heidelberg (2005)
11. Carlier, J.: Second set of fluid mechanics image sequences. In: European Project 'Fluid image analysis and description', FLUID (2005), <http://www.fluid.irisa.fr/>
12. Andreopoulos, A., Tsotsos, J.K.: Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI. *Medical Image Analysis* 12, 335–357 (2008)
13. Kroon, D.J., Slump, C.H.: MRI modality transformation in demon registration. In: *ISBI*, Piscataway, NJ, USA, pp. 963–966. IEEE, Los Alamitos (2009)

Robust Rigid Shape Registration Method Using a Level Set Formulation

Muayed S. Al-Huseiny, Sasan Mahmoodi, and Mark S. Nixon

University of Southampton, School of Electronics and Computer Science, UK
{mssah07r, sm3, msn}@ecs.soton.ac.uk

Abstract. This paper presents a fast algorithm for robust registration of shapes implicitly represented by signed distance functions (SDFs). The proposed algorithm aims to recover the transformation parameters (scaling, rotation, and translation) by minimizing the dissimilarity between two shapes. To achieve a robust and fast algorithm, linear orthogonal transformations are employed to minimize the dissimilarity measures. The algorithm is applied to various shape registration problems, to address issues such as topological invariance, shape complexity, and convergence speed and stability. The outcomes are compared with other state-of-the-art shape registration algorithms to show the advantages of the new technique.

1 Introduction

Shape registration can be viewed as the result of a point-wise transformation between an observation and a reference shape. It is a fundamental task used to match two or more shapes taken, for example, at different times, from different viewpoints, or from different scenes. Virtually all large systems which evaluate images require the registration or a closely related operation as an intermediate step [1]. Shape registration is an essential requirement shared among many computer vision domains and applications, such as, pattern recognition, remote sensing, medical image analysis, and computer graphics to name a few.

The quality of registration is controlled using a similarity/dissimilarity measure. Also, the representation of the shape plays a crucial role in the registration process, and can significantly influence the overall performance of the registration algorithm. Active contours [2], Fourier descriptors [3] and active shapes models [4] are among the methods using explicit representations to describe arbitrary shapes. Although, these representations are powerful enough to capture a certain number of local deformations, they require a large number of parameters to deal with important shape deformations [5]. Non-parametric shape representations such as the signed distance functions (SDFs), are becoming a more popular choice, due to their implicit handling of various shape topologies, and the simple extension to describe higher dimensions than curves and surfaces [5].

Contour-based registration methods [6,7] are among the techniques used widely in shape registration, due to their fast convergence. These techniques however require point correspondence for the boundary of the shapes. However,

contour-based methods fall short, if two shapes to be registered have different Euler numbers.

Gradient descent-based registration techniques [5,8,9] widely used with segmentation applications are mostly characterized by low speed due to their iterative nature, instability and convergence to local minima, difficulty in implementation due to the need to tune the time step and stopping parameters for each transformation, and the limited extent of transformations these techniques can handle.

This paper presents a level set based shape registration algorithm. The algorithm proposed here employs linear transformation and shape moments to compute the parameters individually. We show that the registration technique presented here is robust, fast, and suitable for a wide range of registration problems with shapes' complexities. The results presented here are compared with state-of-the-art registration algorithms in the literature.

In the rest of the paper, we state the transformation problem in section 2, describe the proposed algorithm in section 3, present the experimental results in section 4, and conclude the paper in section 5.

2 The Statement of the Problem

Let $\phi_p(x, y) : \Omega \rightarrow \Re$ and $\phi_q(x, y) : \Omega \rightarrow \Re$ denote Lipschitz functions representing SDFs of shapes $p(x, y)$ and $q(x, y)$. These functions are defined as,

$$\phi_P(x, y) = \left\{ \begin{array}{ll} D_E((x, y), P), & (x, y) \in I_P, \\ -D_E((x, y), P), & (x, y) \in \Omega - I_P, \end{array} \right\} \quad (1)$$

where D_E represents the minimum Euclidean distance between the shape boundary I_P and each point in the domain Ω .

Parameters s , θ , T_x and T_y representing scaling, rotation, and translations in x and y directions respectively are required to transform ϕ_q to minimize the distance between ϕ_p and the transformed ϕ_q , i.e.:

$$(\hat{\theta}, \hat{s}, \hat{T}_x, \hat{T}_y) = \underset{\theta, s, T_x, T_y}{\operatorname{argmin}} \int \int |\phi_p(x, y) - \phi_q(sR_\theta(x + T_x, y + T_y))|^2 dx dy. \quad (2)$$

where,

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

The minimization of Eq 2 leads to a set of non-linear equations with respect to the desired parameters as discussed in [5]. The algorithm minimizing the distance measure (2) is slow to converge, can fall into local minima and requires continuously tuned parameters for smooth convergence [8]. The objective of this paper is therefore to propose a robust algorithm minimizing (2) and avoiding local minima with fast convergence, and no requirement for parameter tuning.

3 Shape Registration

3.1 Rotation

In order to find the optimal angle for rotation we conveniently employ polar coordinates. We use the notion that a rotation in Cartesian domain is displacement in polar domain [10]. Here we employ the translation invariance of the shapes' SDFs,

$$\hat{\phi}_p(x, y) = \phi_p(x - p_x, y - p_y), \tag{3}$$

$$\hat{\phi}_q(x, y) = \phi_q(x - q_x, y - q_y), \tag{4}$$

where (p_x, p_y) and (q_x, q_y) are respectively the centroids of ϕ_p and ϕ_q . A simple and efficient algorithm [11] is used to map $\hat{\phi}_p(x, y)$ and $\hat{\phi}_q(x, y)$ to polar coordinates, to obtain $\hat{\phi}_p(\rho, \omega)$ and $\hat{\phi}_q(\rho, \omega)$, so that: $x = \rho \cos \omega$, and $y = \rho \sin \omega$. These 2D centralized SDFs contain some redundancy in terms of their radial parameterization ρ which has no impact on the angle difference between the two SDFs. In order to remove the dependency of $\hat{\phi}_p(\rho, \omega)$ and $\hat{\phi}_q(\rho, \omega)$ on ρ , we integrate $\hat{\phi}_p(\rho, \omega)$ and $\hat{\phi}_q(\rho, \omega)$ with respect to ρ according to (5) and (6). We also notice that this dependency removal of ρ increases the computational speed.

$$\tilde{\phi}_p(\omega) = \int_{\rho} \hat{\phi}_p(\rho, \omega) d\rho, \tag{5}$$

$$\tilde{\phi}_q(\omega) = \int_{\rho} \hat{\phi}_q(\rho, \omega) d\rho. \tag{6}$$

Let $\bar{\phi}_p$ denote a normalized instance of $\hat{\phi}_p$, i.e.:

$$\bar{\phi}_p(\omega) = \frac{\hat{\phi}_p(\omega)}{\int_{\omega} \hat{\phi}_p(\omega) d\omega}. \tag{7}$$

The unknown angle will be estimated by minimizing the dissimilarity measure in (8),

$$\begin{aligned} \int_{\omega} |\tilde{\phi}_q - \beta \bar{\phi}_p|^2 d\omega &= \int_{\omega} \left(|\tilde{\phi}_q - \beta \bar{\phi}_p|^T |\tilde{\phi}_q - \beta \bar{\phi}_p| \right) d\omega, \\ &= \int_{\omega} \left((\tilde{\phi}_q)^T \tilde{\phi}_q - \beta (\tilde{\phi}_q)^T \bar{\phi}_p - \beta \tilde{\phi}_q (\bar{\phi}_p)^T + \beta^2 (\bar{\phi}_p)^T \bar{\phi}_p \right) d\omega, \\ &= \int_{\omega} \left(|\tilde{\phi}_q|^2 - 2\beta \langle \tilde{\phi}_q \cdot \bar{\phi}_p \rangle + \beta^2 |\bar{\phi}_p|^2 \right) d\omega, \end{aligned} \tag{8}$$

where β is defined as, $\beta := \langle \tilde{\phi}_q \cdot \bar{\phi}_p \rangle$.

Since $|\bar{\phi}_p|^2 = 1$ (Eq 7), we have,

$$\int_{\omega} |\tilde{\phi}_q - \beta \bar{\phi}_p|^2 d\omega = \int_{\omega} |\tilde{\phi}_q|^2 d\omega - \beta^2. \tag{9}$$

Hence the minimization of (8) is achieved by maximizing β . The optimal rotation angle is therefore calculated by finding θ that maximizes β , or,

$$\hat{\theta} = \operatorname{argmax}_{\theta} \beta. \tag{10}$$

The maximum β is computed using the Fourier transform. Let the Fourier transform of $\hat{\phi}_q$ and $\hat{\phi}_p$ be respectively $\tilde{\psi}_q(\xi)$ and $\tilde{\psi}_p(\xi)$, such that,

$$\tilde{\psi}_p(\xi) = \int_{\xi} \bar{\phi}_p(\omega) e^{-i(\omega\xi)2\pi} d\omega. \tag{11}$$

$$\tilde{\psi}_q(\xi) = \int_{\xi} \tilde{\phi}_q(\omega) e^{-i(\omega\xi)2\pi} d\omega, \tag{12}$$

Therefore, by using Parseval’s theorem, we can write:

$$\begin{aligned} \beta(\theta) &= \left\langle \tilde{\phi}_q(\omega) \cdot \bar{\phi}_p(\omega + \theta) \right\rangle = \int_{\omega} \left(\tilde{\phi}_q(\omega) \bar{\phi}_p(\omega + \theta) \right) d\omega \\ &= \int_{\xi} \left(\tilde{\psi}_q(\xi) \tilde{\psi}_p^*(\xi) e^{i(\xi\theta)2\pi} \right) d\xi, \end{aligned} \tag{13}$$

where $(*)$ denotes the complex conjugate. Hence, $\hat{\theta}$ is computed as:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \beta = \operatorname{argmax}_{\theta} \int_{\xi} \left(\tilde{\psi}_q(\xi) \tilde{\psi}_p^*(\xi) e^{i(\xi\theta)2\pi} \right) d\xi. \tag{14}$$

3.2 Scale

We use the geometric moments of SDFs to characterize the shapes’ features to calculate the scaling parameter. It can be shown that if one shape is a scaled version of another then the corresponding SDFs are proportional to the scale factor [5],

$$s\hat{\phi}_p(x, y) = \hat{\phi}_q(sx, sy), \tag{15}$$

where s is the scale parameter. The geometrical moments of the reference SDF $\hat{\phi}_p$ and the observed SDF $\hat{\phi}_q$ are computed as:

$$M_m^{\hat{q}} = \int \int \left(\sqrt{x^2 + y^2} \right)^m \hat{\phi}_q(x, y) dx dy, \tag{16}$$

$$M_m^{\hat{p}} = \int \int \left(\sqrt{x^2 + y^2} \right)^m \hat{\phi}_p(x, y) dx dy, \tag{17}$$

where m represents the degree of the moment. Substituting Eq.15 into Eq.17, we arrive at Eq.18:

$$M_m^{\hat{p}} = \frac{1}{s} \int \int (\sqrt{x^2 + y^2})^m \hat{\phi}_q(sx, sy) \, dx dy. \tag{18}$$

By changing variables, $X = sx$, and $Y = sy$, Eq.18 can be written as,

$$\begin{aligned} M_m^{\hat{p}} &= \int \int \frac{(\sqrt{X^2 + Y^2})^m}{s^m} \hat{\phi}_q(X, Y) \frac{dX \, dY}{s^2}, \\ &= \frac{1}{s^{(m+3)}} \int \int (\sqrt{X^2 + Y^2})^m \hat{\phi}_q(X, Y) \, dX \, dY, \\ &= \frac{1}{s^{(m+3)}} M_m^{\hat{q}}. \end{aligned} \tag{19}$$

Let E be the error defined in Eq.20:

$$E = \sum_m |M_m^{\hat{q}} - s^{m+3} M_m^{\hat{p}}|^2. \tag{20}$$

Since Eq.20 is non-linear with respect to variable s , with a change of variable, the above non-linear least squares problem is reduced to a linear one, i.e.:

$$\acute{E} = \sum_m \left| \log \left(\frac{M_m^{\hat{q}}}{M_m^{\hat{p}}} \right) - (m + 3) \log s \right|^2. \tag{21}$$

Hence the optimal scale parameter \hat{s} is estimated by minimizing \acute{E} :

$$\hat{s} = \underset{s}{\operatorname{argmin}} \acute{E}. \tag{22}$$

It should be noted that the use of Chebyshev or Zernike moments leads to a non-linear least squares problem whose minimization is more difficult and demanding than the current method proposed here. In the case of Chebychev and Zernike moments, the non-linearity does not reduce to a linear problem by using a change of variables.

3.3 Translation

Using the scaling and rotation parameters calculated in 3.1 and 3.2, we optimize Eq.2 to calculate T_x and T_y :

$$\phi_p(x, y) = \phi_q(x - T_x, y - T_y). \tag{23}$$

By employing the same method explained in 3.1, the translation parameters are calculated as:

$$\begin{aligned} \left[\acute{T}_x \ \acute{T}_y \right] &= \operatorname{argmax}_{T_x, T_y} \langle \phi_q, \check{\phi}_p \rangle, \\ &= \operatorname{argmax}_{T_x, T_y} \int_{\omega_x} \int_{\omega_y} (\psi_q(\omega_x, \omega_y) \check{\psi}_p(\omega_x, \omega_y) e^{i(T_x \omega_x + T_y \omega_y) 2\pi}) \, d\omega_x d\omega_y, \end{aligned} \tag{24}$$

where $\check{\phi}_p$ is a normalized instance of ϕ_p , \check{T}_x and \check{T}_y represent the estimated optimal translation parameters, $\psi_q(\omega_x, \omega_y)$ and $\check{\psi}_p(\omega_x, \omega_y)$ represent respectively the 2D Fourier transform of ϕ_q and $\check{\phi}_p$, and ω_x and ω_y are the spatial frequencies.

Remark: Since images in practice are in discrete domain, we employ Fast Fourier transform(FFT) instead of continuous Fourier transforms employed in this section. We are therefore required to modify the definition of SDFs to cope with the periodicity property imposed by FFT.

Let Ω be the image domain. This domain is partitioned by the shape perimeter into two regions, the shape interior (convex hull) I_P and the background, and let $\check{\phi} : I_P \rightarrow \mathbb{R}^+$ be a Lipschitz function that represents the distance transform for the interior of the shape P . This is expressed in (25):

$$\check{\phi}_P(x, y) = \begin{cases} D_E((x, y), P), & (x, y) \in I_P, \\ 0, & (x, y) \in \Omega - I_P, \end{cases} \quad (25)$$

This modified SDF representation is induced by the periodicity requirements of the FFT used in Section 3.

4 Results and Discussions

In the following we present a set of examples, each of which is intended to show the advantage of the shape registration method presented here over other known registration methods in a particular shape registration problem. For better demonstration, the figures show the contours of the observed and reference shapes before and after registration.

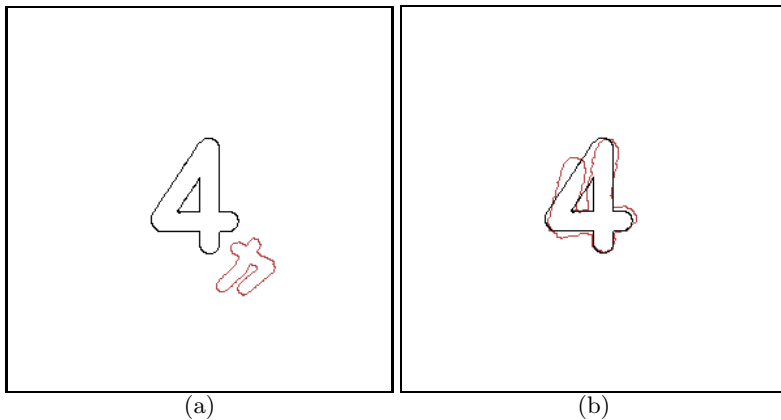


Fig. 1. The registration of shapes with different topologies. (a)The observation is a transformed open '4' with Euler characteristic 1, while the reference is a closed one with Euler characteristic 0. (b)The two shapes superimposed optimally using the proposed approach.

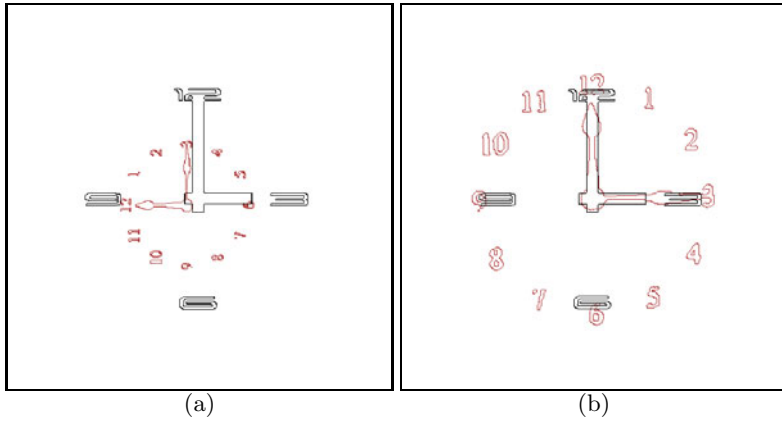


Fig. 2. The registration of shapes with different number of shape components. (a)The reference shape has clock hands and compass point indicators, whereas the observed shape has smaller hands and more conventional indicators. (b)The result after applying our technique.

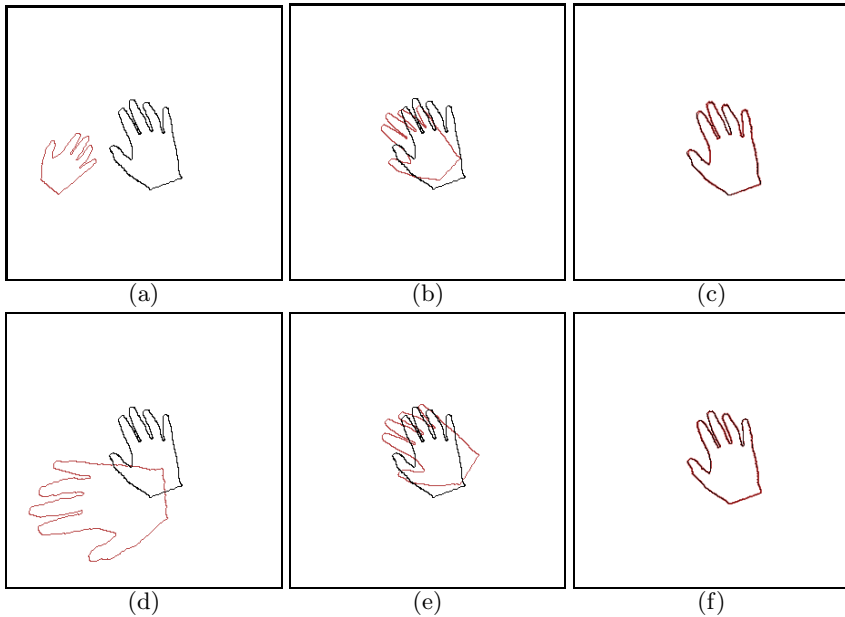


Fig. 3. The registration of identical shapes with synthetic transformations. (a and d) The observed shape is a replica of the reference shape with $\theta=-60$, $s=0.7$, $T_x=-90$, $T_y=20$, and $\theta=75$, $s=1.5$, $T_x=-50$, $T_y=50$ respectively. (b and e) The registration using gradient descent method [5], notice the inaccuracy of registration caused by the local minima issue. (c and f) The registration using the new method, the shapes are almost perfectly registered.

– **Contour Methods:**

In the first example we use two shapes with different Euler characteristics. Such shapes have completely different contours (topologies) and hence the contour based methods fail to calculate the correct transformation parameters, also it is hard to establish automatic contour points correspondences. In Figure 1, the observed shape is an open number four with Euler characteristic one, and the reference shape is a closed four with Euler characteristic zero. These shapes have different topologies, however they have been correctly registered using the registration approach proposed in this paper.

In the second example we register two complex shapes each having different number of components. The employment of contour methods to register such shapes for example by registering the individual objects in the observed shape to their counterparts in the reference shape may do partially, wherein the objects with no counterparts remain unregistered. In Figure 2 the observed shape is a clock face with conventional indicators while the reference shape has compass point indicators, this example demonstrates the registration algorithm proposed here can register two shapes, even if there is no one to one correspondence among components forming the shapes.

– **Gradient Descent Methods:**

In the third example we test a gradient descent based method proposed by Paragios *et al.* [5] using various transformation parameters. Figure 3 shows the local minima problem associated with these methods which leads to inaccurate

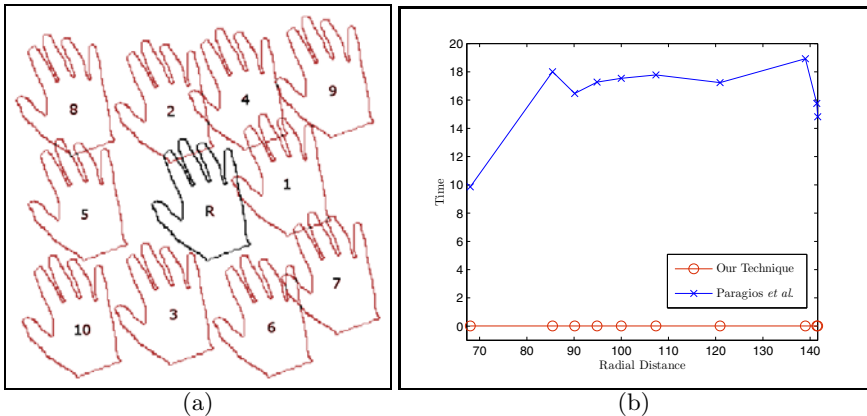


Fig. 4. The registration of similar shapes with synthetic translations. (a) The observed shape is a replica of the reference shape R with ten different translations numbered ascendingly according to the radial distances from R. (b) A plot of the convergence time for each translation, it shows that the average convergence time for the gradient descent method is (16) seconds, while for our technique the average time is (0.012) seconds.

registration. In the figure, two different sets of transformation parameters are used to transform a shape and then gradient descent method is used to find these parameters. This is compared to the result of the same registration problem using our technique with two pass registration for more accuracy.

In the fourth example we study the issue of speed. The observed shape is a replica of the reference shape having the same rotation angle and scale. We employ ten randomly chosen translations for the observed shape. Both our algorithm and the gradient descent method cited in [5] are used to recover the translation parameters. The choice to test the convergence time for translation parameters is induced by the fact that translation is a linear transformation. Figure 4 shows the registration of the observed shape with ten arbitrarily chosen translations. The shapes are numbered according to their radial distance ($r = \sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}$) to the reference shape, where (x_r, y_r) and (x_o, y_o) are respectively the center coordinates of the reference and the observed shapes. In Figure 4(b), we plot the convergence time for both techniques against the radial distance r . From this plot we observe that the proposed method is faster and the speed is almost constant for all translations. The convergence time depends only on the size of the shape domain. The gradient descent method [5] on the other hand has variable speed of convergence. This depends on the actual translation and the magnitude of the gradient in each direction, such factors justify the non linear convergence time noticed in this figure.

5 Conclusions

This paper presents a shape registration algorithm which uses a modified signed distance function to represent the shapes. The proposed algorithm estimates the parameters using closed form expressions. This algorithm exploits Parseval's theorem to estimate the rotation and the translation parameters, and uses the geometric moments to estimate the scale parameters. The registration technique has been tested on shapes selected to demonstrate successful extraction and performance. Our method is robust in registering complex shapes and shapes with various topologies which can not be registered using contour based methods. The experimental results show that our registration algorithm is fast, accurate, stable, and does not fall into local minima.

Acknowledgment. This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886 and PinView Project, 216529. This publication only reflects the authors' views.

References

1. Brown, L.G.: A survey of image registration techniques. *ACM Computing Surveys* 24, 325–376 (1992)
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1, 321–331 (1988)

3. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. *IEEE Transactions on Computers* 21, 269–281 (1972)
4. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. *Computer Vision and Image Understanding* 61, 38–59 (1995)
5. Paragios, N., Rousson, M., Ramesh, V.: Non-rigid registration using distance functions. *Computer Vision and Image Understanding* 89, 142–165 (2003)
6. Marques, J.S., Abrantes, A.J.: Shape alignment – optimal initial point and pose estimation. *Pattern Recognition Letters* 18, 49–53 (1997)
7. Markovsky, I., Mahmoodi, S.: Least-squares contour alignment. *IEEE Signal Processing Letters* 16, 41–44 (2009)
8. Cremers, D., Osher, S.J., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision* 69, 335–351 (2006)
9. Vemuri, B.C., Ye, J., Chen, Y., Leonard, C.M.: Image registration via level-set motion: applications to atlas-based segmentation. *Medical Image Analysis* 7, 1–20 (2003)
10. Casasent, D., Psaltis, D.: Position, rotation and scale invariant optical correlation. *Applied Optics* 15, 1795–1799 (1976)
11. Mukundan, R.: A comparative analysis of radial-tchebichef moments and zernike moments. In: *British Machine Vision Conference 2009* (2009)

A Meshless Method for Variational Nonrigid 2-D Shape Registration

Wei Liu and Eraldo Ribeiro

Computer Vision and Bio-Inspired Computing Laboratory
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, U.S.A.

Abstract. We present a method for nonrigid registration of 2-D geometric shapes. Our contribution is twofold. First, we extend the classic chamfer-matching energy to a variational functional. Secondly, we introduce a meshless deformation model that can adapt computation to the shape boundary. In our method, 2-D shapes are implicitly represented by a distance transform, and the registration error is defined based on the shape contours' mutual distances. Additionally, we model global shape deformation as an approximation blended from local fields using partition-of-unity. The deformation field is regularized by penalizing inconsistencies between local fields. This representation can be adaptive to the shape's contour, leading to registration that is both flexible and efficient. Finally, shape registration is achieved by minimizing a variational chamfer-energy functional combined with the consistency regularizer using an efficient quasi-Newton algorithm. We demonstrate the effectiveness of our registration method on a number of experiments.

1 Introduction

Registering 2-D shapes that have been deformed by nonlinear mappings is a challenging problem that has applications in many areas including medical imaging [1] and shape recognition [2]. Similarities can be drawn between shape registration and general nonrigid image-registration problems, with variational methods marking the state-of-the-art for nonrigid image registration. On the other hand, current variational shape-registration methods are sensitive to shape noise and occlusion. In this paper, we extend the work in [3,4], and propose a robust and efficient variational shape-registration method using an implicit distance transform representation and a meshless deformation model.

Shape registration is an ill-posed problem as there can be many ambiguous solutions. Similarly to nonrigid image registration [5], the ill-posedness in shape registration methods is often addressed by regularizing solutions through statistical [6] or variational priors [3,4]. In contrast with image registration, where texture information may be abundant, shape registration often deals with images containing very sparse signal, that can be highly sensitive to image noise. Although statistical priors help improve robustness, these priors are often tailored to different classes of shapes, requiring a separate class-specific training

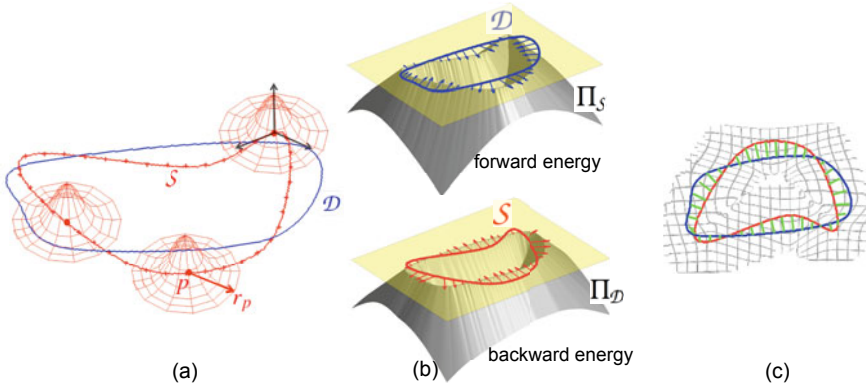


Fig. 1. Meshless shape registration. (a) Source (red curve) and target (blue curve) shapes. Nodes are placed along the contour. Three nodes are illustrated with their corresponding influence regions (Section 3). (b) Forward and backward registration error (Section 2). (c) Blended global deformation map and correspondence after registration.

stage. On the other hand, while variational methods make fewer assumptions about the shapes, these methods can be more sensitive to occlusion and noise.

Our focus in this paper is on variational shape-registration methods. In this class of methods, existing approaches differ in three main aspects [4]: shape representation, deformation model, and registration criterion. Implicit shape models can be obtained by considering a shape to be a distance-transform’s zero level set [3,4]. Advantages of using implicit representations include fewer model parameters, and easy extension to higher dimensions. Moreover, distance functions are redundant 2-D representations of 1-D shapes, and similar distance functions lead to similar shapes. Thus, contour registration can be achieved using traditional image-registration techniques [3,4]. Indeed, the underlining registration criterion can be simply the squared-difference of distance functions, and the deformation model can be non-parametric [3], or parametric as B-splines [4]. However, distance transforms are sensitive to shape noise, and its redundancy leads to unnecessary computation. These problems are only partially addressed [3,4] by limiting the registration around shape contours based on a proximity function.

In this paper, we adopt an implicit shape model based on distance functions, and address some of the above problems by removing the redundancy from both the registration criterion and the deformation model. First, we modify the registration criterion by considering distance errors at shape boundaries only. This criterion can be seen as a variational form of the classic chamfer-matching functional. As in chamfer matching, the proposed functional is robust to both spurious points and shape occlusions. Secondly, we propose a mesh-free deformation model to adapt registration around shape contours. In contrast to B-spline models that rely on a control-point grid with explicit connections, our meshless deformation model represents shape deformation by blending together local deformations using partition-of-unity [7]. These local deformations can be centered

at arbitrarily distributed nodes (particles), allowing us to model shapes of different topologies, and to handle irregular shape deformations. By aligning the nodes along the shape contour, we can remove the redundancy in the registration process. Since rigid shape alignment can be done using off-the-shelf methods such as shape context [1], mutual information [4], and chamfer matching [2], we assume that shapes are aligned beforehand using a rigid transformation, and focus ourselves on the nonrigid registration part (i.e., global-to-local approach [1]).

This paper is organized as follows. In Section 2, we review the general framework for nonrigid registration using distance functions [3,4], and then introduce our variational dissimilarity function. In Section 3, we introduce our meshless shape-deformation representation, and discuss the numerical minimization of the proposed dissimilarity functional. Section 4 shows registration results on both synthetic and real-world images.

2 Distance Functions and Nonrigid Registration

The goal of shape registration is to deform a source shape onto a target shape. This is achieved by searching for the best deformation field that minimizes a dissimilarity measure between the shapes. Formally, if \mathcal{S} and \mathcal{D} represent source and target shapes, respectively, and F is a dissimilarity measure between the two shapes, we seek for a warping field $\mathbf{u}(\mathbf{x})$ that satisfies the following equation:

$$\arg \min_{\mathbf{x}'} F(\mathcal{D}(\mathbf{x}'), \mathcal{S}(\mathbf{x}), \mathbf{x}'), \quad \mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x}), \quad (1)$$

where \mathbf{x} is a coordinate vector. The dissimilarity measure F usually depends on the shape model. In this paper, we implicitly represent a shape \mathcal{S} as the zero level set of its distance transform $\Pi_{\mathcal{S}}$ [3,4], where \mathcal{S} defines a partition of the image domain Ω . The model is given by:

$$\Pi_{\mathcal{S}} = \begin{cases} 0, & \mathbf{x} \in \mathcal{S} \\ +D_{\mathcal{S}}(\mathbf{x}) > 0, & \mathbf{x} \in R_{\mathcal{S}} \\ -D_{\mathcal{S}}(\mathbf{x}) < 0, & \mathbf{x} \in [\Omega - R_{\mathcal{S}}] \end{cases}, \quad (2)$$

where $D_{\mathcal{S}}$ is the minimum Euclidean distance between location \mathbf{x} and shape \mathcal{S} , and $R_{\mathcal{S}}$ is the region inside \mathcal{S} . Here, F can be defined as the squared-sum of distance transform differences, and registration is achieved by minimizing:

$$E(\mathbf{u}) = \underbrace{\int_{\Omega} N_{\delta}(\Pi_{\mathcal{D}} - \Pi_{\mathcal{S}})^2 d\mathbf{x}}_{\text{data term}} + \alpha \underbrace{\int_{\Omega} N_{\delta} (\|\nabla \mathbf{u}_x\|^2 + \|\nabla \mathbf{u}_y\|^2) d\mathbf{x}}_{\text{smoothness regularizer}}. \quad (3)$$

In (3), $\Pi_{\mathcal{S}}$ and $\Pi_{\mathcal{D}}$ are distance transforms of the source and target shapes, respectively. The proximity function N_{δ} limits the data-term evaluation to be near the shape's boundary, and the smoothness term penalizes for spatial variations in the estimated deformation field.

The above representation facilitates the use of existing nonrigid registration techniques to solve shape registration. However, two issues need to be considered. First, although similar distance functions lead to similar shapes, similar shapes may not necessarily produce similar distance functions. For example, a spurious point located far from the shape can offset the distance transform, leading to different 2-D representations. In other words, This implicit representation’s redundancy breaks the continuity between shapes and their representation domains. This argument is supported in [3] by observing that scaling affects distance functions. In fact, a scaling factor is estimated separately in [3], and shape noise is only partially addressed in [3,4] by using the proximity function N_δ . Secondly, registering 2-D distance functions leads to extra computation as deformation models register the whole image plane. The use of the proximity function [3,4] reduces these problems but the formulation becomes more complicated.

Next, we propose a dissimilarity measure by using a novel variational formulation of the chamfer-matching energy that does not use a proximity function.

2.1 Variational Chamfer-Matching Energy

When the source shape S is aligned with the target shape \mathcal{D} , the deformed shape $s(\mathbf{x} + \mathbf{u})$ will coincide with the zero level set of $\Pi_{\mathcal{D}}$, i.e., $s(\mathbf{x} + \mathbf{u}) \Pi_{\mathcal{D}} = 0$. Here, we represent shape S by a binary contour map, and enforce alignment between shapes by minimizing the squared sum $\int_{\Omega} |s(\mathbf{x} + \mathbf{u}) \Pi_{\mathcal{D}}|^2 d\mathbf{x}$, which corresponds to the classic chamfer-matching energy function [2]. However, this functional can be ill-posed. For example, the energy function will vanish for any deformation field that shrinks the source shape to a single point on shape \mathcal{D} . Similarly to *symmetric chamfer-matching energy* [8], we can address this problem by including a symmetric term that measures the distance-error between target and source shapes. Additionally, we compensate for scaling by dividing the distance-error by the contours’ length, and minimize the following functional:

$$E^d(\mathbf{u}) = \frac{1}{A} \left[\underbrace{\int_{\Omega} |s(\mathbf{x} + \mathbf{u}) \Pi_{\mathcal{D}}|^2 d\mathbf{x}}_{\text{forward energy } E^f} + \underbrace{\int_{\Omega} |\mathcal{D}(\mathbf{x}) \Pi_{s(\mathbf{x}+\mathbf{u})}|^2 d\mathbf{x}}_{\text{backward energy } E^b} \right], \tag{4}$$

where $A = \int_{\Omega} s(\mathbf{x} + \mathbf{u}) d\mathbf{x} \int_{\Omega} \mathcal{D}(\mathbf{x}) d\mathbf{x}$ is a normalizing factor. Since $E^d(\mathbf{u})$ is independent on the sign of Π , we will assume that $\Pi_s \geq 0$ and $\Pi_{\mathcal{D}} \geq 0$.

The registration error is directly measured using the shape contours without resorting to a proximity function as in [3]. As in chamfer matching, the usage of distance transform facilitates optimization by providing an energy gradient. For example, the gradient of the forward-energy term can be calculated as follows:

$$\frac{\partial E^f(\mathbf{u})}{\partial \mathbf{u}} = 2 \int_{\Omega} [s(\mathbf{x} + \mathbf{u}) \Pi_{\mathcal{D}}] \Pi_{\mathcal{D}} \frac{\partial s(\mathbf{x} + \mathbf{u})}{\partial \mathbf{x}} d\mathbf{x}. \tag{5}$$

Since s is a binary map, then $\Pi_{\mathcal{D}} \frac{\partial s(\mathbf{x}+\mathbf{u})}{\partial \mathbf{x}} = \frac{\partial \Pi_{\mathcal{D}}}{\partial \mathbf{x}} s(\mathbf{x} + \mathbf{u})$ and $s(\mathbf{x} + \mathbf{u})s(\mathbf{x} + \mathbf{u}) = s(\mathbf{x} + \mathbf{u})$. Substituting these identities into (5), we have:

$$\frac{\partial E^f(\mathbf{u})}{\partial \mathbf{u}} = 2 \int_{\Omega} [s(\mathbf{x} + \mathbf{u}) \Pi_{\mathcal{D}}] \frac{\partial \Pi_{\mathcal{D}}}{\partial \mathbf{x}} d\mathbf{x}. \tag{6}$$

For the backward-energy term in (4), its derivative involves calculating the distance transform of the deformed source shape, i.e., $\Pi_{s(\mathbf{x}+\mathbf{u})}$. Fortunately, by substituting variables, Equation 4 can be re-written as $\int_{\Omega} |\mathcal{D}(\mathbf{x}) \Pi_{s(\mathbf{x}+\mathbf{u})}|^2 d\mathbf{x} = \int_{\Omega} |\mathcal{D}(\mathbf{x} - \mathbf{u}) \Pi_{s(\mathbf{x})}|^2 d\mathbf{x}$, and then expanded as we did in Equation 6 to have:

$$\frac{\partial E^b}{\partial \mathbf{u}} = -2 \int_{\Omega} [\mathcal{D}(\mathbf{x} - \mathbf{u}) \Pi_s] \frac{\partial \Pi_s}{\partial \mathbf{x}} d\mathbf{x} = -2 \int_{\Omega} [\mathcal{D}(\mathbf{x}) \Pi_{s(\mathbf{x}+\mathbf{u})}] \frac{\partial \Pi_{s(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{x}} d\mathbf{x}. \tag{7}$$

In the final step of Equation 7, we have substituted $\mathcal{D}(\mathbf{x} - \mathbf{u})$ by $\mathcal{D}(\mathbf{x})$ to keep the target shape unchanged during registration. Note that, in the chamfer-matching energy functional in (4), we could also use the \mathbf{L}^1 norm instead of the squared-sum (i.e., \mathbf{L}^2 norm). However, our experiments showed that the \mathbf{L}^1 norm is more sensitive to local minima, and leads to slower minimization convergence. This observation echoes a similar finding in classic chamfer matching [2].

Given the above chamfer-matching energy, different regularizers and deformation representations can be used for shape registration. In fact, the second-order regularizer of Equation 3 will still be valid when combined with our data term. Alternatively, the B-Spline representation in 4 can also be used. However, nonparametric estimation may not handle some large deformations [4], while spline-based models are limited by the need to explicitly maintain a regular control-point grid (mesh) and connections. Next, we address some of these issues by adopting a meshless representation that approximates the shape’s deformation field by blending together local polynomial models using partition-of-unity.

3 Meshless Deformation Model

A limitation of B-spline models is their reliance on an explicit-connected control-point grid (i.e., mesh). Inspired by recent developments in computer graphics [9] and mechanical engineering [10], on building shape functions of arbitrary topology from scattered sample points, we propose a meshless deformation model for shape registration. Although there are meshless shape-deformation models based on thin-plate splines and radial basis functions (RBFs) [1], they are less accurate than polynomial-based representations, as radial basis functions cannot exactly represent polynomial deformations (lack of reproducibility) [10]. In our method, local deformation fields are modeled around scattered nodes (particles) as polynomials, and then blended together into a global deformation field using partition-of-unity. In the following subsections, we first introduce the local deformation model, and then explain how to blend them into a global model.

3.1 The Node’s Influence Domain

We commence by modeling shape deformation around scattered nodes using polynomial approximation. These nodes can be placed along the shape’s contour. As polynomials lack compact support, the approximation is restricted to the node’s proximity, a region called the node’s *influence domain*. The influence domain also serves to limit the interaction range between neighboring nodes. Let us define the influence domain M around a node p as a disk of radius r_p (ball in 3-D¹). M can be modeled by a weighting function $w_p(\mathbf{x})$ with local support. Various types of weighting functions exist [10]. We define $w_p(\mathbf{x})$ as:

$$w_p(\mathbf{x}) = \begin{cases} \alpha_p \exp\left(-\kappa \frac{\|\mathbf{p}-\mathbf{x}\|^2}{r_p^2}\right) & , \mathbf{x} \in M \\ 0 & , \mathbf{x} \notin M \end{cases}, \tag{8}$$

where \mathbf{p} denotes the coordinate vector of node p , $\kappa = \frac{1}{9}$, and $\alpha_p \in (0, 1]$ is the node’s predefined influence factor in the final global blending. Thus, a node p is defined by three parameters $(\mathbf{p}, r_p, \alpha_p)$, i.e., its spatial position, the radius (scale) of its influence domain, and its influence factor. Note that while the weighting function in (8) is a radial function, its usage is different from previous RBF models such as thin-plate splines [1]. Here, RBFs are used for blending the local polynomial models, instead of directly representing the shape deformation.

3.2 Local Approximation Model around a Node

The local deformation field $\mathbf{u}^p = (u, v)$ around node p can be expressed as a linear combination of monomials $x^s y^t$ as follows:

$$u(\mathbf{x}) = \sum_{s,t=0}^{s,t=m} a_{s,t} x^s y^t \quad \text{and} \quad v(\mathbf{x}) = \sum_{s,t=0}^{s,t=m} b_{s,t} x^s y^t, \tag{9}$$

In other words, the local deformation field $\mathbf{u}_p(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x})]^T$ is represented as a linear combination of monomial basis functions $\phi^T(\mathbf{x}) = [1, x, y, xy, x^2, y^2, \dots, x^m y^m]$ with coefficient vector $\mathbf{d}_p = [a_{0,0}, b_{0,0}, \dots, a_{m,m}, b_{m,m}]^T$. The sequence of monomials in ϕ is arranged in a Pascal-triangle manner [10].

3.3 Blending Local Models into a Global Deformation Field

Once the local deformation models are at hand, the deformation at a point \mathbf{x} is obtained by blending local fields of nodes around \mathbf{x} , that contain \mathbf{x} in their influence domains. These nodes are called the *support domain* [10] of \mathbf{x} , denoted by $N_{\mathbf{x}} = \{p \mid \mathbf{x} \in M\}$. The blended global-deformation field is given by:

$$\mathbf{u}(\mathbf{x}) = \sum_{p \in N_{\mathbf{x}}} r_p(\mathbf{x}) \mathbf{u}_p(\mathbf{x}), \quad \text{with} \quad r_p(\mathbf{x}) = \frac{w_p(\mathbf{x})}{\sum_{p' \in N_{\mathbf{x}}} w_{p'}(\mathbf{x})}. \tag{10}$$

¹ A 3-D extension is straightforward.

Here, $r_p(\mathbf{x})$ ensures the partition-of-unity (PU), i.e., nodes' contributions at \mathbf{x} must add to one. This blending scheme is equivalent to the Arsigny's polyaffine model [11], and Makram-Ebeid's meshless model [7]. Next, we introduce a novel regularizer to penalize undesired fluctuations in the estimated deformation field.

3.4 Consistency Enforcement

We have shown that the global deformation can be obtained by blending local deformation fields using Equation 10. In spline-based methods [4], estimated deformation fields are consistent across the control points, and regularization is obtained using Sobolev's norm that penalizes the deformation field's spatial variation. In our method, global deformation fluctuations lead to inconsistencies among local deformation fields. As a result, we penalize the local deformation's *spatial inconsistency*, leading to simpler optimization procedures, as well as to a regularizer that is not biased towards the deformation field's lower-order fluctuations, provided that the fluctuation itself is spatially consistent.

Consistency between two local deformation fields, \mathbf{u}_p and \mathbf{u}_q , can be measured from parameters \mathbf{d}_p and \mathbf{d}_q . However, \mathbf{u}_p and \mathbf{u}_q lie on different local coordinate systems, and therefore need to be aligned. Aligning the basis functions ϕ by $\Delta\mathbf{x} = [\delta x, \delta y]$ leads to:

$$\begin{aligned} \phi(\mathbf{x} + \Delta\mathbf{x}) &= [1, x + \delta x, y + \delta y, (x + \delta x)(y + \delta y), \dots, (y + \delta y)^m]^\top \\ &= \mathbf{S}^\top(\Delta\mathbf{x})\phi(\mathbf{x}), \end{aligned} \tag{11}$$

where $\mathbf{S}^\top(\Delta\mathbf{x})$ is the linear *basis-shifting-operator*. Therefore, shifting the local coordinate system leads to shifted polynomial coefficients, and the local deformation consistency between two nodes p and q can be defined as:

$$E_{p,q}^c = [\mathbf{S}'(\mathbf{p} - \mathbf{q})\mathbf{d}_q - \mathbf{d}_p]^\top [\mathbf{S}'(\mathbf{p} - \mathbf{q})\mathbf{d}_q - \mathbf{d}_p]. \tag{12}$$

Here, an equivalent shift operator $\mathbf{S}'(\mathbf{p} - \mathbf{q})$ is created by duplicating and shifting the elements of the basis-shifting-operator. For N nodes, the global consistency regularizer is obtained by penalizing the average pairwise inconsistency in [12]:

$$E^c = \frac{1}{N} \sum_p \left[\sum_{q \in N_p} w_q(\|\mathbf{p} - \mathbf{q}\|) E_{p,q}^c \right]. \tag{13}$$

3.5 Quasi-Newton Registration Algorithm

We now combine both the chamfer and consistency energies into the following functional minimization problem:

$$\mathbf{d}_p = \arg \min_{\mathbf{d}_p} [E^d(\mathbf{u}) + \lambda E^c], \tag{14}$$

where parameter λ defines the relative importance of the deformation's spatial consistency. Minimizing [14] can be efficiently achieved using gradient

descent [4,2]. In this paper, we use a quasi-Newton method [12] for its improved convergence speed. The calculation of the required partial derivatives $\frac{\partial E^d(\mathbf{u})}{\partial \mathbf{d}_p}$ and $\frac{\partial E^c}{\partial \mathbf{d}_p}$ is straightforward following Equations [6,7,12], and [13].

Using the derived gradients, we implemented an optimization algorithm based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [12]. At each iteration of the algorithm, the source shape is first warped using the deformation field reconstructed from local field parameters (Equation [10]), and then its distance transform $\Pi_{S(\mathbf{x}+\mathbf{u})}$ is updated. Both the destination shape \mathcal{D} and its distance transform $\Pi_{\mathcal{D}}$ remain constant. We experimentally determined the search step's lower bound to be 0.2, and that helps avoid getting trapped in local minima. Additionally, we handle large shape deformations by adopting the hierarchical multi-scale registration strategy used in [4] (i.e., a coarse-to-fine approach).

4 Experiments

We tested our method on the Brown University shape dataset [13], and on a cell morphing sequence. Due to the lack of ground truth for shape registration, we demonstrate the results visually in a similar way as in [4,3,1]. For the Brown university dataset, the images were first normalized to 150×150 sizes, and the shapes were globally aligned beforehand using the rigid registration method implemented in [5]. Then, an initially regular grid of nodes was adapted to the shapes by removing nodes that do not overlap with the shape contour. This adaptation reduced the execution time for about 80 percent on average. In the hierarchical registration algorithm, the space between nodes was 5 pixels at the finest scale, and the node's radii were 12.5 pixels, i.e., each node interacted with around 20 neighbors. For all shapes, we selected the regularizer weight $\lambda = 10$. Figure 2 shows registration results obtained using our method. As in [1], we selected three different shapes (person², hand, and fish), and quantitatively evaluated the registration results. The average pixel distances after local registration for person, fish, and hand were 0.14, 0.24, and 0.08, respectively. This result was better than the one reported in [1], and indicates that shapes were well aligned by our method. Additionally, for most cases, the maximum pixel distance was around 3 pixels showing that registration quality was consistent along contours.

Our method was able to register shapes undergoing large deformation (e.g., bending arm in the person's sequence). The method also appears to be quite robust to partial occlusion. Figure 3 shows two examples of aligning occluded shapes. Due to severe occlusion, the shapes' distance transforms (Figure 3(b) and Figure 3(c)) were so distorted that the method in [4] would fail without a proper proximity function (Figure 3(e)). Using only distance values at the shape's boundary, our method was less sensitive to this distortion (Figure 3(d)).

In the case of the cell-morphing sequence, we manually initialized nodes along the cell's contour with roughly equal intervals, and the radii of the nodes were chosen such that each node had approximately two neighbors. Figure 4(d) shows

² Named dude in the original dataset.

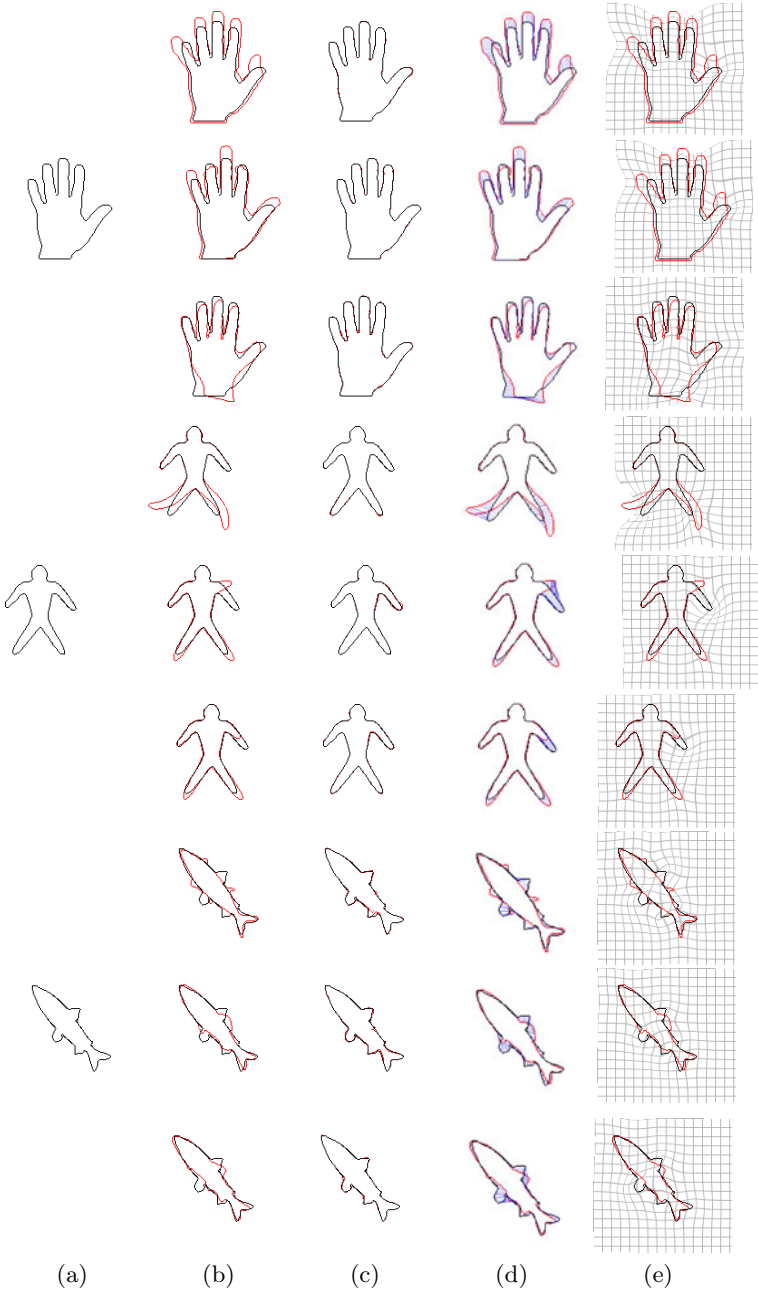


Fig. 2. Brown university shape dataset. (a) Target images. (b) Overlaid target (in black) and source images (in red) before registration. (c) After registration. (d) Correspondence between target and source images. (e) Deformation fields as distorted grids.

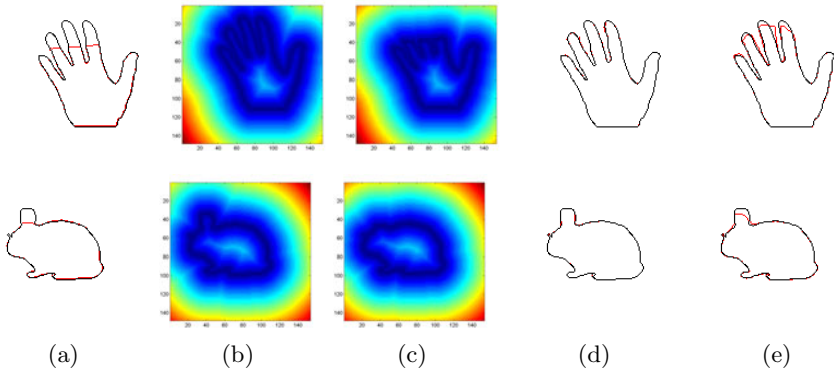


Fig. 3. Registration under partial occlusion. (a) Overlaid target and source images. (b) and (c): Distance transforms of target and source images. (d) Our method handles occlusions well as the registration error is only defined along shape contours. (e) The method in [4] would fail to align shape contours without a suitable proximity function.

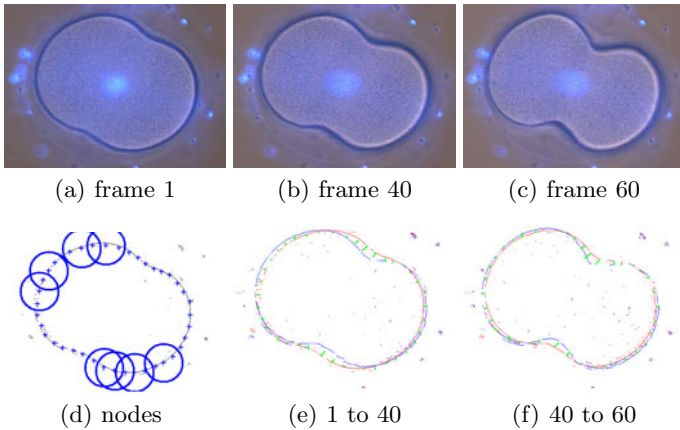


Fig. 4. Cell-morphing sequence. (a-c) Frames of a cell morphing sequence. (d) Sample nodes and corresponding influence regions. Nodes are placed along the contour. (e) and (f) Deformation vectors (green arrows).

the distribution of nodes. Here, the node’s positions are indicated as blue crosses, and their radii by circles. This way, the computation cost was significantly reduced. In Figure 4, we show three frames of the cell sequence, and our registration results. The cell’s deformation consisted of its contour bending inwards in the middle. The living cell’s surface exhibited random Brownian motion, with many spurious points, but our method was still able to register their boundaries.

Despite promising results, our method still encounters problems in registering shapes that have large curvatures, and undergo high-degree deformation, causing local minima in the registration error. We believe that this problem

can be addressed by adopting global-optimization algorithms such as simulated annealing [12], or by including statistical priors [6].

5 Conclusions

A meshless nonrigid shape-registration algorithm was presented. The registration functional is a variational extension of the classic chamfer-matching energy. As in chamfer matching, distance transforms provide registration-error gradients, facilitating efficient registration. Also, we modeled shape deformation using a meshless parametric representation. This model does not rely on a regular control-point grid, and can be adapted to arbitrary shapes. Thus, registration can be focused around the shape contours, greatly improving computational efficiency. We tested the proposed method by registering a number of synthetic shapes, and a deforming cell sequence. Future work includes a 3-D extension of the method, the handling of topological changes, and extensive comparison with state-of-the-art shape registration methods.

References

1. Chen, H., Bhanu, B.: Global-to-local non-rigid shape registration. In: ICPR, Washington, DC, USA, pp. 57–60. IEEE Computer Society, Los Alamitos (2006)
2. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 10, 849–865 (1988)
3. Paragios, N., Rousson, M., Ramesh, V.: Non-rigid registration using distance functions. *Comp. Vision and Image Underst.* 89, 142–165 (2003)
4. Huang, X., Paragios, N., Metaxas, D.: Shape registration in implicit spaces using information theory and free form deformations. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 28, 1303 (2006)
5. Kroon, D.J., Slump, C.H.: MRI modality transformation in demon registration. In: ISBI, Piscataway, NJ, USA, pp. 963–966. IEEE, Los Alamitos (2009)
6. Rousson, M., Paragios, N.: Prior knowledge, level set representations & visual grouping. *Int. J. Comput. Vision* 76, 231–243 (2008)
7. Makram-Ebeid, S., Somphone, O.: Non-rigid image registration using a hierarchical partition of unity finite element method. In: ICCV, vol. 510, p. 7 (2009)
8. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: CVPR, vol. 1 (2003)
9. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. *ACM Trans. Graph.* 22, 463–470 (2003)
10. Liu, G.R.: *Mesh free methods: moving beyond the finite element method*, 2nd edn. CRC, Boca Raton (2009)
11. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and simple calculus on tensors in the Log-Euclidean framework. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 115–122. Springer, Heidelberg (2005)
12. Bonnans, J., Lemaréchal, C.: *Numerical optimization: theoretical and practical aspects*. Springer, New York (2006)
13. Sharvit, D., Chan, J., Tek, H., Kimia, B.B.: Symmetry-based indexing of image databases. *J. of Vis. Comm. and Image Repres.* 9, 366–380 (1998)

A New Simple Method to Stitch Images with Lens Distortion

Myung-Ho Ju and Hang-Bong Kang

Dept. of Computer Eng. And Dept. of Media Eng.
Catholic University of Korea

#43-1 Yokkok 2-dong Wonmi-Gu, Bucheon, Gyonggi-Do Korea
hange15@catholic.ac.kr, bkang@catholic.ac.kr

Abstract. Lens distortion is one of the main problems that makes it difficult to correctly stitch images. Since the lens distortion cannot be linearly represented, it is hard to define the correspondences between images linearly or directly when the images are stitched. In this paper, we propose an efficient image stitching method for images with various lens distortions. We estimate accurate lens distortion using the ratio of lengths between matching lines in each matched image. The homographies between each matched images are estimated based on the estimated lens distortion. Since our technique works in the RANSAC phase, the additional time to estimate the distortion parameters is very short. Our experimental results show that our proposed method can efficiently and automatically stitch images with arbitrary lens distortion better than other current methods.

1 Introduction

Recently, the general process to stitch partially overlapped images is proposed with good results [1-4]. However, when the ideal homographies for the image stitching are used, the misalignment or blurred areas occur due to the lens distortions. Real images taken with various cameras may have lens distortions which are non-negligible. To correctly stitch images, therefore, it is necessary to estimate the distortion parameters accurately.

Estimating the lens distortion parameters may be difficult since the lens distortion has a nonlinear property. Fig. 1 shows some examples in which the image is distorted

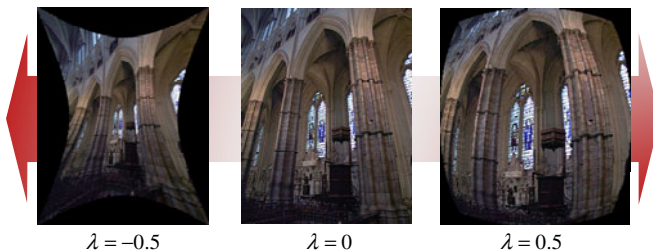


Fig. 1. The lens distortion images based on the Division Model. λ is the distortion parameter.

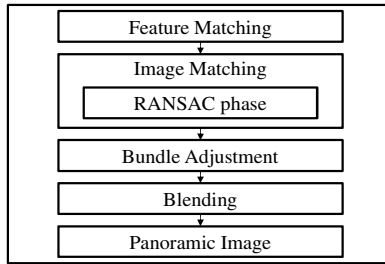


Fig. 2. The general process to stitch images with partially overlapped images

according to the ratio of the lens distortion based on the Division Model [5]. If the distortion parameters are the same for all images we have to stitch, we may use the projection technique such as cylindrical projection that may compensate lens distortions using the focal length. However, images usually have different distortion parameters, and these distortions make some misalignments in the stitched image.

To handle this problem, traditional works compensated the lens distortions at the bundle adjustment phase. However, it did not take into account for nonlinear effects presented at the RANSAC stage. To deal with these effects, the most recent technology solve the lens distortions problem at the RANSAC stage. Fitzgibbon [6] proposed an easy method to formulate and solve geometric computer vision problems using radial distortion. He estimated distortion parameters with homography by solving the eigen problem. However, his method has a problem in the accuracy of distortion parameters because of the ambiguity to which eigenvalue is applied. Josephsons and Byröd [5] solved the polynomial equations including distortion parameters with the Gröbner basis method. Byröd et al. [4] applied their method to image stitching. However, although they could estimate quite accurate distortion parameters, they needed a large computational time since Gröbner basis method has high complexity. To deal with accuracy as well as computational time, we propose a simple technique to stitch images with various lens distortions.

From [1, 2], the general procedure to stitch images is proposed as shown in Fig. 2. Our approach follows this procedure except for the RANSAC phase in the Image Matching stage. We first estimated distortion parameters in the matched images using the ratio of lengths between matching lines. Assuming that the homography between matched images has affine transformation [9], the ratio of lengths on parallel lines is preserved. Each homography is estimated based on the estimated distortion parameters. Finally, the bundle adjustment phase helps to correct the estimated parameters since the estimated distortion parameters are very close to the real ones.

2 Lens Distortion Model

The most popular models to describe the lens distortion are the Polynomial Model [8] and Division Model (DM) [5]. The Polynomial Model is shown as

$$x_u - c = (x_d - c)L(r_d, \lambda), \tag{1}$$

where c is the centre-of-distortion and x_d and x_u are the distorted and undistorted positions of the image, respectively. And $L(r_d, \lambda)$ is defined as

$$L(r_d, \lambda) = 1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \dots + \lambda_p r_d^{2p}, \tag{2}$$

where $r_d = \|x_d\|$ and λ are the distortion parameters.

The Polynomial Model works best for lenses with small distortions. For wide-angle lens such as a fish-eye lens that has large distortion, it requires too many terms to be practical. On the other hand, DM is able to express high distortion at much lower order. This is the main reason to use DM in this paper. DM is defined as

$$x_u - c = \frac{x_d - c}{L(r_d, \lambda)}, \tag{3}$$

where $L(r_d, \lambda)$ is the same as Eq. 2. For the easier calculations, we used $L(r_d, \lambda) = 1 + \lambda_1 r_d^2$. Fig. 1 shows some examples of the distorted images when the distortion parameter moves from -0.5 to +0.5.

3 Estimation of the Distortion Parameters

We assumed that the homography between matched images has affine transformation in which the ratio of the parallel lines' lengths is preserved. However, in the distorted images, this property is not preserved due to the nonlinearity of lens distortion. Our approach is simply to estimate the distortion parameters in the RANSAC stage and find undistorted images in which this property is satisfied.

Assuming that all input images have different lens distortions and that these distortions are represented by DM, the projection between the feature point of the i th input image, $u_i = (x^i, y^i, 1 + \lambda_i r_i^2)$ and the matched feature point of the j th input image, u_j is defined as

$$\begin{bmatrix} x^i \\ y^i \\ 1 + \lambda_i r_i^2 \end{bmatrix} = \mathbf{H}_{ij} \begin{bmatrix} x^j \\ y^j \\ 1 + \lambda_j r_j^2 \end{bmatrix}, \tag{4}$$

where \mathbf{H}_{ij} is the homography from j th image to i th image. λ_i is the distortion parameter of the i th image and r_i is the normalized distance between the feature point u_i and its principal point. Since digital cameras have square pixels and the principal point close to the center of the image, we impose these assumptions to our camera model. The normalized distance is computed from the division of each distance by the maximum distance and resulted in the range 0 ~ 1.0.

With DM, the arbitrary point A in the image is distorted to another point A' on the line drawn from the image center to itself by lens distortion as shown in Fig. 3. In these distortions, there exist three properties. First, the points that have equal distance from the image center (such as A and C in Fig. 3) will still have equal distance from the center after the distortion. Secondly, when two arbitrary points have different distance from the image center (such as A and B in Fig. 3), the point that is farther away than the other point will be also located farther than the other after transformation- (the distance of $\overline{B'B}$ is larger than the distance of $\overline{A'A}$). Finally, the angle between two points is always preserved regardless of the transformation (the angle θ between the line \overline{AO} and \overline{BO} is always the same after any distortion transform) and therefore it is irrelevant with the distortion.

Based on those properties, we will discuss line properties in the distorted images. As shown in Fig. 4(a), if two points P and Q have the same distance from the image center, the distance $d(\overline{PQ})$ between the arbitrary two points P and Q is distorted to $d(\overline{P_dQ_d})$ linearly. The distance $d(\overline{P_dQ_d})$ is computed as

$$d(\overline{P_dQ_d}) = \frac{1}{1 + \lambda r_p^2} d(\overline{PQ}) = \frac{1}{1 + \lambda r_Q^2} d(\overline{PQ}) \tag{5}$$

where r_p and r_Q are the normalized distance from the image center to the point P and Q , respectively.

As shown in Fig. 4 (b), if two points P and Q have different distance from the center, the distance $d(\overline{PQ})$ between the arbitrary two points P and Q is distorted to $d(\overline{P_1Q_2})$. The distance $d(\overline{P_1Q_2})$ is computed as follows. Let P_1 and Q_2 be the distorted points from the point P and Q in the direction of r_p and r_Q , respectively. We considered the distorted points P_2 and Q_1 to have the same normalized distance as

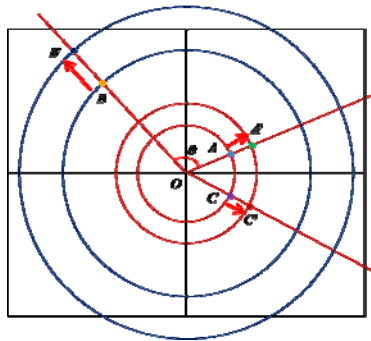


Fig. 3. The properties of the transform according to the lens distortion based on Division Model: when the distortion parameter is larger than 0, each points A , B and C are transformed to A' , B' and C' respectively

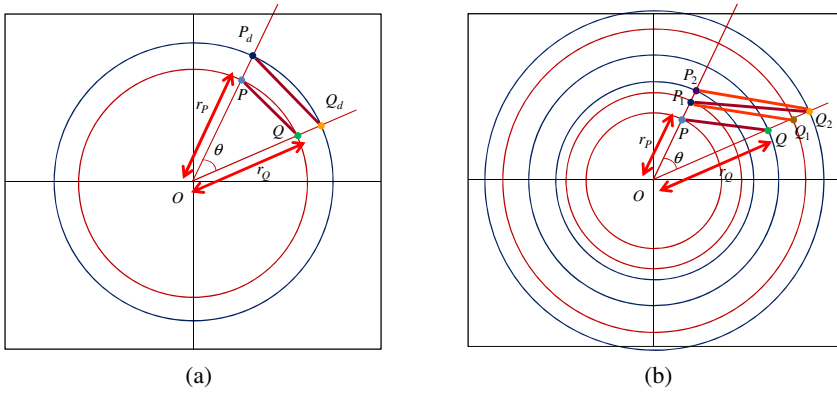


Fig. 4. The transformation of the distance between two distorted points: (a) two points have the same distance from the center, (b) two points have different distance from the center

Q_2 and P_1 , respectively. The distance $d(\overline{P_1Q_1})$ and $d(\overline{P_2Q_2})$ can be simply computed from Eq. 5, while the distance $d(\overline{P_1Q_2})$ is approximated as

$$d(\overline{P_1Q_2}) = \frac{d(\overline{P_1Q_1}) + d(\overline{P_2Q_2})}{2} \tag{6}$$

This simple approximation doesn't always provide an accurate line distance. Fig. 5 shows the line distances computed by Eq. 6 according to the change of the angle $\angle POQ$ and the distance between two points. From Fig. 5, the error of our approximated distance is small when the distortion parameter and the distance between points are not too large. The range of distortion parameters is generally from -0.5 to 0.5. In this range, our approach can compute distance of the line almost accurately.

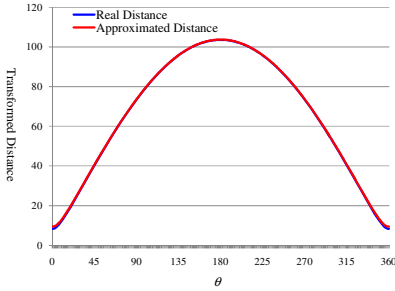
Let us consider a pair of parallel lines, $\overline{A_iB_i}$ and $\overline{C_iD_i}$ on the i th input image, as shown in Fig. 6. The projected lines on the j th input image are $\overline{A_jB_j}$ and $\overline{C_jD_j}$, respectively. Since two lines are parallel, the ratio of the line $\overline{A_iB_i}$ and $\overline{C_iD_i}$ is preserved as

$$d(\overline{A_iB_i}) : d(\overline{C_iD_i}) = d(\overline{A_jB_j}) : d(\overline{C_jD_j}) \tag{7}$$

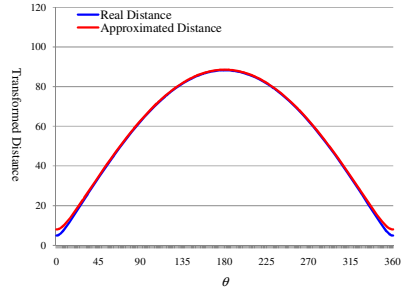
From Eq. 6, Eq. 7 can be expressed as

$$\begin{aligned} & \left(\frac{1}{1 + \lambda_i r_{A_i}^2} d(\overline{A_iB_i}) + \frac{1}{1 + \lambda_i r_{B_i}^2} d(\overline{A_iB_i}) \right) \left(\frac{1}{1 + \lambda_j r_{C_j}^2} d(\overline{C_jD_j}) + \frac{1}{1 + \lambda_j r_{D_j}^2} d(\overline{C_jD_j}) \right) \\ &= \left(\frac{1}{1 + \lambda_j r_{A_j}^2} d(\overline{A_jB_j}) + \frac{1}{1 + \lambda_j r_{B_j}^2} d(\overline{A_jB_j}) \right) \left(\frac{1}{1 + \lambda_i r_{C_i}^2} d(\overline{C_iD_i}) + \frac{1}{1 + \lambda_i r_{D_i}^2} d(\overline{C_iD_i}) \right) \end{aligned} \tag{8}$$

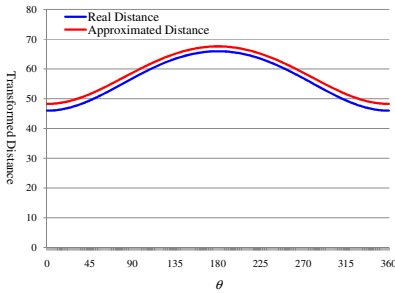
where λ_i is the distortion parameter of the i th image and r_{A_i} is the normalized distance from the image center to the point A_i . Eq. 8 is arranged by the bi-quadratic



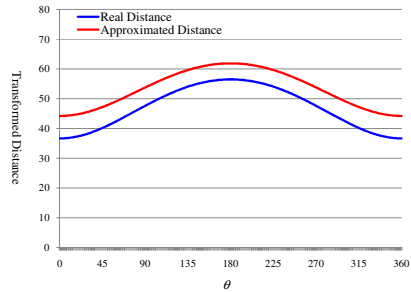
(a) $\lambda=0.1$ and the distance between each points is small (normal distance = 0.1)



(b) $\lambda=0.8$ and the distance between each points is small (normal distance = 0.1)



(c) $\lambda=0.1$ and the distance between each points is large (normal distance = 0.5)



(d) $\lambda=0.8$ and the distance between each points is large (normal distance = 0.5)

Fig. 5. The error graph for the approximated distance according to the distortion parameter and the distance between points

equation. Since the distortion parameter is quite small (in this paper, we assume that it is in $-0.5 \sim 0.5$), we can ignore the terms which have the degree of λ that is larger than quadratic. For a simple equation, we can remove the quadratic terms of distortion parameters since the quadratic terms have many multiplied terms of the normalized distances which are always smaller than 1.0. Finally, the Eq. 8 is approximated as

$$a\lambda_i + b\lambda_j + c\lambda_i\lambda_j = d,$$

where

$$\begin{aligned} a &= 2d(\overline{A_i B_i})d(\overline{C_j D_j})(R(A_i B_i) + 2R(C_i D_i)) - 2d(\overline{A_j B_j})d(\overline{C_i D_i})(2R(A_i B_i) + R(C_i D_i)) \\ b &= 2d(\overline{A_i B_i})d(\overline{C_j D_j})(2R(A_j B_j) + R(C_j D_j)) - 2d(\overline{A_j B_j})d(\overline{C_i D_i})(R(A_j B_j) + 2R(C_j D_j)) \\ c &= \left[\begin{aligned} &d(\overline{A_i B_i})d(\overline{C_j D_j}) \left(2R(A_i B_i)R(A_j B_j) + 4R(C_i D_i)R(A_j B_j) + 2R(C_i D_i)R(C_j D_j) \right) \\ &+ r_{A_i}^2 r_{C_j}^2 + r_{A_i}^2 r_{D_j}^2 + r_{B_i}^2 r_{C_j}^2 + r_{B_i}^2 r_{D_j}^2 \end{aligned} \right] \\ &\left[\begin{aligned} &-d(\overline{A_j B_j})d(\overline{C_i D_i}) \left(2R(A_i B_i)R(A_j B_j) + 4R(A_i B_i)R(C_j D_j) + 2R(C_i D_i)R(C_j D_j) \right) \\ &+ r_{A_j}^2 r_{C_i}^2 + r_{A_j}^2 r_{D_i}^2 + r_{B_j}^2 r_{C_i}^2 + r_{B_j}^2 r_{D_i}^2 \end{aligned} \right] \\ d &= 4d(\overline{A_j B_j})d(\overline{C_j D_j}) - 4d(\overline{A_i B_i})d(\overline{C_j D_j}) \end{aligned} \tag{9}$$

where $R(A_i B_i) = r_{A_i}^2 + r_{B_i}^2$.

Even though a , b , c and d in Eq. 9 seem to be complicated, it has many duplicated parts. So, we can easily compute them by simple operations of additions and multiplication in practice. Eq. 9 is converted to $AX = B$, where $A = [a \ b \ c]$, $B = [c]$ and $X = [\lambda_i \ \lambda_j \ \lambda_i \lambda_j]^T$. When there are at least three matching lines between the matched images, it can be solved by least square problem and we can estimate the approximated distortion parameter λ_i and λ_j of the i th and j th input image easily.

To compensate our approximation, we define the following three additional weights. First, we define the belief weight w_{belief} , since the approximated distortion parameter is positively or negatively larger than the real one due to the dropped terms which are larger than quadratic. To rescale the parameter, we experimentally choose 0.8 for the belief weight. Secondly, we define the parallel weight $w_{parallel}$. In the RANSAC phase, we detect three most parallel lines which have the smallest included angle between lines. Sometimes they may not be parallel with each other. To solve this problem, we define the parallel weight using the cosine function of the included angle between each line as

$$w_{parallel} = \frac{1}{\Lambda} \sum_{\theta \in S} \begin{cases} \cos \theta & \text{if } \theta < 90^\circ \\ 1 - \cos \theta & \text{elsewise} \end{cases} \tag{10}$$

where S is the set of all angles between lines, and Λ is the normalized term.

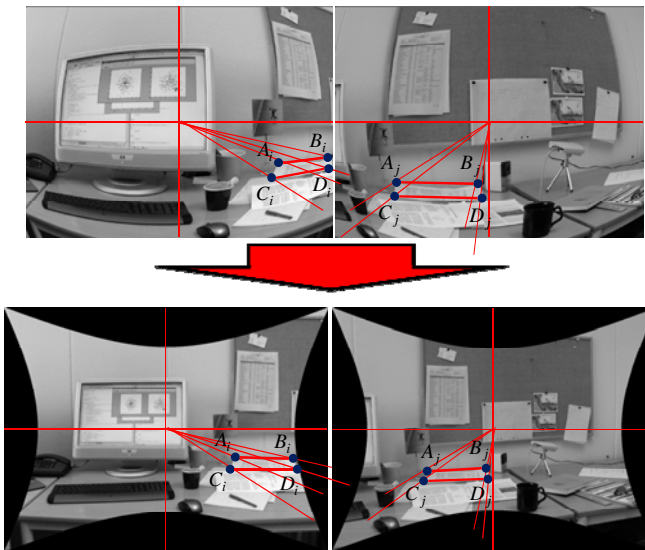


Fig. 6. An example of distortion correction using our method

Thirdly, we consider the length weight w_{length} since the resulted λ from longer lengths has large errors. The length weight is defined as

$$w_{length} = \prod_{AB \in L} \begin{cases} 1 & \text{if } d(\overline{AB}) > \nu \\ \exp\left(-\frac{(d(\overline{AB}) - \nu)^2}{\Lambda}\right) & \text{elsewise} \end{cases} \quad (11)$$

where L is a set of lines used for the calculation, ν is the threshold and Λ is the normalize factor. We choose 10 for ν and 4.0 for Λ in our experiments.

Finally, we estimate the distortion parameter as

$$\lambda = w_{belief} w_{parallel} w_{length} \lambda_{est} \quad (12)$$

where λ_{est} is the estimated distortion parameter from Eq. 9.

4 Experimental Results

Our approach is implemented as shown in Fig. 2 except for the RANSAC phase. In the RANSAC phase, we randomly choose eight points extracted from SIFT [7]. We detect the three most parallel lines from points. Using these lines, we calculated the distortion parameters for image pair using the Eq. 12. We estimate the homography using four of eight points based on the estimated distortion parameters.

To evaluate the performance of our proposed method, we tested our method with various input images which are taken from several different indoor and outdoor areas such as a school gate, restaurant, hallway, etc. The input images are seriously distorted by arbitrarily adjusting each distortion parameters based on DM as shown in Fig. 7 and Fig. 9(a). To compare our proposed method with one of the state-of-art method for lens distortion, we tested the images from [4] as shown in Fig. 8.



Fig. 7. Examples of distorted input images based on Division Model

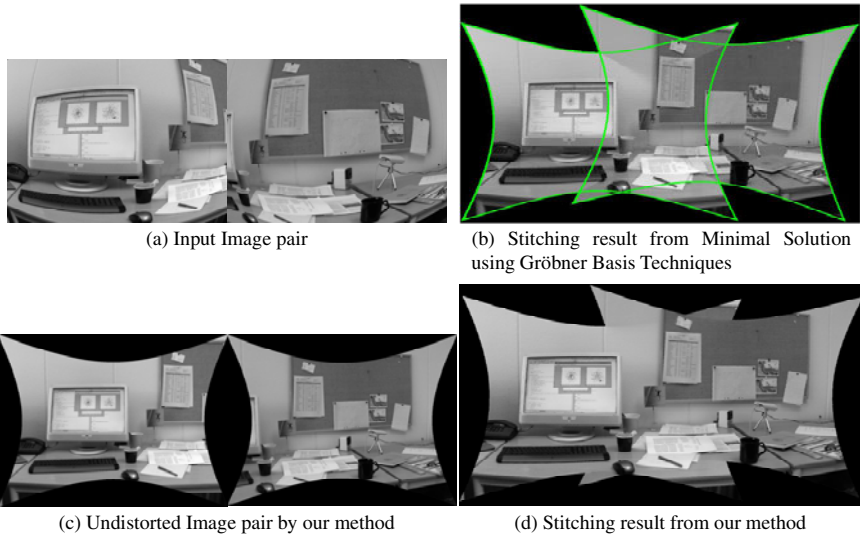


Fig. 8. The comparison results with [6]. Our result is also compensated the heavy distortions well similar to [6].

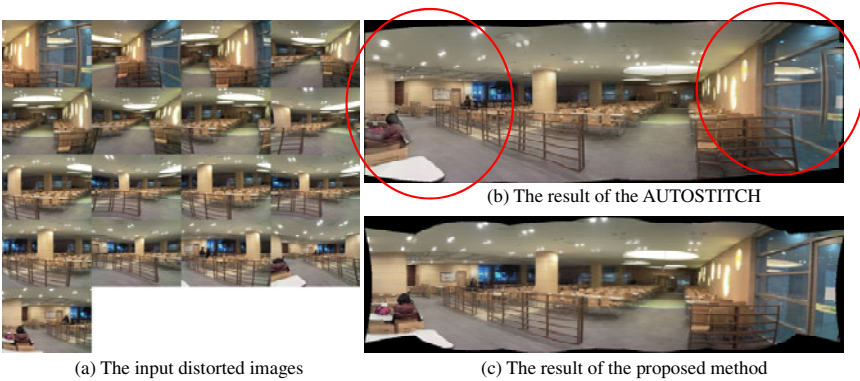


Fig. 9. The comparison results with AUTOSTITCH Demo. The input images are captured in a restaurant.

In comparison with the results from Minimal Solution using Gröbner Basis Techniques [4], we got similar ones as shown in Fig. 8. Our method needs to solve the least square problem of 3×3 matrix while [4] needs to solve the eigen problem of 90×132 coefficient matrix. The computational time of the Gröbner basis method is 13 milliseconds/instance on a standard 2Ghz machine when the method is implemented in MATLAB. On the other hand, we implemented our method using Visual C++ on a 2.33GHz machine and ended with the running time is about 8.5 milliseconds/instance. Although the difference between two methods is about 4.5 milliseconds/instance, it is increased to 900 milliseconds (about 1 sec) because the RANSAC usually performs 200 iterations. So, our approach is faster than other existing methods.

Fig. 9 shows one of the comparison results with AUTOSTITCH Demo program [10] when the arbitrarily distorted images are stitched. While the result of the AUTOSTITCH Demo program shows much error (shown as red circle) as shown in Fig. 9 (b), our proposed approach shows a visually pleasant stitched result like Fig. 9 (c).

5 Conclusion

In this paper, we proposed a new simple method to stitch images with arbitrary lens distortions. We estimated the accurate lens distortion using the ratio of lengths between corresponding lines in each matched image. The homographies between each matched image are estimated based on the estimated lens distortion. Since our technique works in the RANSAC phase, the additional time to estimate the distortion parameters is very short. We obtained better results than the previously suggested methods. However, when the lines from features are almost orthogonal (this is the worst case scenario and the parallel weight in Eq. 12 has almost zero value), our method cannot estimate the correct distortion parameters because the ratio of the lines is not equal. In the future research, we will work on this problem.

Acknowledgements

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract .UD1000011D.

References

1. Brown, M., Lowe, D.G.: Recognising Panoramas. In: Proceeding of the International Conference on Computer Vision, vol. 2, pp. 1218–1225 (2003)
2. Brown, M., Lowe, D.G.: Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision* 74, 59–73 (2007)
3. Szeliski, R.: Image Alignment and Stitching: A Tutorial. *Foundations and Trends in Computer Graphics and Computer Vision* 2(1), 1–104 (2006)
4. Byröd, M., Brown, M., Aström, K.: Minimal Solutions for Panoramic Stitching with Radial Distortion. In: Proceeding of The 20th British Machine Vision Conference (2009)
5. Josephson, K., Byröd, M.: Pose Estimation with Radial Distortion and Unknown Focal Length. In: Proceeding of Computer Vision and Pattern Recognition Conference (2009)
6. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: Proceedings of Computer Vision and Pattern Recognition Conference, pp. 125–132 (2001)
7. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceeding of the International Conference on Computer Vision, pp. 1150–1157 (1999)
8. Li, H., Hartley, R.: A Non-iterative Method for Correcting Lens Distortion from Nine Point Correspondences. In: Proceeding of the International Conference on Computer Vision Workshop (2005)
9. Hartley, R., Zisserman, A.: *Multiple View Geometry in computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN:0521540518
10. AUTOSTITCH Website,
<http://people.cs.ubc.ca/~mbrown/autostitch/autostitch.html>

Robust Mosaicking of Stereo Digital Elevation Models from the Ames Stereo Pipeline

Taemin Kim¹, Zachary Moratto¹, and Ara V. Nefian²

¹ NASA Ames Research Center, Moffett Field, CA, 94035

² Carnegie Mellon University

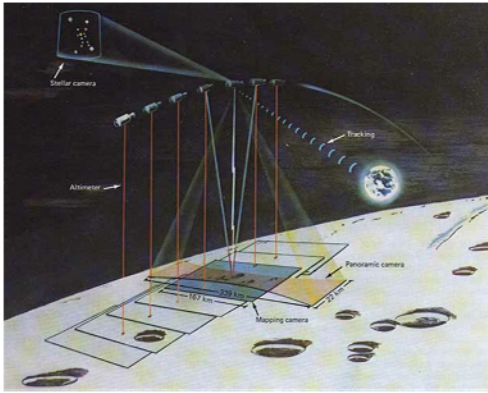
Abstract. Robust estimation method is proposed to combine multiple observations and create consistent, accurate, dense Digital Elevation Models (DEMs) from lunar orbital imagery. The NASA Ames Intelligent Robotics Group (IRG) aims to produce higher-quality terrain reconstructions of the Moon from Apollo Metric Camera (AMC) data than is currently possible. In particular, IRG makes use of a stereo vision process, the Ames Stereo Pipeline (ASP), to automatically generate DEMs from consecutive AMC image pairs. However, the DEMs currently produced by the ASP often contain errors and inconsistencies due to image noise, shadows, etc. The proposed method addresses this problem by making use of multiple observations and by considering their goodness of fit to improve both the accuracy and robustness of the estimate. The stepwise regression method is applied to estimate the relaxed weight of each observation.

1 Introduction

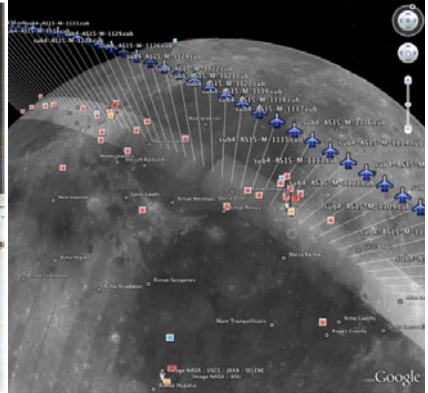
Since 2007, the NASA Lunar Mapping and Modeling Project (LMMP) has been actively developing maps and tools to improve lunar exploration and mission planning [1]. One of the requirements for LMMP is to construct geo-registered DEMs from historic imagery. To meet this need, IRG has developed the Ames Stereo Pipeline (ASP), a collection of cartographic and stereogrammetric tools for automatically producing DEMs from images acquired with the Apollo Metric Camera (AMC) during Apollo 15-17 (Figure 1).

There are many applications of lunar maps, including outreach and education, mission planning, and lunar science. The maps and imagery that IRG released in 2009 for “Moon in Google Earth” have engendered great public interest in lunar exploration. The cartographic products that IRG is currently producing (DEMs, ortho-projected imagery, etc.) will be used to plan future missions, to assess landing sites, and to model geophysical processes. Consequently, by improving these maps, we directly benefit a wide community.

The NASA Ames Intelligent Robotics Group (IRG) currently uses a stereo vision process to automatically generate DEMs from image pairs [2]. However, two DEMs generated from different image pairs have different values for the same point due to noise, shadows, etc. in the images (Figure 2a). Consequently, IRG’s current



(a) The Mapping Cameras System

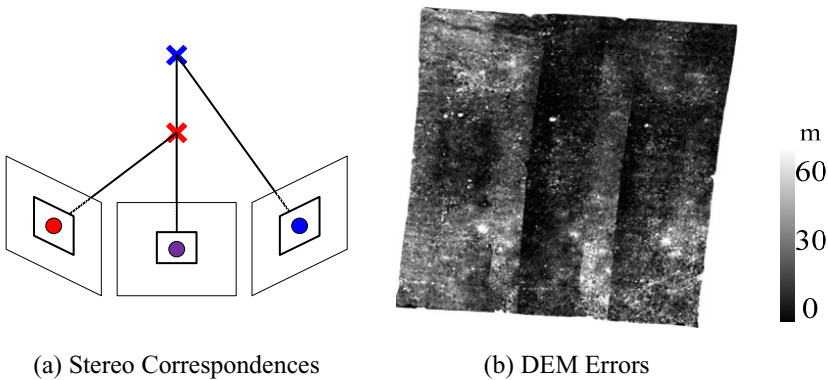


(b) Apollo 15 Orbit 33

Fig. 1. AMC Data System. (a) The AMC captures a series of pictures of the Moon's surface. (b) Satellite station positions for Apollo Orbit 33 visualized in Google Moon.

topographical reconstruction of the Moon contains fairly substantial random errors (Figure 2b). It is important to construct consistent DEMs to estimate the photometric properties [3-5].

This paper will address this problem by finding robust elevation from multiple DEMs that minimize the weighted squared error of all associated DEM patches. The proposed method determines the unique 3D position by weighted averaging of all elevation values from stereo DEMs. The accuracy and robustness of DEMs produced by IRG will thus be improved by making use of multiple observations and by considering their goodness of fit. The reconstructed DEMs from lunar orbital imagery are presented. This paper tackles one of the challenging problems of planetary mapping from orbital imagery.



(a) Stereo Correspondences

(b) DEM Errors

Fig. 2. Stereo and Multiple View Correspondences. (a) Two stereo correspondences have different elevation values (two crosses). (b) DEMs created by stereo can have substantially large errors.

2 Ames Stereo Pipeline

The Ames Stereo Pipeline (ASP) is the stereogrammetric platform that was designed to process stereo imagery captured by NASA spacecraft and produce cartographic products since the majority of the AMC images have stereo companions [6]. The entire stereo correlation process, from an image pair to DEM, can be viewed as a multistage pipeline (Figure 3). At the first step, preprocessing includes the registration to align image pairs and filtering to enhance the images for better matching. Triangulation is used at the last step to generate a DEM from the correspondences.

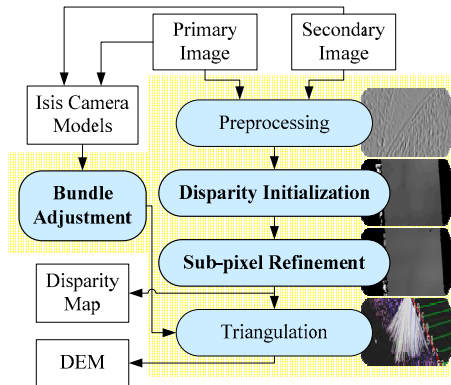


Fig. 3. Dataflow of the Ames Stereo Pipeline. Preprocessing includes the registration and filtering of the image pair. A stereo correlator (disparity initialization and sub-pixel refinement) constructs the disparity map based on normalized cross correlation. DEMs are generated by a triangulation method in which corrected camera poses are used by bundle adjustment.

2.1 Disparity Initialization

Stereo correlation, which is the process at the heart of ASP, computes pixel correspondences of the image pair (Figure 4). The map of these correspondences is called a *disparity map*. The best match is determined by applying a cost function that compares the two windows in the image pair. The normalized cross correlation is robust to slight lighting and contrast variation in between a pair of images [7]. For large images, this is computationally very expensive, so the correlation process is split into two stages. (1) The disparity map initialization step computes coarse correspondences using a multi-scale search that is highly optimized for speed (Figure 4c). (2) Correlation itself is carried out by sliding a small, square template window from the left image over the specified search region of the right image (Figure 4d-f).

Several optimizations are employed to accelerate disparity map initialization [8]: (1) a box-filter-like accumulator that reduces duplicate operations during correlation; (2) a coarse-to-fine multi-scale approach where disparities are estimated using low resolution images, and then successively refined at higher resolutions; and (3) partitioning of

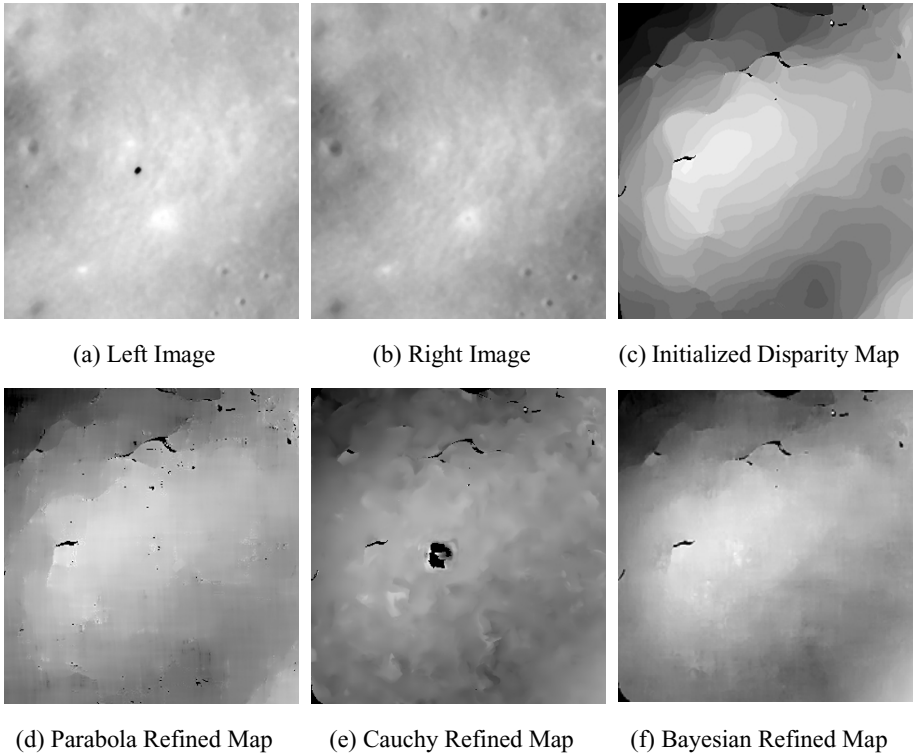


Fig. 4. Stereo Correlation. (a-b) An image pair. (c-f) Horizontal disparity maps. (c) The fast discrete correlator constructs the coarse disparity map. (d-f) Refined disparity maps from the initialized map. (f) Bayesian sub-pixel correlator generates a smoother map than the others.

the disparity search space into rectangular sub-regions with similar values of disparity determined in the previous lower resolution level of the pyramid.

2.2 Sub-pixel Refinement

Refining the initialized disparity map to sub-pixel accuracy is crucial and necessary for processing real-world data sets [9]. The Bayesian expectation maximization (EM) weighted affine adaptive window correlator was developed to produce high quality stereo matches that exhibit a high degree of immunity to image noise (Figure 4f). The Bayesian EM sub-pixel correlator also features a deformable template window that can be rotated, scaled, and translated as it zeros in on the correct match. This affine-adaptive window is essential for computing accurate matches on crater or canyon walls, and on other areas with significant perspective distortion due to foreshortening. A Bayesian model that treats the parameters as random variables was developed in an EM framework. This statistical model includes a Gaussian mixture component to model image noise that is the basis for the robustness of the algorithm. The resulting DEM is obtained by the triangulation method (Figure 5).

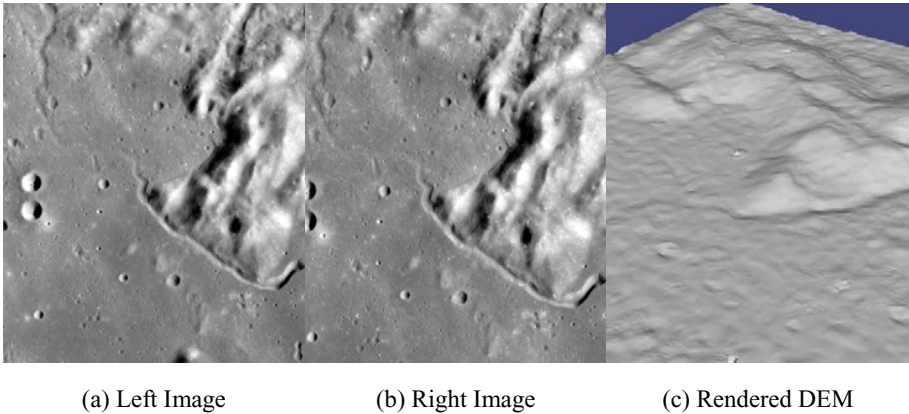


Fig. 5. Generation of DEM. (a) and (b) Apollo Metric Camera image pair of Apollo 15 site. (c) A DEM of Hadley Rille is rendered from an image pair.

2.3 Bundle Adjustment

After stereo correlation is performed, Bundle Adjustment (BA) corrects the three-dimensional postures of cameras and the locations of the objects simultaneously to minimize the error between the estimated location of the objects and their actual location in the images. Camera position and orientation errors have a direct effect on the accuracy of DEMs produced by the ASP. If they are not corrected, these uncertainties will result in systematic errors in the overall position and slope of the DEMs (Figure 6a). BA ensures that observations in multiple different images of a single ground feature are self-consistent (Figure 6b).

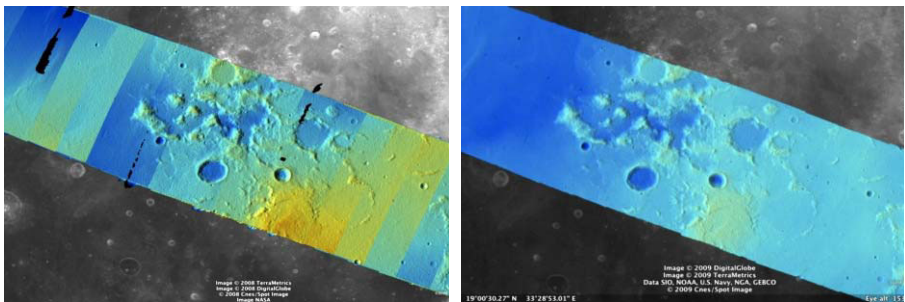


Fig. 6. Bundle Adjustment. Color-mapped, hill-shaded DEM mosaics from Apollo 15 Orbit 33 imagery illustrate the power of BA. (a) Prior to BA, large discontinuities exist between overlapping DEMs. (b) After BA, DEM alignment errors are minimized, and no longer visible.

In BA the position and orientation of each camera station are determined jointly with the 3D position of a set of image tie-points points chosen in the overlapping regions between images. This optimization is carried out along with thousands (or more) of similar constraints involving many different features observed in other images.

Tie-points are automatically extracted using the SURF robust feature extraction algorithm [10]. Outliers are rejected using the random sample consensus (RANSAC) method [11]. The BA in ASP determines the best camera parameters that minimize the reprojection error [12]. The optimization of the cost function uses the Levenberg-Marquardt algorithm [13].

3 Robust Estimation

Suppose a set of observations $P = \{x_k\}_{k=1}^n$ from a normal distribution with some of them are contaminated by outliers. Their Boolean membership to inliers is represented by an indicator vector $\mathbf{w} = [w_k] \in \{0,1\}^n$:

$$w_k = \begin{cases} 0 & \text{if } x_k \text{ is outlier} \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Let w be the total number of inliers:

$$w = \sum_{i=1}^n w_k. \tag{2}$$

The membership measure supporting the normality is defined to minimize its square error and to maximize the number of inliers. The mean of the inliers is written by

$$\bar{x} = \frac{1}{w} \sum_{i=1}^n w_k x_k. \tag{3}$$

The squared error is written by

$$\begin{aligned} SSE &= \sum_{k=1}^n w_k (x_k - \bar{x})^2 \\ &= \frac{\sum_{k=1}^n w_k \sum_{k=1}^n w_k x_k^2 - \left(\sum_{k=1}^n w_k x_k\right)^2}{\sum_{k=1}^n w_k}, \end{aligned} \tag{4}$$

An objective function which minimizes s^2 and maximizes w simultaneously is hard to define. Even if it is defined, computing the optimum \mathbf{w} is NP-hard.

The Boolean membership vector is relaxed to be a real value in $[0,1]$ to convert the combinatorial optimization problem into a tractable nonlinear optimization problem [14]. The membership matrix reflects the likelihood of the inliers. Let us redefine $\mathbf{w} = [w_i] \in [0,1]^n$, where w_i is the membership of point x_i to inliers. Then the equations from (3) to (7) are still valid. From the expected value of SSE ,

$$E(SSE) = \left(w - \frac{1}{w} \sum_{k=1}^n w_k^2 \right) \sigma^2, \tag{5}$$

The mean squared error is defined by

$$MSE = \frac{wSSE}{w^2 - \sum_{k=1}^n w_k^2} = \frac{\sum_{k=1}^n w_k \sum_{k=1}^n w_k x_k^2 - \left(\sum_{k=1}^n w_k x_k \right)^2}{\left(\sum_{k=1}^n w_k \right)^2 - \sum_{k=1}^n w_k^2}, \quad (6)$$

The squared error of each point is defined by

$$\frac{\partial SSE}{\partial w_i} = (x_i - \bar{x})^2, \quad (7)$$

The statistic to test statistical significance for x_i follows the F distribution:

$$s_i = \frac{(x_i - \bar{x})^2}{v_i} \bigg/ \frac{\sum_{k=1}^n w_k x_k^2 - \bar{x}^2}{v} \sim F_{v_i, v}, \quad (8)$$

where v_i and v are the degrees of freedom such that

$$v_i = 1 - \frac{w_i}{w} \quad \text{and} \quad v = w - \frac{1}{w} \sum_{k=1}^n w_k^2 \quad (9)$$

The updating principle is to increase the weight of a observation if its p -value of the statistic is smaller than the significance level or to decrease the weight. Similar with the gradient descent method, the update rule is implemented by

$$\Delta w_i = \eta \{1 - F_{v_i, v}^{-1}(s_i) - \alpha\}, \quad (10)$$

where α and η are significance level and learning rate.

4 Experimental Results

The NASA Exploration Systems Mission Directorate (ESMD) has been charged with producing cartographic products via LMMP for use by mission planners and scientists in NASA's Constellation program. As part of the LMMP, we have produced 70 preliminary DEMs and ortho-images derived from Apollo 15 Metric Camera (AMC) orbit 33 imagery using the ASP. The DEM Mosaic is implemented based on the NASA Vision Workbench (VW). The NASA VW is a general purpose image processing and computer vision library developed by the IRG at the NASA Ames Research Center.

The unified DEM mosaics are constructed with the stereo DEMs. The significance level α was set to be 0.3 and the typical learning rate η is 0.01. Figure 7a shows a DEM mosaic image constructed from its adjacent stereo DEMs which has large variances between DEMs as shown in Figure 2b. As you can see in the figure, the unified DEM mosaic provides the seamless elevation map. Figure 7b shows the estimated weight for the current DEM as we can expected that the central part of the DEM has large confidence.

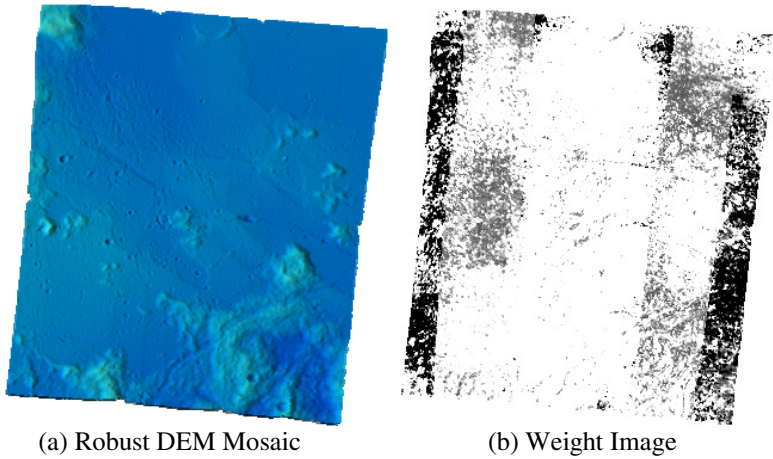


Fig. 7. DEM Mosaic and its weight

5 Conclusion

The robust estimation method was proposed to determine the unique elevation from multiple digital elevation models generated by the Ames Stereo Pipeline. Introducing the relaxed membership of each observation to be an inlier, the stepwise regression method is applied to estimate the relaxed weight of the observation. The proposed method addresses this problem by making use of multiple observations and by considering their goodness of fit to improve both the accuracy and robustness of the estimate. A gradient descent method was used to optimized the weight because it is hard to define the objective function in a closed form. The residual analysis will be desirable to provide a quantitative measure of the proposed method. The parametric representation of surface model will be valuable to enhance the recovery resolution and robustness of the algorithm.

Acknowledgement

This research was supported by an appointment to the NASA Postdoctoral Program at the Ames Research Center, administered by Oak Ridge Associated Universities through a contract with NASA.

References

1. Noble, S.K., et al.: The Lunar Mapping and Modeling Project. LPI Contributions 1515, 48 (2009)
2. Broxton, M., et al.: 3D Lunar Terrain Reconstruction from Apollo Images. In: *Advances in Visual Computing*, pp. 710–719 (2009)
3. Kim, T., Nefian, A.V., Broxton, M.J.: Photometric recovery of Apollo metric imagery with Lunar-Lambertian reflectance. *Electronics Letters* 46, 631

4. Kim, T., Nefian, A., Broxton, M.: Photometric Recovery of Ortho-Images Derived from Apollo 15 Metric Camera Imagery. In: *Advances in Visual Computing*, pp. 700–709 (2009)
5. Nefian, A.V., et al.: Towards Albedo Reconstruction from Apollo Metric Camera Imagery, p. 1555.
6. Broxton, M.J., et al.: The Ames Stereo Pipeline: NASA's Open Source Automated Stereogrammetry Software. NASA Ames Research Center (2009)
7. Menard, C.: Robust Stereo and Adaptive Matching in Correlation Scale-Space. Institute of Automation, Vienna Institute of Technology (1997)
8. Sun, C.: Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision* 47(1), 99–117 (2002)
9. Nefian, A., et al.: A Bayesian Formulation for Subpixel Refinement in Stereo Orbital Imagery. In: *International Conference on Image Processing*, Cairo, Egypt (2009)
10. Bay, H., et al.: Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110(3), 346–359 (2008)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. Assoc. Comp. Mach* 24(6), 381–395 (1981)
12. Triggs, B., et al.: Bundle adjustment - a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) *ICCV-WS 1999*. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
13. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge (2003)
14. Kim, T., Woo, J., Kweon, I.S.: Probabilistic matching of lines for their homography, pp. 3453–3456 (2009)

Tissue Fate Prediction in Acute Ischemic Stroke Using Cuboid Models

Fabien Scalzo^{1,2}, Qing Hao¹, Jeffrey R. Alger¹, Xiao Hu², and David S. Liebeskind¹

¹ Dept. of Neurology, University of California, Los Angeles (UCLA)

² Dept. of Neurosurgery, NSDL, University of California, Los Angeles (UCLA)

Abstract. Early and accurate prediction of tissue outcome is essential to the clinical decision-making process in acute ischemic stroke. We present a quantitative predictive model that combines tissue information available immediately after onset, measured using fluid attenuated inversion recovery (FLAIR), with multi-modal perfusion features (Tmax, MTT, and TTP) to infer the likely outcome of the tissue. A key component is the use of randomly extracted, overlapping, cuboids (i.e. rectangular volumes) whose size is automatically determined during learning. The prediction problem is formalized into a nonlinear spectral regression framework where the inputs are the local, multi-modal cuboids extracted from FLAIR and perfusion images at onset, and where the output is the local FLAIR intensity of the tissue 4 days after intervention. Experiments on 7 stroke patients demonstrate the effectiveness of our approach in predicting tissue fate and its superiority to linear models that are conventionally used.

1 Introduction

Time is a critical factor in the treatment of stroke patients. Early identification of salvageable brain tissue after acute ischemic stroke provides an essential insight during the clinical decisions-making process, and may impact on the long-term patient outcome.

Thrombolytic therapy, which uses specific drugs to break-up or dissolve the blood clot, perfectly illustrates this timing issue. Several studies have demonstrated that thrombolysis applied with recombinant tissue plasminogen activator (rt-PA) is effective for acute ischemic stroke patients when administered within 3h, and 6h of symptom onset for intra-venous, and intra-arterial, respectively. Several clinicians have argued that this arbitrary time frame might be too restrictive for some patients who may potentially have benefited from this therapy but, instead, have been unnecessarily excluded. There is a recognized need for accurate strategies to identify, as early as possible, patients with salvageable penumbral tissue who could benefit from such a therapy.

Diffusion (DWI) and Perfusion-weighted (PWI) magnetic resonance imaging (MRI) provide useful information to distinguish between the irreversibly infarcted region from the penumbral area at early stages. Clinicians have often relied on the visual analysis of the mismatch between DWI and PWI to determine the penumbral tissue. This conventional approach has several limitations: increased diffusion signals can be reversible, and the determination of the threshold of critical perfusion by PWI is still a matter of ongoing debate [1].

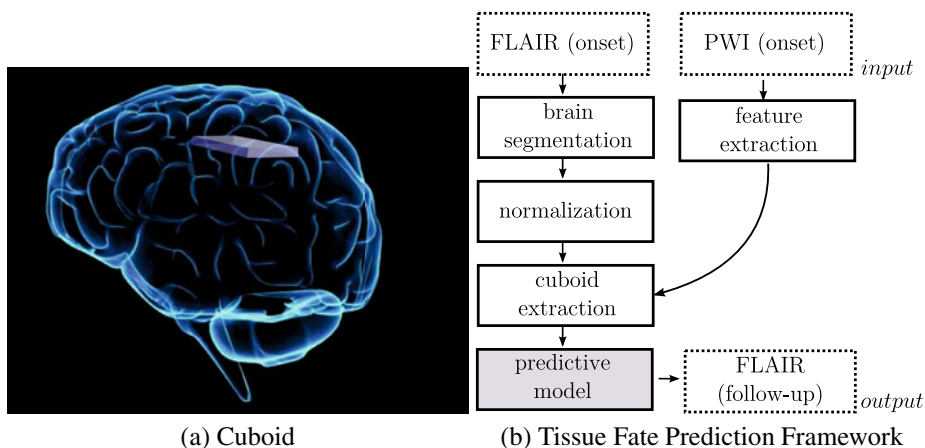


Fig. 1. (a) Illustration of a cuboid extracted from the brain volume. A cuboid describes the local intensity of the volume and captures spatial correlation between voxels. Our framework samples a large number of overlapping cuboids to predict tissue fate. (b) In the proposed prediction framework, the skull is first stripped from the Fluid-attenuated inversion recovery sequence (FLAIR) and the result is normalized. In parallel, Perfusion weighted images (PWI) are processed to extract multi-modal perfusion features (Tmax, MTT, and TTP). Assuming that perfusion features and FLAIR are co-registered, Cuboids are then sampled at the same locations from the different volumes and used as input to the predictive regression model that produces the likely FLAIR image after intervention.

Recently, considerable attention has been given to the development of automatic, quantitative predictive models [23] that can estimate the likely evolution of the endangered tissue. They are expected to outperform the DWI/PWI mismatch approach. Typically, automatic predicting models are trained on a voxel-by-voxel basis by integrating multimodal perfusion information from cases with known follow-up tissue fate. The model can then be used to predict the tissue outcome of newly admitted patients. Methods can be discriminated by the type of classification technique they employ. Current algorithms span from simple thresholding, Gaussian, generalized linear model (GLM) [45], to Gaussian Mixture Models (GMM) [6]. A recent study suggests that taking into account spatial correlation between voxels improves the prediction accuracy [7]. Such a correlation is usually not used by voxel-based models.

This paper introduces a quantitative predictive model of tissue fate that improves current approaches in two ways. First, adaptive, overlapping *cuboids* (*i.e.* rectangular volumes) (Fig 1(a)) that implicitly capture local spatial correlation between neighboring voxels are exploited. Their size is automatically determined during learning. Unlike previous attempts [7] to model spatial correlation, the use of cuboids is not bounded to a particular classification or regression method to infer tissue fate. Our approach can also be seen as a 3D generalization of subwindows-based stochastic methods [8] that have been successfully used on 2D image classification problems. The second contribution of this work is to exploit a spectral regression [9] to model the nonlinear relation that exists between the multi-modal perfusion features and the tissue fate. This contrasts with

Linear or Gaussian models used in previous approaches. The framework combines the local tissue information available immediately after onset, in terms of fluid attenuated inversion recovery (FLAIR), with multi-modal perfusion features (Tmax, MTT, TTP) by extracting cuboids at similar locations. The output is the local FLAIR intensity of the tissue approximately 4-days after a successful recanalization intervention.

2 Methods

2.1 Patients, and MRI Data Acquisition

MRI data were collected from patients identified with symptoms of acute stroke and admitted at the University of California-Los Angeles Medical Center within 6 hours of symptom onset. The use of these data was approved by the local Institutional Review Boards.

Inclusion criteria for this study were: (1) presenting symptoms suggestive of ischemic stroke, (2) last known well time within 6 hours, (3) multimodal MRI (including MRI/PWI) of the brain obtained within 6 hours of last known well time performed before recanalization therapy, (4) final diagnosis of ischemic stroke, and (5) follow-up scan that exhibits a Partial (TIMI¹ grade 2) or complete (TIMI grade 3) recanalization achieved in approximately four days.

A fluid attenuated inversion recovery (FLAIR) sequence was also recorded. FLAIR sequence is acquired with long TI to remove the effects of fluid from the resulting images. It is an excellent technique to identify brain tissue damage since it can reveal early parenchymal changes associated with ischemia and prior cerebral lesions [10].

2.2 Prediction Framework

The prediction framework introduced in this paper relies essentially on a regression model that is learned in a supervised fashion from a set of training images with known outcome. Once a model has been trained, it can be used to predict the tissue outcome, in terms of followup fluid-attenuated inversion recovery (FLAIR) intensity, on new cases. The framework (Figure 1(b)) consists of different modules, described in the next subsections, that pre-process the data to make them suitable for the predictive model.

Automatic Brain Segmentation. During learning, the framework requires FLAIR images acquired immediately after onset and at followup to be co-registered. The skull and non-brain tissue might interfere with the registration process and have therefore to be stripped from the images.

To perform this brain extraction step, we use the efficient FSL Brain Extraction Tool (BET) [11] that is integrated into the UCLA Loni Pipeline software. BET exploits the following procedure to remove the unwanted structure: it estimates an intensity threshold to discriminate between brain/non-brain voxels, determines the center of gravity of the head, defines a sphere based on the center of gravity of the volume, and deforms it toward the brain surface.

¹ TIMI stands for Thrombolysis in Myocardial Ischemia.

FLAIR Image Normalization. Because FLAIR images were acquired with different settings, and originated from different patients, their intensity value was not directly comparable. To allow for inter-patient comparisons, FLAIR images were normalized with respect to the average intensity values in normal-appearing white matter on the side opposite to the stroke side. The normal-appearing white matter was delineated manually by an experienced researcher both on onset and follow-up images.

Image Registration. FLAIR images at onset and followup were automatically registered for each patient independently. Registration is necessary because the outcome (measured as a voxel value in the followup image) of the extracted cuboids has to correspond to the same anatomical location in the different volumes.

Because the intensity of FLAIR images may present large variations between onset and followup (due to changes in the tissue perfusion caused by the stroke), a direct image registration method using voxels values would fail to accurately align the volumes. Instead, we use a straightforward automatic approach that first computes a mask of the volume using a K-Means clustering for each volume. Then a conventional affine registration and a cross-correlation fitness function are exploited to register the followup FLAIR on the original FLAIR volume at onset.

MRI Feature Maps. For each patient, PWI images were processed with a software developed at UCLA, the Stroke Cerebral Analysis 2 (SCAN 2) package. The software is written with the Interactive Data Language produced by ITT Visual Systems.

Following acquisition of the perfusion sequence, perfusion maps can be generated automatically. These perfusion parameter maps include cerebral blood flow (CBF), cerebral blood volume (CBV), mean transit time (MTT), and time-to-peak (TTP) of the residue function (Tmax), following deconvolution of an arterial input function identified from the contralateral middle cerebral artery and tissue concentration curves.

Multi-modal Cuboid Extraction. For training, we exploit a set of FLAIR images F at onset, their corresponding perfusion feature maps $M_{1\dots n_M}$, and the co-registered followup FLAIR images F' acquired approximately 4 days after intervention.

The dataset that is used in our experiments to learn and to evaluate the predictive model is created by extracting cuboids of fixed size $w \times l \times d$ among these training images. Each cuboid $c \in \mathbb{R}^s$ is described by its raw voxel values, yielding a vector of $s = w \times l \times d$ numerical attributes. Our method extracts a large number of possibly overlapping cuboids at random positions from training images. Their position is randomly chosen so that each cuboid is fully contained in the volume. In practice, given a sampled location $\{i, j, k\}$, we extract a cuboid c_F in the FLAIR image at $F(i, j, k)$, and a corresponding cuboid in each perfusion map $c_{1\dots n_M}$ at $M_{1\dots n_M}(i, j, k)$. Then we merge them into a multi-modal cuboid $x = \{c_F, c_{M,1}, \dots, c_{M,n_M}\}$ that corresponds to the concatenation of the cuboids extracted at the same location in the different volumes. Each multi-modal cuboid is then labeled by the intensity value y of the central voxel in the corresponding follow-up FLAIR image $y = F'(i, j, k)$. The input of the final training dataset consists in the set of cuboids $x \in X$, and the outputs are the corresponding followup FLAIR voxel intensities $y \in Y$.

The optimal size of the cuboids was automatically selected so that it maximized the prediction accuracy in a leave-one-out crossvalidation experiment (Section 3). Note that

the sampling of the volume in terms of a large number of cuboids can be seen as a 3D generalization of subwindows-based methods [8] that have been widely used on 2D image classification problems.

2.3 Regression-Based Predictive Model

Our predictive model takes the form a regression model that maps the tissue outcome $y \in Y$, described in terms of the voxel intensity in the followup FLAIR image, as a function of multi-modal cuboids $x \in X$ extracted at the same locations.

Ideally, the estimation of the regression model $y = f(x)$ is made from a set of training samples that cover the data space in a somehow uniform fashion. As it was mentioned in a recent study [12], a different number of infarcting and noninfarcting voxels might impact negatively the overall performance of the system. Naturally, the brain volume usually contains a much larger number of noninfarcted voxels. Therefore, we resample the set of multi-modal cuboids so that an equal number of infarcting and noninfarcting cuboids are used for training.

In the context of pattern recognition, the literature of regression analysis has been particularly proficient in the last couple of years with the emergence of robust, nonlinear methods. Our choice goes for the Spectral Regression (SR) analysis [9] that will be evaluated against Multi-Linear Regression [13].

Multiple Linear Regression. A common way to obtain a regression model is to perform a Multiple Linear Regression (MLR) analysis [13]. The intuition behind this technique is to fit a model such that the sum-of-squares error (SSE) between the observed and the predicted values is minimized.

Spectral Regression Analysis. The Spectral Regression analysis (SR) [9] is a recent method which combines spectral graph analysis and standard linear regression. The main idea consists in finding a regression model that has similar predictions $\hat{y}_i \in \hat{Y}$ for data samples $x_i \in X$ that are close (*i.e.* that are nearest neighbors in a graph representation), such that the following measure ϕ is minimized:

$$\phi = \sum_{i,j=1}^n (\hat{y}_i - \hat{y}_j)^2 W_{i,j} \quad (1)$$

where $W \in \mathbb{R}^{n \times n}$ is the affinity (*i.e.* item-item similarity) matrix that associates a positive value to $W_{i,j}$ if the samples x_i, x_j belong to the same class.

Spectral Regression is typically a linear regression algorithm. However, it can easily be extended to nonlinear problems by using a kernel projection. Also known as the “kernel trick”, this technique allows to use a linear regression technique to solve a nonlinear problem by mapping the observations into a higher-dimensional space, where the linear regression is subsequently used. In our framework, a Radial Basis Function (RBF) kernel is used as a projection matrix,

$$K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}, \gamma > 0 \quad (2)$$

We further refer to this technique as the Kernel Spectral Regression (KSR).

3 Experiments

The experiments aim at evaluating the capacity of the framework to accurately predict tissue fate in terms of followup FLAIR intensity values. At the same time, this section also provides a comparative analysis of the different regression methods that have been presented previously. Their strength is quantified on a tissue fate prediction task by measuring their accuracy, generalization power on new patients, sensitivity to the number of training samples, and cuboid size. Evaluations are conducted on a database made of 7 patients (Section 2.1).

The following acronyms LIN, KSR will be used along this section to refer to Multiple Linear and Kernel Spectral Regression methods, respectively.

3.1 Intra- vs. Inter-patient Prediction

In this experiment, we evaluate the accuracy of the KSR regression method to predict tissue fate. Two experimental protocols are used. The first trains a predictive model for each patient separately such that it is built from a subset of its own data. These models amount to solve a fitting problem and are referred to as the intra-patient models. A more realistic, but also more challenging task is to build the model by excluding data from the patient intended to be evaluated. This inter-patient evaluation protocol is achieved using a standard leave-one-out crossvalidation.

In both intra- and inter-patient evaluations, T_{max} perfusion maps are used in conjunction of the FLAIR images at onset as the input data x to a KSR regression predictive model, and a varying number multi-modal annotated cuboids are sampled at random locations in images. The accuracy is expected to be better as the number of labelled examples increases, from 60 to 3300.

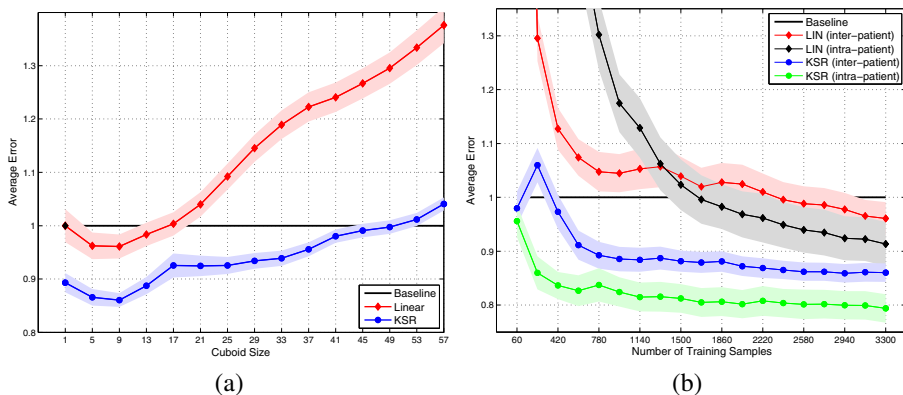


Fig. 2. (a) Effect of the cuboid size on the average prediction error for LIN and KSR regression models using a leave-one-out crossvalidation strategy on less than 3000 samples. (b) Effect of the number of training samples on the average prediction error (Eq. 3) and standard deviation for a KSR regression model either trained from a subset of the own data of each patient (Intra-patient), or using a leave-one-out crossvalidation strategy so that the model used to predict a patient is trained from the data of other patients only. Note that T_{max} was used as the perfusion maps.

For comparison matter, the average prediction error is reported between the voxel values in the followup FLAIR y_i and the prediction \hat{y}_i ,

$$e = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (3)$$

We normalized all the reported errors using the Multiple Linear method with a cuboid size of 1 voxel as baseline.

Figure 2(b) shows that, when trained with 3300 training samples, the intra-patient models reach an error rate of $e_{KSRi} = 0.79$ for KSR and $e_{LINi} = 0.91$ for LIN. For the inter-patient model, KSR, which obtains an error rate of $e_{KSR} = 0.86$, also outperforms the LIN method $e_{LIN} = 0.96$. The inter-patient results demonstrate that the KSR models generalize very well to new data and do not seem to overfit the data. Inter-patient models keep improving as the number of training data increases. Figure 4 illustrates the predicted tissue fate for the first 4 patients.

3.2 Comparative Analysis of Regression Methods

In this section, we evaluate the accuracy of each regression method (LIN, KSR) to predict tissue fate. Similarly to the previous experiment, we use a leave-one-out cross-validation protocol (inter-patient) and report the average prediction error as a function

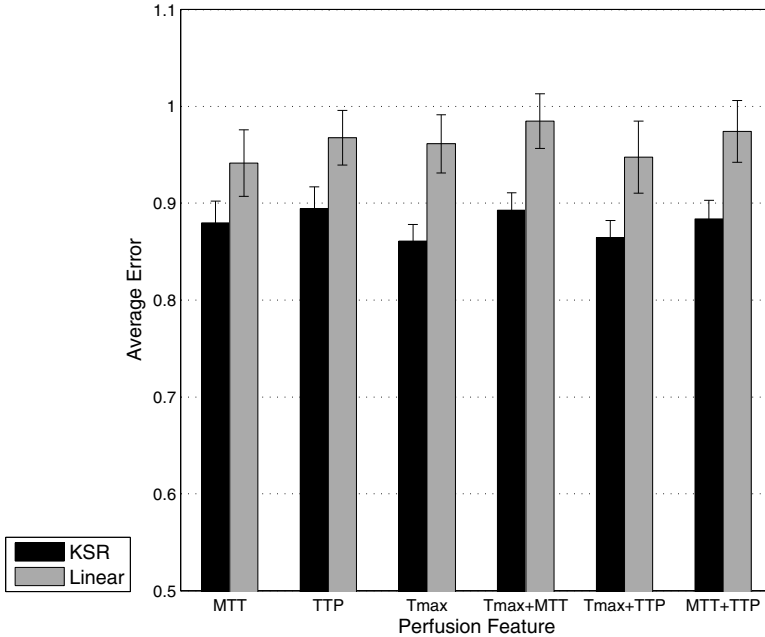


Fig. 3. Average prediction error (Eq. 3) and standard deviation for LIN, KSR regression models using a leave-one-out crossvalidation strategy for different combinations of perfusion maps used as input. When combined with Tmax feature maps, KSR regression method achieves the best prediction results.

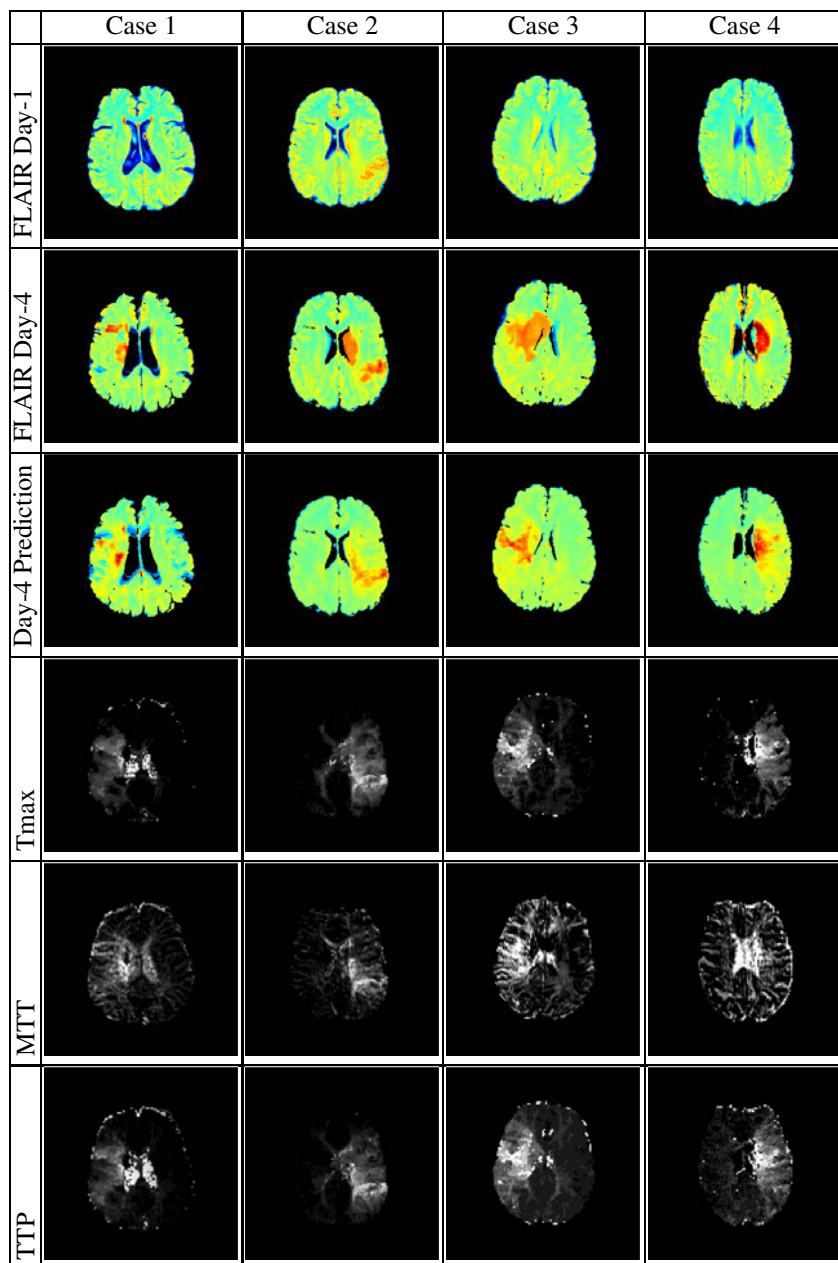


Fig. 4. Prediction results for the first 4 patients using our framework. The rows respectively correspond to the FLAIR at onset, followup FLAIR at day 4, the corresponding prediction obtained from our model, and the Tmax, MTT, and TTP perfusion maps. The predictions were obtained using a KSR inter-patient model trained with Tmax perfusions at onset, 3300 multi-modal cuboids of size $9 \times 9 \times 1$.

of the number of training samples. The multi-modal cuboids are extracted from FLAIR images and Tmax perfusion maps, both at onset.

Results, that are depicted in Figure 2(b), shows that KSR, $e_{KSR} = 0.86$ outperforms linear regression models, $e_{LIN} = 0.96$. This suggests that the relation between the multi-modal cuboid x and the tissue fate y is a non-linear one and cannot be fully captured by linear approaches.

These results are confirmed in Figure 2(a) which illustrates the impact of the cuboid size on the overall performance using approximately 3000 training samples. For images of size $256 \times 256 \times 20$, the optimal cuboid size is $9 \times 9 \times 1$ no matter which regression method is used. Note that the best thickness of the cuboids was equal to 1 slice, because of the poor resolution of perfusion images. A better resolution could probably lead to a more precise estimation of the slice thickness and offer better prediction results.

3.3 Comparing Perfusion Feature Maps

The experimental protocol is similar to the one used for the inter-patient evaluation. The average prediction error of FLAIR intensity (Eq. 3) is measured using a leave-one-out cross-validation. Training of the linear and KSR regression model is performed on 3300 multi-modal cuboids sampled from different perfusion maps (MTT, TTP, and Tmax). Figure 3 reports the average prediction error for linear and KSR regression models for different combination of perfusion maps used as input. When combined with Tmax, KSR achieves the best prediction results. Besides these perfusion maps, we also conducted some experiments on CBV, and CBF perfusion maps. However, for several patients, some of the maps could not be computed accurately and therefore lead to inconsistent results. Future works with more patients will investigate if the use of CBF and CBV could actually improve the prediction accuracy of our model.

4 Conclusion

We described a framework to predict the likely outcome of tissue fate for ischemic stroke patients. The proposed approach exploits a nonlinear regression model that is able to predict the tissue fate on new patients. Our experimental results validate the method by improving those obtained by a linear, voxel-based approaches. This can be explained by two reasons. First, the better performance using a KSR regression approach may rely on the fact that this technique can robustly capture the nonlinear relation between FLAIR images, perfusion maps at onset, and followup FLAIR. Second, the use of optimized cuboids improves a voxel-based approach by at least 1 percent regardless the regression technique that is used. This can be explained by the fact that cuboids implicitly represent spatial correlation information between voxels.

Acknowledgments

This work was supported by the National Institutes of Health [K23 NS054084 and P50 NS044378 to D.S.L.], and [R21-NS055998, R21-NS055045, R21-NS059797, R01 NS054881 to X.H.]

References

1. Heiss, W., Sobesky, J.: Can the penumbra be detected: MR versus PET imaging. *J. Cereb Blood Flow Metab* 25, 702 (2005)
2. Shen, Q., Ren, H., Fisher, M., Duong, T.: Statistical prediction of tissue fate in acute ischemic brain injury. *J. Cereb Blood Flow Metab* 25, 1336–1345 (2005)
3. Shen, Q., Duong, T.: Quantitative Prediction of Ischemic Stroke Tissue Fate. *NMR Biomedicine* 21, 839–848 (2008)
4. Wu, O., Koroshetz, W., Ostergaard, L., Buonanno, F., Copen, W., Gonzalez, R., Rordorf, G., Rosen, B., Schwamm, L., Weisskoff, R., Sorensen, A.: Predicting tissue outcome in acute human cerebral ischemia using combined diffusion- and perfusion-weighted MR imaging. *Stroke* 32, 933–942 (2001)
5. Wu, O., Christensen, S., Hjort, N., Dijkhuizen, R., Kucinski, T., Fiehler, J., Thomalla, G., Rother, J., Ostergaard, L.: Characterizing physiological heterogeneity of infarction risk in acute human ischaemic stroke using MRI. *Brain* 129, 2384–2393 (2006)
6. Rose, S., Chalk, J., Griffin, M., Janke, A., Chen, F., McLachan, G., Peel, D., Zelaya, F., Markus, H., Jones, D., Simmons, A., OSullivan, M., Jarosz, J., Strugnell, W., Doddrell, D., Semple, J.: MRI based diffusion and perfusion predictive model to estimate stroke evolution. *Magnetic Resonance Imaging* 19, 1043–1053 (2001)
7. Nguyen, V., Pien, H., Menenzes, N., Lopez, C., Melinosky, C., Wu, O., Sorensen, A., Cooperman, G., Ay, H., Koroshetz, W., Liu, Y., Nuutinen, J., Aronen, H., Karonen, J.: Stroke Tissue Outcome Prediction Using A Spatially-Correlated Model. In: *PPIC* (2008)
8. Maree, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: *CVPR*, vol. 1, pp. 34–40 (2005)
9. Cai, D., He, X., Han, J.: Spectral Regression for Efficient Regularized Subspace Learning. In: *ICCV* (2007)
10. Liebeskind, D., Kidwell, C.: Advanced MR Imaging of Acute Stroke: The University of California at Los Angeles Endovascular Therapy Experience. *Neuroimag. Clin. N. Am.* 15, 455–466 (2005)
11. Smith, S.: Fast robust automated brain extraction. *Human Brain Mapping* 17, 143–155 (2002)
12. Jonsdottir, K., Ostergaard, L., Mouridsen, K.: Predicting Tissue Outcome From Acute Stroke Magnetic Resonance Imaging: Improving Model Performance by Optimal Sampling of Training Data. *Stroke* 40, 3006–3011 (2009)
13. Chatterjee, S., Hadi, A.S.: Influential observations, high leverage points and outliers in linear regression. *Statistical Science* 1, 379–393 (1986)

3D Vector Flow Guided Segmentation of Airway Wall in MSCT

Margarete Ortner¹, Catalin Fetita¹, Pierre-Yves Brillet², Françoise Prêteux³,
and Philippe Grenier⁴

¹ Dept. ARTEMIS, Institut TELECOM / TELECOM SudParis, Evry, France

² Université Paris 13, AP-HP, Avicenne Hospital, Bobigny, France

³ Mines ParisTech, Paris, France

⁴ Université Paris 6, AP-HP, Pitié-Salpêtrière Hospital, Paris, France

Abstract. This paper develops a 3D automated approach for airway wall segmentation and quantification in MSCT based on a patient-specific deformable model. The model is explicitly defined as a triangular surface mesh at the level of the airway lumen segmented from the MSCT data. The model evolves according to simplified Lagrangian dynamics, where the deformation force field is defined by a case-specific generalized gradient vector flow. Such force formulation allows locally adaptive time step integration and prevents model self-intersections. The evaluations performed on simulated and clinical MSCT data have shown a good agreement with the radiologist expertise and underlined a higher potential of the proposed 3D approach for the study of airway remodeling versus 2D cross-section techniques.

1 Introduction

Automated segmentation of multislice computed tomography data (MSCT) of the bronchial wall and its quantification are key stones for the diagnosis of respiratory pathologies specially regarding asthma and chronic obstructive pulmonary disease (COPD). Inflammation and the remodeling of the airway wall are commonly used indicators in clinical practice to rate and classify the severity of the disease. In particular for follow-up studies a quantitative assessment is necessary to evaluate treatment efficiency or progression of the disease. Therefore a great demand for accurate and reproducible measurements of the airway lumen and the wall thickness is expressed. The major difficulties of this challenge lie in small caliber airways, thin or irregular airway walls and the partial volume effect even in larger bronchi.

The existing methods of semi-automated and automated segmentation of the airway wall perform in the bronchus cross-section plane and can be classed in: full-width at half-maximum (FWHM) based [1], pattern-based [2,3] or shape-independent. Although it was shown [4] that FWHM methods yields in an under-segmentation of the airway wall (under-estimation of the wall thickness), their simplicity makes them the most used in practice. Pattern-based segmentation methods show remarkable results on phantom studies or excised animal lungs,

as they rely on the detection of circular or ellipsoidal shapes. Concerning the practical aspect, these methods are generally not accurately applicable on clinical data. More recent techniques which are independent of a shape-specificity are based on a luminal segmentation, a central axis computation followed by a cross-section image reformation. The wall segmentation is then performed in the 2D space. This 3D/2D approach allows measurements independent from the CT acquisition plane at every desired spatial location [5]. Unfortunately the whole process depends on the medial axis estimation, which can be ambiguous in case of high curvature or irregularities. Further issues are the impossibility of cross-section investigation at subdivision areas and sparse measurements along the bronchi.

To overcome the mentioned problems, a purely volumetric segmentation approach is here developed, §2, based on an explicit deformable model which geometry is defined as a triangular mesh of the inner airway wall surface. The deformation dynamics is governed by a simplified Lagrangian law where the force definition exploits a patient-specific generalized gradient vector flow (GGVF) map. Evaluated on both simulated and clinical MSCT data, §3, the proposed approach shows a great potential for the analysis of airway remodeling.

2 GGVF-Guided Deformable Model

The developed deformable model is patient-specific and its geometry is built up based on the 3D segmentation of the airway lumen from MSCT data. For example, one of the approaches discussed in [6] can be applied here. An explicit 3D surface mesh model is then obtained via a marching tetrahedra algorithm [7]. This choice was based on the fact that marching tetrahedra is considered disambigous [8] so that the geometry and topology of the small bronchi are preserved. The mesh resolution (density) is furthermore adapted with respect to the local caliber of the segmented airway lumen. The caliber information is provided by means of a 3D granulometric analysis of the lumen binary object [9]. The model dynamics obey a simplified (zero mass) Lagrange equation of motion:

$$m\ddot{x}_i + \gamma_i\dot{x}_i = F_{ext}(x_i) + F_{int}(x_i), m = 0, \quad (1)$$

where \ddot{x}_i and \dot{x}_i denotes the acceleration and velocity of vertex i and γ_i the damping factor. F_{int} represents internal forces which are derived from the mesh itself to keep a locally smooth shape of the surface. F_{ext} represents external forces which attract the model to special image features. Contrary to generally used deformable model definitions, our model does not evolve along its faces or vertex normals but along a vector field. This vector field, built up by a generalized gradient vector flow (GGVF), prevents self-intersections and allows a variable time step for the integration of eq. [1].

2.1 GGVF

The GGVF is introduced in 1998 by Xu and Prince [10] and is widely used in the segmentation and simulation community. Regarding medical imaging, generally

a 2D GGVF is used, like in vessel boundary detection [11,10], whereas the 3D version is seldom exploited [12,13] due to computation complexity. The GGVF formulation is given by:

$$\mathbf{V} = \iiint_{\Omega} g(|\nabla f|) \cdot |\nabla \mathbf{V}|^2 - h(|\nabla f|) \cdot (\mathbf{V} - \nabla f) \, dx dy dz, \quad (2)$$

where \mathbf{V} defines the GGVF, which is initialized by a gradient field ∇f ; $g(|\nabla f|)$ and $h(|\nabla f|)$ are weighting functions which represent the diffusion and the original data influence. The GVF is a special case with a constant $g(|\nabla f|) = \mu$ and $h(|\nabla f|) = (|\nabla f|)^2$. Our model sticks to the generalized version as the static diffusion may lead to undesired results at segment subdivision, as shown in Fig. 1(a), 1(b) for a simple 2D case. Note that the objective here is to obtain a force field adapted to the model geometry and preventing self-intersections or surface folding during the deformation. We defined [10]:

$$g(|\nabla f|) = e^{-(|\nabla f|/K)^2}, \quad h(|\nabla f|) = 1 - g(|\nabla f|), \quad (3)$$

where K is set to 2.0 to ensure a diffusion at all points, while still enhancing large gradient values on strong edges. Diffusion strength depending on the initializing gradient is shown in Fig. 1(c).

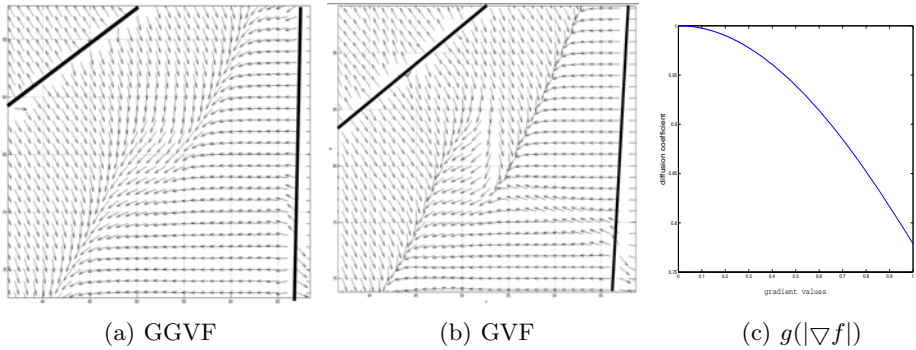


Fig. 1. (a),(b) Normalized vector field computed at a branch bifurcation (branch axis shown in black) obtained with the GGVF (eq. 3) and the GVF, respectively; (c) Diffusion coefficient depending on the gradient force

The GGVF of eq. 2 is obtained as the equilibrium solution of the following discrete PDE [10]:

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t \cdot g(|\nabla f|) \cdot \nabla^2 \mathbf{v}^t - h(|\nabla f|) \mathbf{v}^t + h(|\nabla f|) \cdot \nabla f \quad (4)$$

where ∇^2 denotes the Laplacian operator. To speed up the computation of the GGVF, therefore making it usable for realistic applications, a **semi-implicit form** (eq. 5), promises similar results in less time. Such formulation exploits the

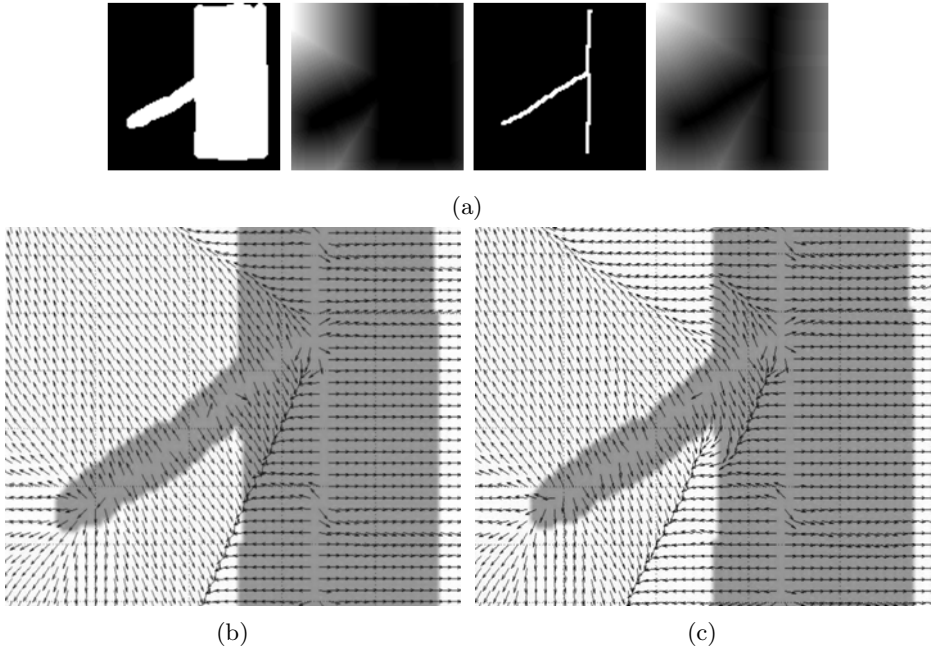


Fig. 2. 2D illustration of GGVF depending on the initialization: (a) from left to right: lumen segmentation, $d(lumen)$, lumen axis, $d(axis)$; (b) normalized GGVF field with $\alpha : \beta = 1:10$, (c) normalized GGVF field on a $\alpha : \beta = 10:1$ weighted initialization. The lumen position is superimposed in gray, showing that surface deformation will penetrate in the lumen near the bifurcation in (b).

information already computed at the current time step, which allows a faster diffusion over the space. The transformation into a semi-implicit form uses values of neighbors which are already calculated at the iteration $t+1$ and rewrites the formulation in eq. 5.

$$\mathbf{v}^{t+1} = [\mathbf{v}^t + \Delta t \cdot g(|\nabla f|) \nabla^2 \mathbf{v}^t + h(|\nabla f|) \nabla f] / [1 + (6 \cdot g(|\nabla f|) + h(|\nabla f|)) \Delta t], \quad (5)$$

where $\nabla^2 \mathbf{v}_{x,y,z}^t$ at each point (x, y, z) is estimated from the 6-connected causal neighborhood:

$$\begin{aligned} \nabla^2 \mathbf{v}_{x,y,z}^t = & \mathbf{v}_{x,y,z-1}^{t+1} + \mathbf{v}_{x,y-1,z}^{t+1} + \mathbf{v}_{x-1,y,z}^{t+1} + \\ & \mathbf{v}_{x,y,z+1}^t + \mathbf{v}_{x,y+1,z}^t + \mathbf{v}_{x+1,y,z}^t - 6 \cdot \mathbf{v}_{x,y,z}^t, \end{aligned} \quad (6)$$

The **initialization of the GGVF** plays an important role in achieving the desired properties of the external force field. In order to ensure a patient-specific behavior of the model dynamics, we use a weighted sum of two gradient fields computed on distance functions evaluated with respect to the segmented airway lumen $d(lumen)$ and the lumen centerline $d(axis)$.

$$\mathbf{v}^0 = \frac{\alpha \cdot \nabla d(\text{lumen}) + \beta \cdot \nabla d(\text{axis})}{\alpha + \beta} \tag{7}$$

The choice of α, β coefficients should respect a higher weight for the lumen component, in order to strengthen the influence of the model shape on the GGVF field, namely at subdivisions involving segments of different calibers. Fig. 2 illustrates this situation by comparing axis versus lumen reinforcements. Particularly, the GGVF field on this bifurcation shows the required deviation to obtain a merging flow outside of the luminal area. The solution $\beta = 0$ is not appropriate in our case as the initialization of the model will occur exactly on the luminal surface or slightly inside, where the GGVF orientation would be disturbed because of the image discretization. A value $\beta > 0$ guarantees the presence of a vector flow inside the lumen, correctly oriented towards the airway wall, which prevents surface folding during the deformation.

2.2 Model Dynamics

Internal and external forces guide and control the deformation of the model. The force competition for model displacement is applied at each mesh vertex x_i in the direction of the GGVF field, except for the surface smoothing forces, which depend only on the model geometry. Such displacement along the GGVF field prevents self-intersections of the model.

The **external forces** are defined as $F_{ext}(x_i) = \frac{\mathbf{V}(x_i)}{\|\mathbf{V}(x_i)\|} \cdot (f_g(x_i) + f_b(x_i))$. Their orientation is given by the GGVF field, \mathbf{V} , and their magnitude imposed by the image gradient component f_g and a balloon force f_b . The gradient force f_g attracts the mesh to strong edges in the image and is computed by a Canny-Deriche [14] operator. The second force f_b hinders an adherence to the inner bronchial wall and is driven by the image intensity values. All forces are normalized with respect to the maximal gray level in the image volume \hat{I} :

$$f_g(x_i) = \frac{\nabla(G_\sigma * I)(x_i)}{\hat{I}} \cdot \frac{\mathbf{V}(x_i)}{\|\mathbf{V}(x_i)\|}, \quad f_b(x_i) = \frac{I(x_i)}{\hat{I}}, \tag{8}$$

where I denotes the original image intensity and G_σ stands for Gaussian smoothing kernel with a standard deviation σ .

The **internal forces** are a combination of elastic and regularization forces. The elastic force f_e inhibits a progression of vertices into airway-vessel contact regions. Its direction is given by the GGVF field, while its magnitude penalizes local variations of the wall thickness with respect to the average thickness computed in a cross-section ring, eq. 9, Fig. 3. The elastic force is the counterpart to the external forces and restrains the motion along the GGVF field.

$$\mathbf{f}_e(x_i) = -\left(\frac{d(x_i, M_0)}{d_\mu(x_i, M_0)} - 1\right) \cdot \frac{\mathbf{V}(x_i)}{\|\mathbf{V}(x_i)\|}, \quad d_\mu = \sum_{j \in R_i} \frac{1}{|R_i|} d(x_j, M_0), \tag{9}$$

where $d(x_i, M_0)$ stands for the Euclidian distance between the vertex x_i and the lumen surface M_0 .

The classic regularization term [15], \mathbf{f}_r , directs vertices towards the local tangent plane to optimize the surface curvature. It depends on the local neighborhood and is only applied as long as a deformation at this point takes place, that is, when a minimum displacement occurs under the action of F_{ext} and \mathbf{f}_e :

$$\frac{1}{k_g} \cdot f_g + \frac{1}{k_b} \cdot f_b + \frac{1}{k_e} \cdot (\mathbf{f}_e \cdot \frac{\mathbf{V}}{\|\mathbf{V}\|}) > 0, \quad (10)$$

where k_g, k_b, k_e are elastic coupling constants.

The displacement of the vertex x_i under the action of external and internal forces (eq. 10) is given by the elastic coupling constants imposed on force weighting:

$$\Delta \mathbf{x}_i = (\alpha_g f_g(x_i) + \alpha_b f_b(x_i)) \cdot \frac{\mathbf{V}(x_i)}{\|\mathbf{V}(x_i)\|} + \alpha_e \mathbf{f}_e(x_i) + \alpha_r \mathbf{f}_r(x_i). \quad (11)$$

Adaptive mesh resolution is possible through remeshing, which demands a definition of maximal and minimal edge length depending on the spatial location, naming the smaller or larger bronchi. This ensures computational efficiency, as a coarser mesh is defined on larger, less interesting parts, and high segmentation sensitivity is preserved on thin structures. The constraint of the edge length ξ depends on the Euclidean distance between the vertex of the model and the correlating point on the axis $\xi(d(va, v)) \leq d(v, vn) \leq \lambda \xi(d(va, v))$, where vn denotes the neighbors of vertex v and va its corresponding point on the central axis. The mesh resolution adaptation is performed at each iteration step using classic mesh optimization procedures (edge-subdivision, edge-inversion, edge-collapse) [16].

Prevention of self-intersection is the strength of this method as our special interest is a reliable segmentation of the bronchial wall at airway subdivisions. In order to illustrate this aspect, Fig. 4 shows two deformation models with the same parameter settings. Fig. 4(a) displays clean bifurcations due to GGVF guidance, whereas in Fig. 4(b) the deformation is directed by the mesh model normals. In the second case, the two bifurcations show not only self-intersections, but even growth inside the model, both unacceptable for any measurement or flow simulation. This contribution is important as it allows the analysis of airway wall remodeling at subdivision regions which is not possible with 2D techniques.

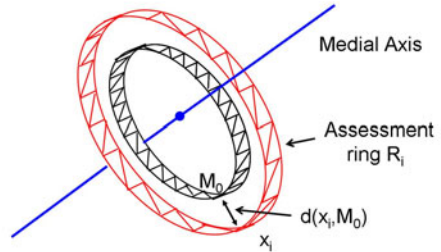


Fig. 3. Parameters for f_e definition

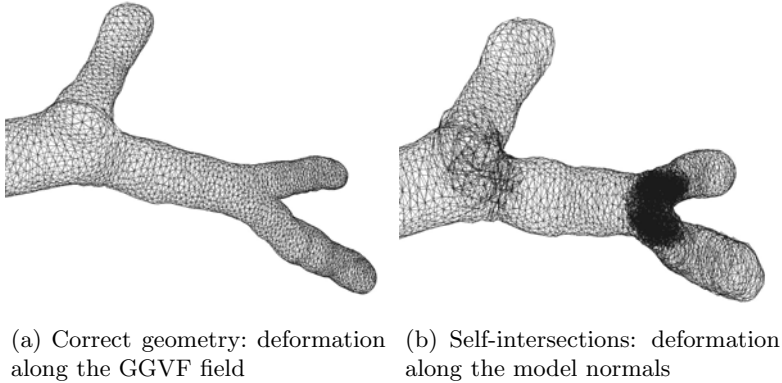


Fig. 4. Deformable models according to the imposed motion vector field

3 Results

The developed deformable model was quantitatively evaluated on a synthetic simulated thorax MSCT data of perfectly known parameters and visually assessed on clinical data. The parameters of §2 were set according to studies performed in 2D or on training cases: $\alpha = 10\beta$, $\alpha_g = \frac{1}{k_g} = 2$, $\alpha_b = \frac{1}{k_b} = 3$, $\alpha_e = \frac{1}{k_e} = 8$, $\alpha_r = \frac{1}{k_r} = 0.5$.

3.1 Quantitative Evaluation through a Ground Truth Model

A simulated MSCT data was obtained from another study [17]. It corresponds to a thickened-wall airway tree built up as a mesh model of the inner/outer surfaces (Fig. 5(a)) based on real clinical data. The synthetic MSCT images (Fig. 5(b)) are obtained through a simulation of the CT acquisition taking into account the reference airway mesh model and the lung texture properties, either simulated or extracted from original data. The synthetic MSCT data is segmented by the proposed approach and the result is compared with the reference ground truth mesh in terms of Hausdorff distance computed in the neighborhood of each vertex. Fig. 5(d) and Fig. 5(e) illustrate in color codes the error obtained for the inner and outer surfaces. As expected, most disagreements occur in the extrapulmonary regions (trachea, main and lobar bronchi) where there is no intensity differentiation between airway wall and the surrounding tissue. Even taking these parts into account, the absolute estimation error (mainly due to vascular contact) is less than 1.17 mm, whereas the error for segmental and more distal bronchi is less than 0.56 mm.

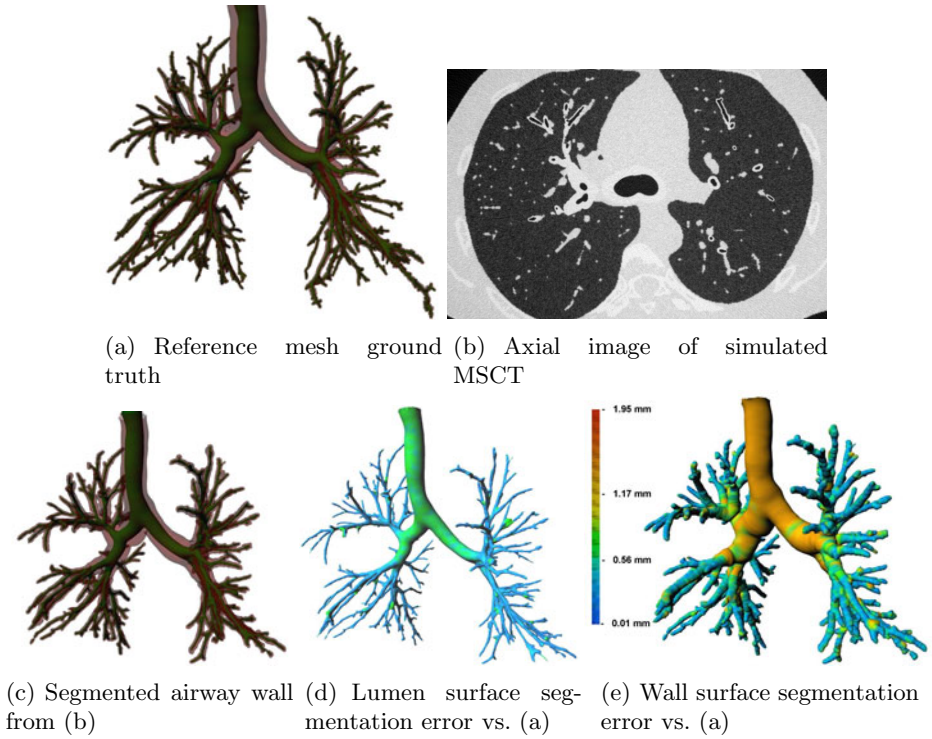


Fig. 5. Approach evaluation on a simulated MSCT data set

3.2 Qualitative Evaluation on Clinical Data

MSCT data acquisitions of 15 clinical cases of mild and severe asthmatics with thin collimation and sharp reconstruction kernels were used as input to our approach. Some of the resulting segmentations are shown in Fig. 6.

The segmentation accuracy was evaluated in consensus by two experienced radiologists which analyzed cross-sections of the composed original-segmented data. The visual assessment was performed both in the axial plane (Fig. 6(d)) and in bronchus cross-section planes for airways selected beyond the segmental level (Fig. 6(e)). The airway selection exploited the central axis and was restricted to regions outside bronchial subdivisions (up to 100 analysis points per patient). The qualitative investigation showed a good agreement between the segmentation result and the clinical expertise. Moreover, the advantage of the 3D segmentation of the airway wall with respect to the 2D approaches is the possibility of investigating the subdivision regions in terms of bronchial remodeling. An appropriate visual assessment of the 3D airway wall thickness may consist of a color coding based on the Hausdorff distance computed between the inner and outer wall surface meshes (Fig. 6(f)). Such visualization will facilitate the analysis of bronchial remodeling in follow-up studies.

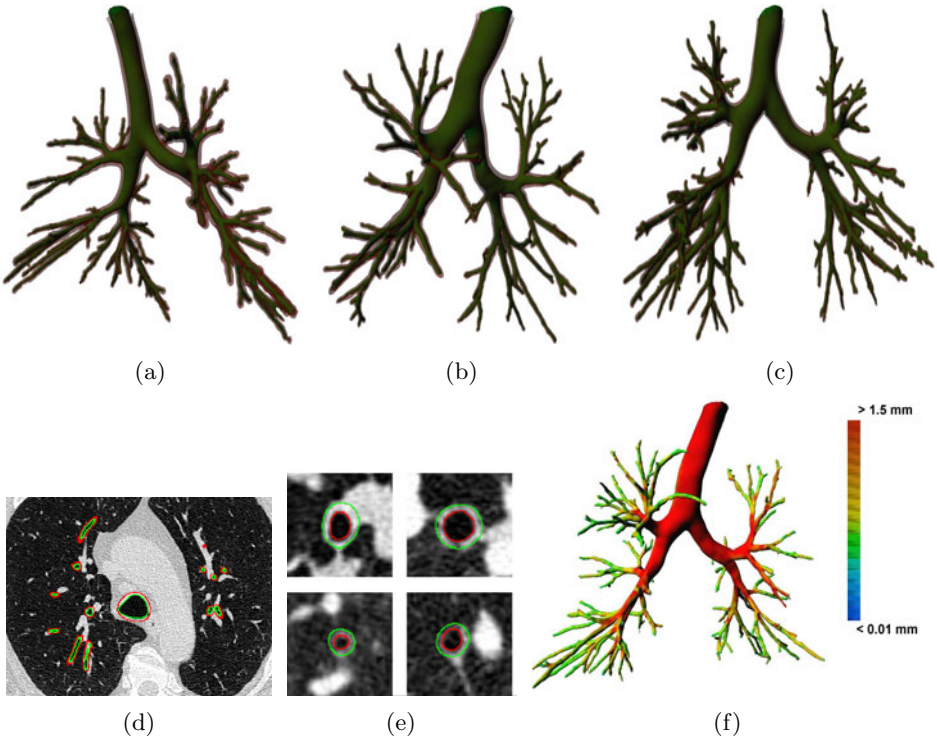


Fig. 6. Example of airway wall segmentation results on mild asthmatic patients (a)-(c); (d),(e) axial and airway cross-section analysis of the segmented data; (f) color coding of the airway wall thickness

4 Discussion and Conclusion

The proposed airway wall segmentation method presents several advantages over the state of the art techniques. First, it is fully automatic and performs in the 3D space allowing a better profit from the structure continuity. Second, it is applied to the whole bronchial tree by exploiting a patient-specific deformable model built up from the airway lumen. The deformation field is computed over a generalized gradient vector flow adapted to the airway geometry, which avoids self-intersections of the model. Third, it makes the analysis of bronchial subdivisions possible, which could be a key issue for the study of airway remodeling in asthma and COPD. Finally, the resulting model provides a link with the functional investigations, via computational fluid dynamics simulations.

The perspective of this work is twofold. On one hand, a study of the robustness with respect to the CT acquisition protocol and the model parameter range will be performed. On the other hand, an evaluation of the gain in sensitivity with respect to 2D segmentation approaches for airway remodeling assessment (namely by including the information at subdivisions) is to be done in clinical trials.

References

1. Matsuoka, S., Kurihara, Y., Nakajima, Y.: Serial change in airway lumen and wall thickness at thin-section CT in asymptomatic subjects. *Radiology* 234, 595–603 (2005)
2. King, G.G., Muller, N., Whittall, K., et al.: An analysis algorithm for measuring airway lumen and wall areas from high-resolution computed tomographic data. *Am. J. of Respiratory and Critical Care Medicine* 161, 574–580 (2000)
3. Wiemker, R., Blaffert, T., Buelow, T., et al.: Automated assessment of bronchial lumen, wall thickness and bronchoarterial diameter ratio of the tracheobronchial tree using high-resolution CT. *Int. Congress Series*, vol. 1268, pp. 967–972 (2004)
4. Hoffman, E., Reinhardt, J., Sonka, M., et al.: Characterization of the interstitial lung diseases via density-based and texture-based analysis of computed tomography images of lung structure and function. *Acad. Rad.* 10, 1104–1118 (2003)
5. Saragaglia, A., Fetita, C., Prêteux, F., et al.: Accurate 3D quantification of bronchial parameters in MDCT. In: *Proc. SPIE Mathematical Methods in Pattern and Image Analysis*, vol. 5916, pp. 323–334 (2005)
6. Lo, P., van Ginneken, B., Reinhardt, J., de Bruijne, M.: Extraction of airways from CT (2009)
7. Chan, S.L., Purisima, E.O.: A new tetrahedral tessellation scheme for isosurface generation. *Computers & Graphics* 22, 83–90 (1998)
8. Newman, T.S., Yi, H.: A survey of the marching cubes algorithm. *Computers & Graphics* 30, 854–879 (2006)
9. Fetita, C., Ortner, M., Brillet, P.Y., Hmeidi, Y., Preteux, F.: Airway shape assessment with visual feed-back in asthma and obstructive diseases, vol. 7652, p. 76251E (2010)
10. Xu, C., Prince, J.: Generalized gradient vector flow external forces for active contours. *Signal Processing* 71, 131–139 (1998)
11. Kim, H., Kim, H., Ahn, C., et al.: Vessel Boundary Detection or its 3D Reconstruction by Using a Deformable Model (GVF Snake), vol. 27, pp. 3440–3443 (2005)
12. Bauer, C., Bischof, H.: Extracting curve skeletons from gray value images for virtual endoscopy, pp. 393–402 (2009)
13. Chen, X., Teoh, E.: 3D object segmentation using B-Surface. *Image and Vision Computing* 23, 1237–1249 (2005)
14. Deriche, R.: Optimal edge detection using recursive filtering. In: *Proc. of the 1st Int. Conf. on Computer Vision*, vol. 1, pp. 501–505 (1987)
15. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum* 21, 109–218 (2002)
16. Hoppe, H., DeRose, T., Duchamp, T., et al.: Mesh optimization. In: *SIGGRAPH 1993: Proc. Computer graphics and interactive techniques*, pp. 19–26 (1993)
17. Ortner, M., Fetita, C., Brillet, P.Y., Preteux, F., Grenier, P.: Ground truth and CT image model simulation for pathophysiological human airway system, vol. 7625 (2010) 76252K

Graph-Based Segmentation of Lymph Nodes in CT Data

Yao Wang^{1,3} and Reinhard Beichel^{1,2,3}

¹ Dept. of Electrical and Computer Engineering

² Dept. of Internal Medicine

³ The Iowa Institute for Biomedical Imaging
The University of Iowa, Iowa City, IA 52242, USA
`reinhard-beichel@uiowa.edu`

Abstract. The quantitative assessment of lymph node size plays an important role in treatment of diseases like cancer. In current clinical practice, lymph nodes are analyzed manually based on very rough measures of long and/or short axis length, which is error prone. In this paper we present a graph-based lymph node segmentation method to enable the computer-aided three-dimensional (3D) assessment of lymph node size. Our method has been validated on 22 cases of enlarged lymph nodes imaged with X-ray computed tomography (CT). For the signed and unsigned surface positioning error, the mean and standard deviation was 0.09 ± 0.17 mm and 0.47 ± 0.08 mm, respectively. On average, 5.3 seconds were required by our algorithm for the segmentation of a lymph node.

Keywords: lymph node segmentation, graph-based segmentation.

1 Introduction

The lymphatic system of the human body is a component of the immune system that plays an important role in dealing with viruses, bacteria, and illnesses. Multidetector computed tomography (CT) has become the primary lymph node imaging modality in clinical routine and offers excellent spatial resolution for measuring lymph nodes [9]. Assessment of the condition of lymph nodes is utilized for diagnosis, monitoring, and treatment of diseases like cancer. In current clinical practice, lymph nodes are analyzed manually based on very rough measures of long and/or short axis length. The accuracy and reproducibility of size measurements is critical for determining response to therapy in clinical practice and informed research studies [9]. A true 3D quantification of lymph nodes size promises to be more accurate and reproducible in longitudinal studies, since it obviates the need to explicitly determine lymph node axes, which is a source of potentially large errors.

Lymph node segmentation in volumetric CT data is a challenging task due to low contrast to adjacent structures and potentially inhomogeneous density-values (Fig. 1). So far, computer-aided 3D segmentation approaches have not been utilized routinely in research or clinical practice. It is generally accepted that a

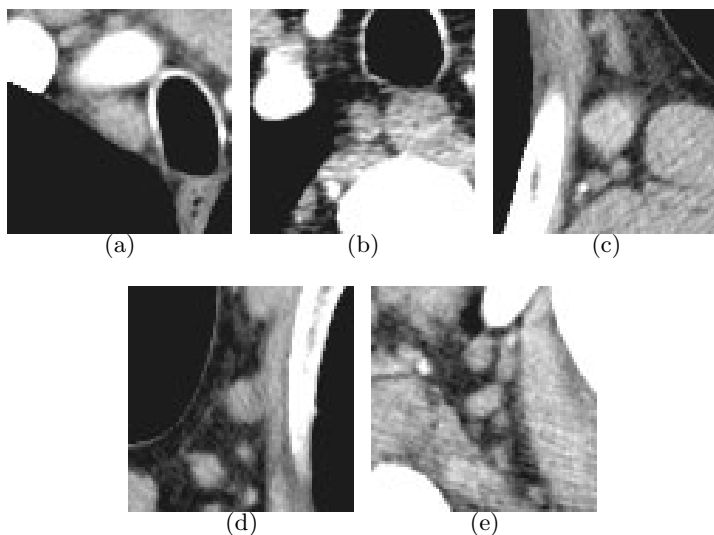


Fig. 1. Examples of cross-sectional CT images of lymph nodes

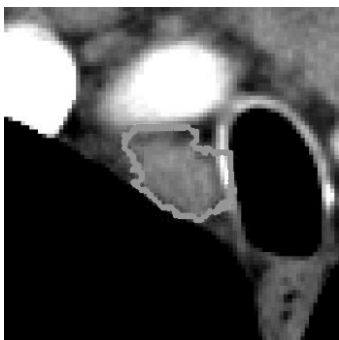


Fig. 2. Example of a failed lymph node segmentation. In this case, an edge-based segmentation approach was utilized. Edges of neighboring structures like contrast enhanced vessels or airways are stronger and cause segmentation errors.

manual segmentation is too time consuming, and currently available automated segmentation methods frequently fail to deliver usable results.

In this paper, we propose an optimal surface finding approach for lymph node segmentation in CT data sets. Our method avoids common problems of lymph node segmentation (Fig. 2) by utilizing a cost function that consists of a weighted edge and a region homogeneity term in combination with a surface smoothness constraint.

2 Related Work

Literature about the segmentation of lymph nodes is rare. Lu et al. propose to use a semi-automatic live-wire-based approach for the segmentation of central

chest lymph nodes [6]. A fast marching approach for semi-automatic segmentation of lymph nodes in 2D CT data has been reported in [10]. The authors propose to use their algorithm on a slice-by-slice basis for the 3D segmentation of lymph nodes in volumetric image data. No quantitative evaluation results were provided. Cordes et al. utilize a manual segmentation approach for neck lymph node segmentation to aid planning of neck dissections [2]. Dornheim et al. presented a 3D mass-spring model for the segmentation of neck lymph nodes in CT data [3]. The volumetric segmentation error ranged between 39% and 52% [3]. In [4], a 3-D minimum directional difference filter is used in combination with region growing and several false positive reduction strategies to extract abdominal lymph nodes in volumetric CT data. The authors report that the proposed method could detect 57.0% of the enlarged lymph nodes with approximately 58 false positives per case [4]. The approach presented in [7] utilizes a combination of a deformable surface and a statistical shape model for lymph node segmentation. In addition, a tool for manual intervention is provided to help the algorithm in converging to the desired object contours. In 6.9% out of 29 cases, the authors reported problems in getting a “usable segmentation” [7]. In conclusion, the robustness of lymph node segmentation methods needs to be further improved, and it is desirable to have a 3D segmentation approach that requires only little user interaction.

3 Methods

Our segmentation approach consists of three main processing steps. First, the user roughly identifies the approximate center point (voxel) \mathbf{c}_k of lymph node k to be segmented. Second, the local region around a lymph node is converted to a graph representation to transform the segmentation problem into a graph search (optimization) problem. For this purpose, we propose to utilize the optimal surface finding framework introduced in [5] and define a cost function that enables us to segment lymph nodes and to avoid common shortcomings of existing approaches (e.g., see Fig. 2). Third, the optimization problem is solved and the result is converted from a graph representation to a surface mesh and labeled volume. In the following sections, we describe our segmentation approach in detail.

3.1 Preprocessing

The density-values of lymph nodes are typically within a range between -100 and 150 Hounsfield units (HU). In a first preprocessing step, we limit the CT density values to this range by applying the following transfer function:

$$f_{iw}(x, y, z) = \begin{cases} 150 \text{ HU} & \text{if } 150 \text{ HU} < f(x, y, z) \\ f(x, y, z) & \text{if } -100 \text{ HU} \leq f(x, y, z) \leq 150 \text{ HU} \\ -100 \text{ HU} & \text{if } f(x, y, z) < -100 \text{ HU} \end{cases} ,$$

where $f(x, y, z)$ represents the density-value of a voxel at the coordinates x , y , and z . To reduce image noise, a $3 \times 3 \times 3$ median filter is applied to the transformed volume data set.

3.2 Graph Construction

A directed graph $G = (V, E)$, consisting of vertices V and edges E , representing a spherical volume of interest $\mathcal{VOT}(\mathbf{c}_k)$ around the approximate center point \mathbf{c}_k of lymph node k is generated (Fig. 3). This is accomplished by building a sphere-shaped triangular mesh (Fig. 3(a)) around \mathbf{c}_k with radius r . The radius r is a constant and chosen to be more than twice as large as the maximally expected radius of lymph nodes. Let n_v be the number of vertices of the spherical mesh. For each mesh vertex \mathbf{p}_i with $i \in n_v$, the volume f_{iw} is resampled along the line between center point \mathbf{c}_k and vertex \mathbf{p}_i in an equidistant fashion by using a linear interpolation function (Figs. 3(b) and 3(c)). Note that \mathbf{p}_i represents already a sample point, whereas \mathbf{c}_k is not used as sample point. The gray-value density samples on the line between \mathbf{c}_k and \mathbf{p}_i form the elements $g_i(j)$ of column i with $j \in [0, 1, \dots, n_e - 1]$. The number of elements per column is denoted as n_e , and $g_i(n_e - 1)$ represents the sample point at the location of vertex \mathbf{p}_i . The neighborhood relation between columns is defined by the mesh structure. If (p, q) is an edge of the triangular mesh, then column p and column q are adjacent, and directed edges are introduced to form a directed graph G . This includes the set up of a surface smoothness constraint Δ between any two adjacent columns to specify the maximum allowable change in columns for the surface [5]. Fig. 3(d) depicts a 2D example of the graph generation for $\Delta = 1$. By constructing the graph G as described above, we are able to utilize the surface detection algorithm for the segmentation of lymph nodes, which can have a spherical, elliptical, or slightly kidney-like shape.

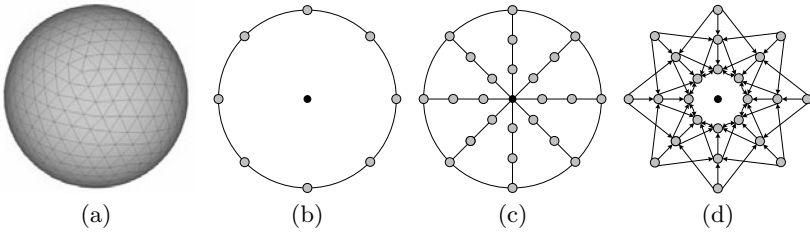


Fig. 3. Graph generation for the optimal surface finding step. (a) Spherical triangle mesh utilized for graph building. (b)-(d) 2D illustration of the graph generation process.

3.3 Cost Function

Designing an appropriate cost function is of paramount importance for any graph-based segmentation method. For our application, we define the cost function for element j of column i (node costs) as follows:

$$c_i(j) = [1 - w_{shape}(j)c_{edge_i}(j)] + \alpha c_{gh_i}(j), \quad (1)$$

where $c_{edge_i}(j)$ represents an edge and $c_{gh_i}(j)$ a gray-value homogeneity cost term. w_{shape} is a global shape weight derived from all the n_v columns of the graph structure, and α is used to adjust the influence of the homogeneity term relative to the edge term. In the following paragraphs, we will describe the components of our cost function in detail by using an example presented in Fig. 4

A cross-sectional CT image of a lymph node is shown in Fig. 4(a). The line in Fig. 4(a) represents a column of the graph. The corresponding gray-values are depicted in Fig. 4(b). The gray-value homogeneity term

$$c_{gh_i}(j) = \max_{a=0,1,\dots,j} \{g_i(a)\} - \min_{a=0,1,\dots,j} \{g_i(a)\} \tag{2}$$

measures the variation of gray-values along the path from column element $g_i(0)$ to $g_i(j)$ (Fig. 4(c)). The larger the value for $c_{gh_i}(j)$ the more unlikely it is that the element j of column i belongs to the lymph node.

The contrast around lymph nodes can vary considerably (Fig. 1). Consequently, an edge term that solely relies on edge magnitude is problematic as demonstrated by the example shown in Fig. 2. To avoid this problem, an edge cost function c_{edge_i} that identifies potential edge locations, but doesn't directly utilize edge magnitude information, is generated. First, the derivative of $c_{gh_i}(j)$ is calculated by using a central difference function:

$$c'_{gh_i}(j) = \frac{1}{3} \sum_{a=1}^3 [c_{gh_i}(j+a) - c_{gh_i}(j-a)] \tag{3}$$

Note that values for $c'_{gh_i}(j)$ are only generated for $j = 3, 4, \dots, n_e - 4$ to avoid dealing with undefined border values (Fig. 4(d)). Second, all local maxima of $c'_{gh_i}(j)$ are detected and the corresponding locations are stored in the set M_i . Third, the edge term is calculated by using

$$c_{edge_i}(j) = \max_{a \in M_i} \{p(j, a)\} \tag{4}$$

with $p(j, a) = e^{-\frac{(j-a)^2}{2\sigma^2}}$. The function $p(j, a)$ is used to model uncertainty regarding the exact edge location (Fig. 4(e)). The relative importance of possible edge locations is globally estimated by

$$w_{shape}(j) = \frac{\sum_{i=0}^{n_v-1} c_{edge_i}(j)}{\max_{a=0,1,\dots,n_e-1} \{\sum_{i=0}^{n_v-1} c_{edge_i}(a)\}} \tag{5}$$

The corresponding example plot is shown in Fig. 4(f). The idea behind this approach is as follows. Since the user specifies the approximate center \mathbf{c}_k of a lymph node, its edges approximately appear in concentric patterns around \mathbf{c}_k . In contrast, other nearby structures (e.g., vessels) within radius r do not lead to such a consistent pattern. Therefore, it is very likely that the weight elements of $w_{shape}(j)$ have larger values in proximity of the real lymph node edge, and we can utilize $w_{shape}(j)$ to weight the edge cost function term $c_{edge_i}(j)$. This allows us to avoid problems as shown in Fig. 2

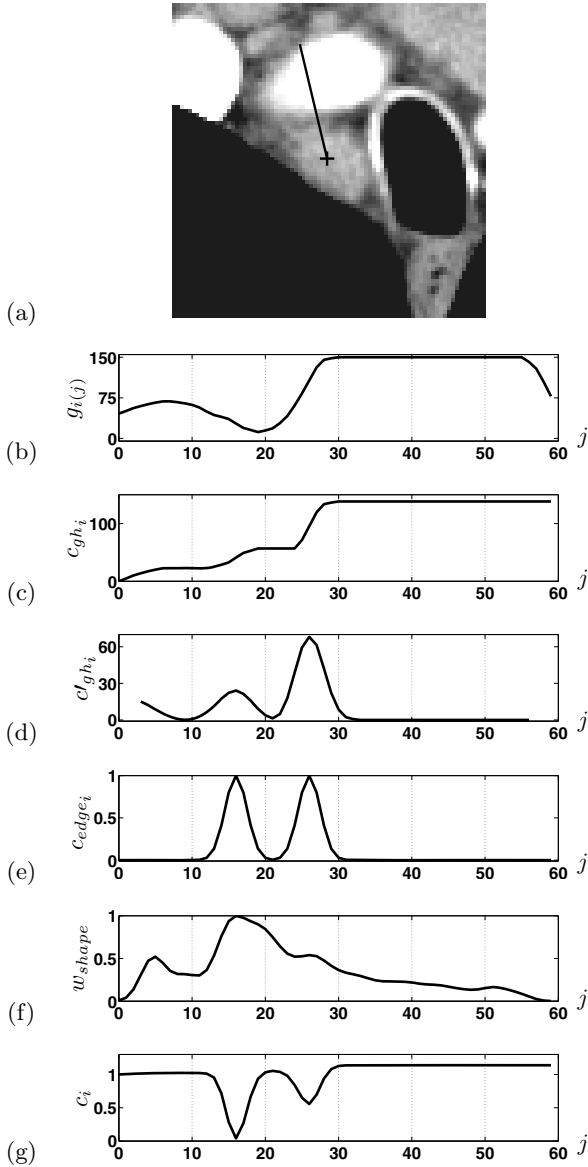


Fig. 4. Cost function calculation. (a) Cross-sectional CT image showing a lymph node. The line indicates a column of the graph structure G . (b) To the column corresponding gray-value profile. (c) Gray-value homogeneity function and (d) its derivative. (e) Edge cost function. (f) Global shape weight. (g) Final cost function for the column shown in (b).

In all our experiments, we used the following parameters for the cost function: $\alpha = 0.001$ and $\sigma = 1.5$. For graph generation, the column length was set to $r = 20$ mm, and the smoothness constraint $\Delta = 4$ was utilized. The spherical mesh consisted of $n_v = 642$ vertices. The number of elements per column was $n_e = 60$.

3.4 Lymph Node Segmentation

Once the graph G is generated and all costs are calculated, a maximum flow algorithm is used to solve the graph optimization problem [5], which runs in low degree polynomial time. The utilized surface finding framework guarantees to produce a globally optimal surface captured by our graph G according to the utilized cost function $c_i(j)$. For the representation of the segmentation result, the initial spherical triangle mesh is utilized. By adjusting the radial position of the vertices to the by means of optimization found surface position of the same column, a mesh of the segmentation result can be generated, since no topology changes in the mesh structure are required. In addition, the mesh-based representation of the segmentation result is converted into a volume-based representation by using a voxelization method [8].

4 Evaluation

4.1 Image Data and Independent Reference

For our experiments, we utilized 22 enlarged lymph nodes of the mediastinum, abdomen, head/neck, and axillary regions from several CT scans. The intra-slice resolution of the CT images ranged from 0.6 to 0.7 mm, and the inter-slice resolution was between 0.5 and 3 mm. Some of the CT data sets showed contrast enhanced blood vessels. To generate an independent reference for segmentation performance calculation, all lymph nodes were segmented by a medical expert. The reference segmentation was done in a slice-by-slice fashion by using a semiautomatic live wire [1] segmentation tool. This process took approximately 10 minutes per lymph node.

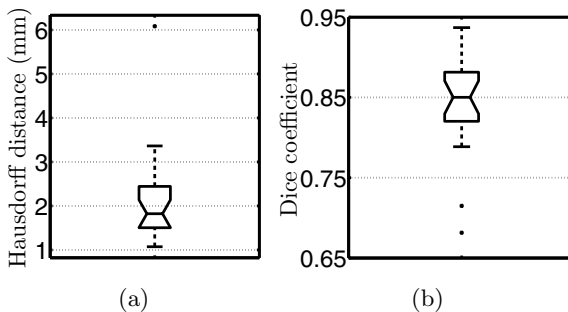


Fig. 5. Segmentation results on 22 test cases. (a) Box-plot of Hausdorff distance and (b) Dice coefficient.

4.2 Results

On average, 5.3 seconds were required on a workstation with a 2.40 GHz CPU to perform the computation of a lymph node segmentation. The manual specification of the approximate centers of lymph nodes to be segmented can be typically done in less than 20 seconds per node. For the mean signed and unsigned border positioning error, the average and standard deviation was 0.09 ± 0.17 mm and 0.47 ± 0.08 mm, respectively. The average Hausdorff distance was 2.12 mm, and

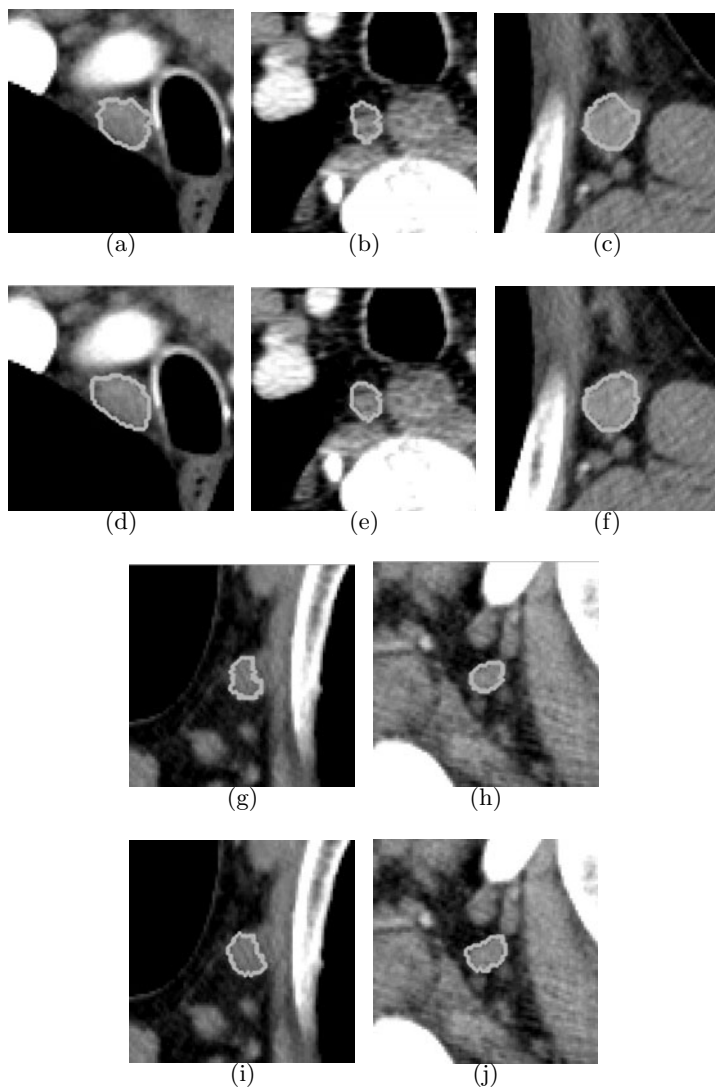


Fig. 6. Examples of segmentation results of our approach (a, b, c, g, and h) and corresponding independent reference segmentations (d, e, f, i, and j)

the corresponding box-plot is depicted in Fig. 5(a). The median is well below 2 mm, and in one case the Hausdorff distance was 6.08 mm due to a local segmentation inaccuracy. Over all 22 test cases, an average Dice coefficient of 0.84 was calculated. A box-plot of the Dice coefficient is shown in Fig. 5(b). A comparison between segmentations generated with our approach and the independent reference for five test cases is shown in Fig. 6.

5 Discussion

On 22 test cases, our method showed a low bias (border positioning error). The reported absolute border positioning error is on average smaller than the size of a voxel in x -, y -, or z -direction. Fig. 6 shows mostly local differences between our segmentation results and the independent reference, which is consistent with the distance-based error measures. The utilized volumetric error index is quite sensitive to small border positioning errors, because of the small size of some lymph nodes in relation to the voxel size of the CT scans. The required segmentation time is low compared to manual or semiautomatic slice-by-slice segmentation. To identify the lymph node to be segmented, only the approximate center of the node needs to be specified by the user. Based on this input, the segmentation is calculated automatically. Thus, if an automated lymph node detection method is available, the segmentation process can be completely automated.

6 Conclusion

We have presented a new approach for the segmentation of lymph nodes in volumetric CT data sets. Our method transforms the segmentation task into a graph optimization problem. By developing a cost function that consists of a weighted edge and a region homogeneity term, we were able to avoid common problems in lymph node segmentation. Based on the resulting lymph node segmentation, several quantitative indices of lymph nodes size (e.g., short and long axis length, volume, etc.) can be derived in an automated fashion. We expect that this will help physicians to better assess the status of lymph nodes and determine response to therapy in longitudinal studies.

Acknowledgments

This work was supported in part by the National Institutes of Health under Grant R01 EB004640 and Grant U01 CA140206.

References

1. Barrett, W., Mortensen, E.: Interactive live-wire boundary extraction. *Medical Image Analysis* 1(4), 331–341 (1997)
2. Cordes, J., Dornheim, J., Preim, B., Hertel, I., Strauss, G.: Pre-operative segmentation of neck CT datasets for the planning of neck dissections. In: *SPIE (Medical Imaging)*. 2, vol. 6144, pp. 6144–6163 (2006)

3. Dornheim, J., Seim, H., Preim, B., Hertel, I., Strauss, G.: Segmentation of neck lymph nodes in CT datasets with stable 3D mass-spring models segmentation of neck lymph nodes. *Acad. Radiol.* 14(11), 1389–1399 (2007)
4. Kitasaka, T., Tsujimura, Y., Nakamura, Y., Mori, K., Suenaga, Y., Ito, M., Nawano, S.: Automated extraction of lymph nodes from 3-D abdominal CT images using 3-D minimum directional difference filter. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part II. LNCS*, vol. 4792, pp. 336–343. Springer, Heidelberg (2007)
5. Li, K., Wu, X., Chen, D.Z., Sonka, M.: Optimal surface segmentation in volumetric images — A graph-theoretic approach 28, 119–134 (2006)
6. Lu, K., Merritt, S.A., Higgins, W.E.: Extraction and visualization of the central chest lymph-node stations. In: *Proc. SPIE (Medical Imaging): Computer-Aided Diagnosis*, vol. 6915, pp. 69151B-1–69151B-15 (2008)
7. Maleike, D., Fabel, M., Tetzlaff, R., von Tengg-Kobligk, H., Heimann, T., Meinzer, H., Wolf, I.: Lymph node segmentation on CT images by a shape model guided deformable surface method. In: *Proc. SPIE (Medical Imaging): Image Processing*, vol. 6914 (2008)
8. Reitinger, B., Bornik, A., Beichel, R.: Efficient volume measurement using voxelization. In: Joy, K., Szirmay-Kalos, L. (eds.) *Proc. of the Spring Conference on Computer Graphics 2003*, pp. 57–64. Comenius University, Bratislava (April 2003)
9. Schwartz, L., Bogaerts, J., Ford, R., Shankar, L., Therasse, P., Gwyther, S., Eisenhauer, E.: Evaluation of lymph nodes with RECIST 1.1. *European Journal of Cancer* 45(2), 261–267 (2009)
10. Yan, J., Zhuang, T., Zhao, B., Schwartz, L.H.: Lymph node segmentation from CT images using fast marching method. *Comput. Med. Imaging Graph* 28, 33–38 (2004)

Electron Microscopy Image Segmentation with Graph Cuts Utilizing Estimated Symmetric Three-Dimensional Shape Prior

Huei-Fang Yang* and Yoonsuck Choe

Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843-3112
hfyang@cse.tamu.edu, choe@tamu.edu

Abstract. Understanding neural connectivity and structures in the brain requires detailed three-dimensional (3D) anatomical models, and such an understanding is essential to the study of the nervous system. However, the reconstruction of 3D models from a large set of dense nanoscale microscopy images is very challenging, due to the imperfections in staining and noise in the imaging process. To overcome this challenge, we present a 3D segmentation approach that allows segmenting densely packed neuronal structures. The proposed algorithm consists of two main parts. First, different from other methods which derive the shape prior in an offline phase, the shape prior of the objects is estimated directly by extracting medial surfaces from the data set. Second, the 3D image segmentation problem is posed as Maximum A Posteriori (MAP) estimation of Markov Random Field (MRF). First, the MAP-MRF formulation minimizes the Gibbs energy function, and then we use graph cuts to obtain the optimal solution to the energy function. The energy function consists of the estimated shape prior, the flux of the image gradients, and the gray-scale intensity. Experiments were conducted on synthetic data and nanoscale image sequences from the Serial Block Face Scanning Electron Microscopy (SBFSEM). The results show that the proposed approach provides a promising solution to EM reconstruction. We expect the reconstructed geometries to help us better analyze and understand the structure of various kinds of neurons.

1 Introduction

Understanding neural connectivity and functional structure of the brain requires detailed 3D anatomical reconstructions of neuronal models. Recent advances in high-resolution three-dimensional (3D) image acquisition instruments [12], Serial Block-Face Scanning Electron Microscopy (SBFSEM) [3] for example, provide sufficient resolution to identify synaptic connections and make possible the

* This work was supported in part by NIH/NINDS grant #1R01-NS54252. We would like to thank Stephen J. Smith (Stanford) for the SBFSEM data.

reconstruction of detailed 3D brain morphological neural circuits. The SBFSEM utilizes backscattering contrast and cuts slices off the surface of the block by a diamond knife, generating images with a resolution in the order of tens of nanometers. The lateral (x - y) resolution can be as small as 10 – 20 nm/pixel, and the sectioning thickness (z -resolution) is around 30 nm. The high imaging resolution allows researchers to identify small organelles, even to trace axons and to identify synapses, thus enabling reconstruction of neural circuits. With the high image resolution, the SBFSEM data sets pose new challenges: (1) Cells in the SBFSEM image stack are densely packed, and the enormous number of cells make manual segmentation impractical, and (2) the inevitable staining noise, the incomplete boundaries, and inhomogeneous staining intensities increase the difficulty in the segmentation and the subsequent 3D reconstruction and visualization.

To reconstruct neural circuits from the SBFSEM image volumetric data, segmentation, that is partitioning an image into disjoint regions, is a fundamental step toward a fully neuronal morphological model analysis. Segmentation of the SBFSEM images amounts to delineating cell boundaries. Different approaches have been proposed in the literature for such a task. Considering segmentation as the problem of restoring noisy images, Jain *et al.* [4] proposed a supervised learning method which trained a convolutional network with 34,000 free parameters to classify each voxel as being inside or outside a cell. Another similar approach in the machine learning paradigm was proposed by Andres *et al.* [5]. These methods require data sets with ground truth, and creation of labeled data sets is labor-intensive. Semi-automatic tracking methods utilizing level-set formulation [6] or active contours [7] have also been investigated. Computation of level-set method is expensive, and the solution can sometimes get stuck in local minima.

In this paper, we propose a 3D segmentation framework with estimated 3D symmetric shape prior. First, different from other methods which derive the shape prior in an offline phase, the shape prior of the objects is estimated directly by extracting medial surfaces of the data set. Second, the 3D image segmentation problem is posed as Maximum A Posteriori (MAP) estimation of Markov Random Field (MRF). First, the MAP-MRF formulation minimizes the Gibbs energy function, and then we use graph cuts to obtain the optimal solution to the energy function. The energy function consists of the estimated shape prior, the flux of the image gradients, and the gray-scale intensity.

2 Graph Cut Segmentation

Image segmentation is considered as a labeling problem that involves assigning image pixels a set of labels [8]. Taking an image I with the set of pixels $\mathcal{P} = \{1, 2, \dots, M\}$ and the set of labels $\mathcal{L} = \{l_1, l_2, \dots, l_K\}$, the goal of image segmentation is to find an optimal mapping $X : \mathcal{P} \rightarrow \mathcal{L}$. According to the random field model, the set of pixels \mathcal{P} is associated with random field $X = \{X_p : p \in \mathcal{P}\}$, where each random variable X_p takes on a value from the set of labels \mathcal{L} . A possible labeling $x = \{X_1 = x_1, X_2 = x_2, \dots, X_M = x_M\}$, with $x_p \in \mathcal{L}$, is called a

configuration of X . Each configuration represents a segmentation. Finding the optimal labeling x^* is equivalent to finding the maximum a posteriori (MAP) estimate of the underlying field given the observed image data D :

$$x^* = \operatorname{argmax}_{x \in X} \Pr(x | D). \quad (1)$$

By Bayes' rule, the posterior is given by:

$$\Pr(x | D) \propto \Pr(D | x) \Pr(x), \quad (2)$$

where $\Pr(D | x)$ is the likelihood of D on x , and $\Pr(x)$ is the prior probability of a particular labeling x , being modeled as a Markov random field (MRF) which incorporates contextual constraints based on piecewise constancy [8]. An MRF satisfies the following two properties with respect to the neighborhood system $\mathcal{N} = \{N_p | p \in \mathcal{P}\}$:

$$\textit{Positivity} : \quad \Pr(x) > 0, \quad \forall x \in \mathcal{X}, \quad (3)$$

$$\textit{Markovianity} : \Pr(x_p | x_{\mathcal{P}-\{p\}}) = \Pr(x_p | x_{N_p}), \quad \forall p \in \mathcal{P}. \quad (4)$$

Furthermore, according to Hammersley-Clifford theorem [9], a random field with Markov property obeys a Gibbs distribution, which takes the following form:

$$\Pr(x) = \frac{1}{Z} \exp(-E(x)), \quad (5)$$

where Z is a normalizing constant called the partition function, and $E(x)$ is the Gibbs energy function, which is:

$$E(x) = \sum_{c \in \mathcal{C}} V_c(x_c), \quad (6)$$

where \mathcal{C} is the set of cliques, and $V_c(x_c)$ is a clique potential. Taking a log likelihood of Equation 2, the MAP estimate of $\Pr(x | D)$ is equivalent to minimizing the energy function:

$$-\log \Pr(x | D) = E(x | D) = \sum_{p \in \mathcal{P}} V_p(x_p | D) + \sum_{p \in \mathcal{P}} \sum_{q \in N_p} V_{pq}(x_p, x_q | D), \quad (7)$$

where $V_p(x_p | D)$ and $V_{pq}(x_p, x_q | D)$ are the unary and piecewise clique potentials, respectively.

Minimizing the energy function $E(x | D)$ is NP-hard, and the approximate solution can be obtained by graph-cuts using α -expansion algorithm [10]. The graph cuts represent an image as a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with a set of vertices (nodes) \mathcal{V} representing pixels or image regions and a set of edges \mathcal{E} connecting the nodes. Each edge is associated with a nonnegative weight. The set \mathcal{V} includes the nodes of the set \mathcal{P} and two additional nodes, the source s and the sink t . All nodes $p \in \mathcal{V}$ are linked to the terminals s and t with weight w_{sp} and w_{pt} ,

respectively. Edges between the nodes and the terminals are called t-links, and edges between node p and its neighborhood q with weight w_{pq} are called n-links. The t-links and n-links model the unary and piecewise clique potentials, respectively. A cut \mathcal{C} is a subset of edges \mathcal{E} that separates terminals in the induced graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$ and thus partitions the nodes into two disjoint subsets while removing edges in the cut \mathcal{C} . The partitioning of a graph by a cut corresponds to a segmentation in an image. The cost of a cut, denoted as $|\mathcal{C}|$, is the sum of the edge weights in \mathcal{C} . Image segmentation problem then turns into finding a minimum cost cut that best partitions the graph, which can be achieved by the min-cut/max-flow algorithm [11]. One criterion of minimizing the energy function by graph cuts is that $V_{pq}(x_p, x_q)$ is submodular, that is, $V_{pq}(0, 0) + V_{pq}(1, 1) \leq V_{pq}(1, 0) + V_{pq}(0, 1)$ [10].

3 Symmetric Shape Prior Estimation

Anatomical structures, such as axons, dendrites, and soma, exhibit locally symmetric shapes to the medial axis which is also referred to as the skeleton and is commonly used for shape representation. The medial axis of a 3D object is generally referred to as the medial surface. Extracting medial surface approaches include distance field based methods [12], topological thinning, gradient vector flow methods [13] and others [14]. We follow the method proposed by Bouix *et al.* [12] to extract the medial surface. Gray-scale images are first converted to binary images, and a Euclidean distance function to the nearest boundary is computed at each voxel, as shown in Figure 1(b). Guiding the thinning procedure by exploiting properties of the average outward flux of the gradient vector field of a distance transform, the resulting medial surface for a particular object is shown in Figure 1(c). Finally, the estimated shape is obtained by first expanding each point in the extracted medial surface with the shortest distance to the

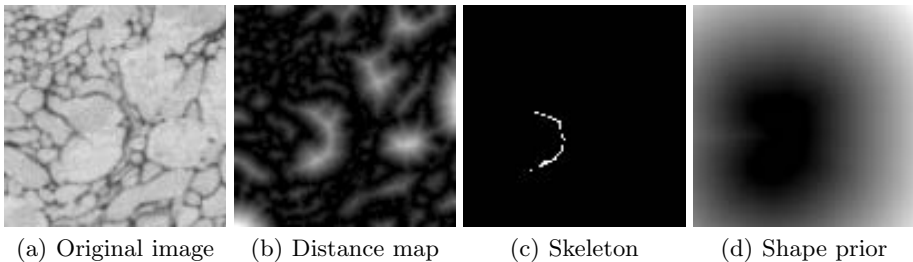


Fig. 1. Method of shape prior estimation. (a) is an image drawn from the input stack. (b) is the distance map computed from the binary image stack. (c) shows the extracted skeleton (white curves) from the distance map. (d) shows the estimated shape prior. Dark is the expanded region, and bright indicates the points outside of the expanded region which are represented by a distance function.

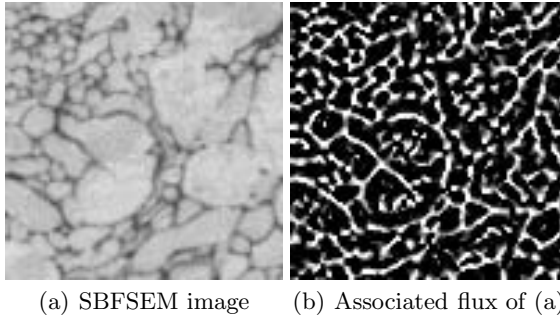


Fig. 2. Flux of the gradient vector fields of a slice from the image stack. (a) shows part of an original gray-scale intensity image from the SBFSEM stack. (b) is the associated flux of image gradients of (a), where the foreground objects have negative flux (dark), and the background objects have positive flux (bright).

boundary. The points outside the expanded region are represented by a distance function:

$$D(p) = \|p - s_p\|, \quad (8)$$

where $\|p - s_p\|$ represents the Euclidean distance from p to the nearest pixel s_p in the expanded region. Shown in Figure 1(d) is the estimated shape prior for the object in Figure 1(c). The estimated shape prior is then incorporated in the unary term (Equation 11) acting as a constraint in the minimization process.

4 Definition of Unary Potential

A unary term defines the cost that a node is assigned a label x_p , that is, the corresponding cost of assigning node p to the foreground or the background. In our segmentation framework, a unary term consists of two parts: the flux of the gradient vector field and the shape prior estimated in section 3.

4.1 Flux

Flux has recently been introduced by Vasilevskiy and Siddiqi [15] into image analysis and computer vision. They incorporated flux into level-set method to segment blood vessel images. After that, flux has also been integrated into graph cuts [16, 17] to improve the segmentation accuracy. The introduction of flux into graph cuts can reduce the discretization artifacts which is a major shortcoming in graph cuts [16]. By definition, considering a vector field v defined for each point in \mathcal{R}^3 , the total inward flux of the vector field through a given continuous hypersurface S is given by the surface integral [15]:

$$F(S) = \int_S \langle N, v \rangle dS, \quad (9)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean dot product, N are unit inward normals to surface element dS consistent with a given orientation. In the implementation of Equation 9, the calculation of the flux is simplified by utilizing the divergence theorem which states that the integral of the divergence of a vector field v inside a region equals to the outward flux through a bounding surface. The divergence theorem is given by:

$$\int_R \operatorname{div} v \, dR = \oint_S \langle N, v \rangle \, dS, \quad (10)$$

where R is the region. For the numerical implementations, we consider the flux through a sphere in the case of 3D, and v is defined as the normalized (unit) image gradient vector field of the smoothed volume I_σ , $\frac{\nabla I_\sigma}{\|\nabla I_\sigma\|}$. Figure 2(b) shows the flux of gradient vector fields of Figure 2(a). Note that the flux is computed in 3D but only 2D case is shown here. The foreground object has negative flux (dark) whereas the background has positive flux (bright).

4.2 Incorporating Flux and Shape Prior

Combining the flux of gradient vector fields and the estimated shape prior yields a new unary term. Inspired by [17], we assigned the edge weights between node p and terminals s and t as:

$$\begin{aligned} w_{sp} &= -\min(0, F(p)), \\ w_{pt} &= \max(0, F(p)) + \alpha D(p), \end{aligned} \quad (11)$$

where $F(p)$ denotes the flux at point p , and α is a positive parameter adjusting the relative importance of the shape prior $D(p)$. In our experiments, the value of α was set to 0.2.

5 Definition of Piecewise Potential

In the SBFSEM images, the foreground and background can be discriminated by their gray-scale intensities. Compared to the background, the foreground objects usually have higher intensity values. Boundaries can thus be determined if the intensity differences between points are large. To capture the boundary discontinuity between pixels, the weight between node p and its neighbor q is defined as [18]:

$$w_{pq} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{\|p - q\|}, \quad (12)$$

where I_p and I_q are point intensities ranging from 0 to 255, $\|p - q\|$ is the Euclidean distance between p and q , and σ is a positive parameter set to 30. Equation 12 penalizes a lot for edges with similar gray-scale intensities while it penalizes less for those with larger gray-scale differences. In other words, a cut is more likely to occur at the boundary, where the edge weights are small. For 3D images, the 6-, 18-, or 26-neighborhood system is commonly used. Here, the 6-neighborhood system was used in our experiments.

6 Experimental Results

Experiments were conducted on synthetic data sets and an SBFSEM image stack in order to evaluate the performance of the proposed approach.

6.1 Synthetic Data Sets

The synthetic data sets consisted of two image stacks, each of which having the size of $100 \times 100 \times 100$. Gaussian noise was added to each image slice to simulate noise during image acquisition process. Here, three different levels of Gaussian noise with standard deviation $\sigma = 0.0447, 0.0632$, and 0.0775 were added to the two synthetic data sets, thus resulting in a total of 6 image stacks. Shown in Figure 3(a) is a noisy image slice from the synthetic image stack in Figure 3(b). The reconstruction results of the two synthetic image stacks are shown in Figure 4(b) and Figure 4(f), respectively, and their ground truth is given in Figure 4(a) and Figure 4(e) accordingly. As can be seen from the close-up comparisons of the reconstruction results and the ground truth, the reconstruction results are almost identical to the ground truth with minor differences.

To quantitatively measure the performance of the proposed segmentation method, we used the F-measure, $F = \frac{2PR}{P+R}$, where P and R are the precision and recall of the segmentation results relative to the ground truth. More specifically, let Z be the set of voxels of the obtained segmentation results and G be the ground truth, then $P = \frac{|Z \cap G|}{|Z|}$ and $R = \frac{|Z \cap G|}{|G|}$, where $|\cdot|$ is the number of voxels. The average precision and recall values of the reconstruction results of the 6 synthetic image stacks were 0.9659 and 0.9853, respectively, yielding an average of F-measure being 0.9755. We also applied Dice coefficient (DC) [19] to measure the similarity between two segmentations. DC measures the overlapped

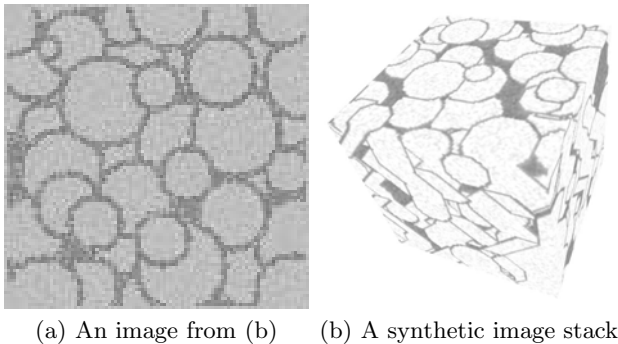


Fig. 3. A synthetic image and a synthetic data set. (a) is a noisy image with 100×100 pixels selected from the synthetic image stack in (b). (b) shows one of the two synthetic image stacks, which contains 100 images.

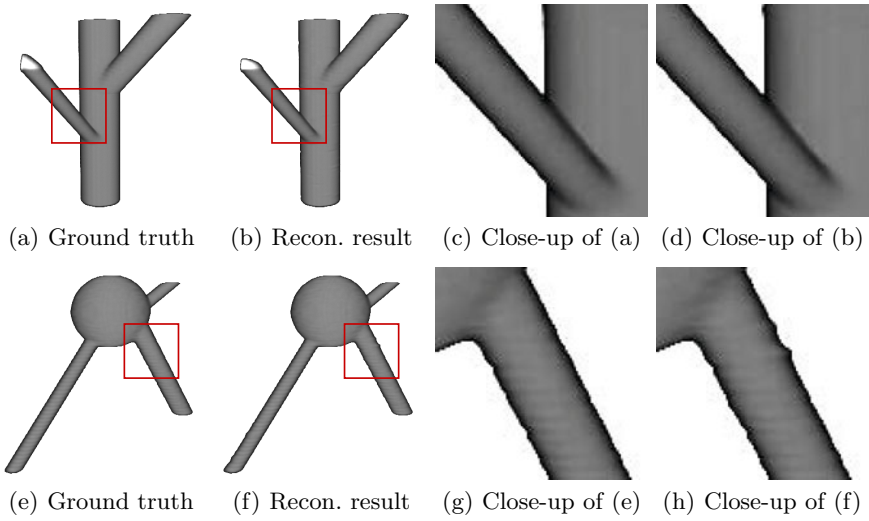


Fig. 4. Ground truth and reconstruction results of the synthetic data sets. (a) and (e) are the ground truth of the two synthetic data sets. (b) and (f) are the reconstruction results from the image stacks in which Gaussian noise with $\sigma = 0.04477$ was added. As can be seen from the close-up comparisons of the ground truth and the reconstruction results, the reconstruction results are almost identical to the ground truth with minor differences.

regions between the obtained segmentation results and the ground truth, defined as $DC = \frac{2|Z \cap G|}{|Z| + |G|}$, where 0 indicates no overlap between two segmentations, and 1 means two segmentations are identical. The average DC value on the synthetic data sets was 0.9748, implicating the reconstruction results are almost identical to the ground truth.

The mean computation time using a Matlab implementation of the proposed approach for processing a synthetic image stack ($100 \times 100 \times 100$) on a standard PC with Core 2 Duo CPU 2.2 GHz and 2 GB memory was 20 seconds.

6.2 SBFSEM Image Stack

Experiments on the SBFSEM data were conducted on one image stack ($631 \times 539 \times 561$), on different parts (sub-volumes) of it. Figure 5(a) shows an EM image with 631×539 pixels from the larval zebrafish tectum volumetric data set, shown in Figure 5(b). The reconstruction results of the proposed method are shown in Figure 6, where Figure 6(a) and Figure 6(b) show parts of neurons, and Figure 6(c) shows the elongated structures.

For the validation of the proposed algorithm, we manually segmented a few neurons using TrakEM2 [20], serving as the ground truth. Again, F-measure and DC were used as the evaluation metrics. The average precision and recall values of the reconstruction results shown in Figure 6 were 0.9660 and 0.8424,

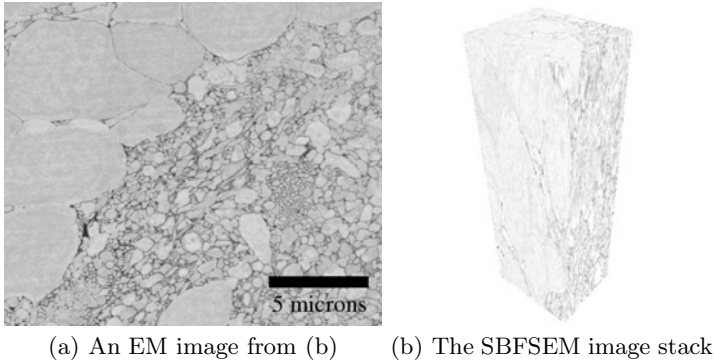


Fig. 5. An image and a volumetric SBFSEM data set. (a) shows an EM image with 631×539 pixels from the SBFSEM stack in (b). Note that cells in the SBFSEM images are densely packed. (b) is the SBFSEM image stack of larval zebrafish optic tectum, containing 561 images.

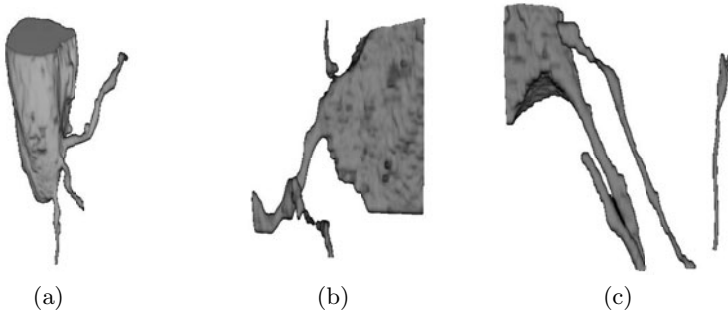


Fig. 6. Reconstruction results of the proposed method. (a) and (b) show parts of neurons, and (c) shows the elongated axon structures.

respectively, and thus the average of F-measure was 0.9. The average DC value of the reconstruction results was 0.8918, showing that the proposed method can reconstruct the neuronal structures from the SBFSEM images.

7 Conclusion and Future Work

We presented a 3D segmentation method with estimated shape prior for the SBFSEM reconstruction. The shape prior was estimated directly from the data set based on the local symmetry property of anatomical structures. With the help of the shape prior along with the flux of image gradients and image gray-scale intensity, the proposed segmentation approach can reconstruct neuronal structures from densely packed EM images. Future work includes applying the method to larger 3D volumes and seeking a systematically quantitative and qualitative validation method for the SBFSEM data set.

References

1. Helmstaedter, M., Briggman, K.L., Denk, W.: 3D structural imaging of the brain with photons and electrons. *Current Opinion in Neurobiology* 18, 633–641 (2008)
2. Briggman, K.L., Denk, W.: Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology* 16, 562–570 (2006)
3. Denk, W., Horstmann, H.: Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology* 2, e329 (2004)
4. Jain, V., Murray, J.F., Roth, F., Turaga, S., Zhigulin, V.P., Briggman, K.L., Helmstaedter, M., Denk, W., Seung, H.S.: Supervised learning of image restoration with convolutional networks. In: *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 1–8 (2007)
5. Andres, B., Köthe, U., Helmstaedter, M., Denk, W., Hamprecht, F.A.: Segmentation of sbfsem volume data of neural tissue by hierarchical classification. In: Rigoll, G. (ed.) *DAGM 2008. LNCS*, vol. 5096, pp. 142–152. Springer, Heidelberg (2008)
6. Macke, J.H., Maack, N., Gupta, R., Denk, W., Schölkopf, B., Borst, A.: Contour-propagation algorithms for semi-automated reconstruction of neural processes. *J. Neuroscience Methods* 167, 349–357 (2008)
7. Jurrus, E., Hardy, M., Tasdizen, T., Fletcher, P., Koshevoy, P., Chien, C.B., Denk, W., Whitaker, R.: Axon tracking in serial block-face scanning electron microscopy. *Medical Image Analysis* 13, 180–188 (2009)
8. Li, S.Z.: *Markov random field modeling in image analysis*. Springer, New York (2001)
9. Hammersley, J.M., Clifford, P.: *Markov field on finite graphs and lattices* (1971)
10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 147–159 (2004)
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 1222–1239 (2001)
12. Bouix, S., Siddiqi, K., Tannenbaum, A.: Flux driven automatic centerline extraction. *Medical Image Analysis* 9, 209–221 (2005)
13. Hassouna, M.S., Farag, A.A.: Variational curve skeletons using gradient vector flow. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 2257–2274 (2009)
14. Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using the poisson equation. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 1991–2005 (2006)
15. Vasilevskiy, A., Siddiqi, K.: Flux maximizing geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 1565–1578 (2002)
16. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: *Proc. IEEE Int'l Conf. Computer Vision*, pp. 564–571 (2005)
17. Vu, N., Manjunath, B.S.: Graph cut segmentation of neuronal structures from transmission electron micrographs. In: *Proc. Int'l Conf. Image Processing*, pp. 725–728 (2008)
18. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. *Int'l J. Computer Vision* 70, 109–131 (2006)
19. Dice, L.R.: Measures of the amount of ecologic association between species. *Ecology* 26, 297–302 (1945)
20. Cardona, A., Saalfeld, S., Tomancak, P., Hartenstein, V.: TrakEM2: open source software for neuronal reconstruction from large serial section microscopy data. In: *Proc. High Resolution Circuits Reconstruction*, pp. 20–22 (2009)

Retinal Vessel Extraction with the Image Ray Transform

Alastair H. Cummings and Mark S. Nixon

University of Southampton, Southampton, SO17 1BJ, UK
{ahc08r,msn}@ecs.soton.ac.uk

Abstract. Extraction of blood vessels within the retina is an important task that can help in detecting a number of diseases, including diabetic retinopathy. Current techniques achieve good, but not perfect performance and this suggests that improved preprocessing may be needed. The image ray transform is a method to highlight tubular features (such as blood vessels) based upon an analogy to light rays. The transform has been employed to enhance retinal images from the DRIVE database, and a simple classification technique has been used to show the potential of the transform as a preprocessor for other supervised learning techniques. Results also suggest potential for using the ray transform to detect other features in the fundus images, such as the fovea and optic disc.

1 Introduction

Automated detection of blood vessels within eye fundus images is an important step in aiding diagnosis of a number of diseases, diabetic retinopathy in particular. The possible application of vision techniques to this problem was first identified by Chaudhuri et al. [1] through use of matched filters. A morphological approach was taken by Zana and Klein [2] whilst Jiang and Mojon [3] investigated different approaches to thresholding. Staal et al. [4] both assembled the DRIVE database upon which our work is tested and described a supervised learning method of classification of pixels based upon ridges within the image. Soares et al. [5] developed a supervised technique based upon Morlet wavelets and Niemeijer et al. [6] compared previous unsupervised techniques with a simple classifier on the DRIVE database. More recent work has built upon matched filters through optimisation of parameters [7] and use of extra methods to decide on thresholds [8]. The best supervised techniques are capable of achieving an accuracy on the DRIVE database of around 0.95, and we propose that superior preprocessing methods are necessary to improve performance further. We describe experiments with a preprocessor based upon a physical analogy to light rays; the image ray transform.

Physical analogies are an exciting paradigm in computer vision enabling the creation of novel techniques that approach the problems of feature extraction from entirely different angles [9]. These analogy based techniques have the advantage of being based on physical properties of natural phenomena such as water, heat or force fields and so are more easily understood by those using

them. In addition to the intuitive nature of the algorithms, the parameters used have meanings that are clear and have real world analogues. Although analogy operators are heavily based upon a well defined physical concepts, the analogies can be adapted outside this definition to increase their effectiveness and flexibility whilst maintaining the strengths provided by the analogy. These properties are a clear advantage over many standard techniques for which the mechanics can be hard to grasp and parameter selection is not clear.

Heat flow has been used as an analogy due to its smoothing properties. Anisotropic diffusion [10] is an edge-aware smoothing technique that allows heat to flow across areas of low but not high edge strength allowing Gaussian noise to be eliminated whilst maintaining important edge features. This is one of the earliest examples of a principled vision technique based upon an analogy. Water flow has also been used [11] as an analogy for image segmentation, and has been tested for medical applications including retinal images.

Other techniques have not used light in such a strongly analogical sense; The Eikonal equation describes the time a ray takes to travel from the boundary of an anisotropic medium to a point within it and has been used in a number of applications [12]. However none of these fully take advantage of the possible analogical formulation and is most often used as a distance metric. We use the image ray transform, a transform based upon an analogy to light rays, to detect retinal vasculature. Previously it has been used to enhance circle detection [13], but here we take advantage of its ability to highlight tubular features as shown by its previous use enabling enrolment for ear biometrics [14].

This paper describes a retinal blood vessel extraction technique using the image ray transform which detailed in section 2. Section 3 describes the application of the transform for retinal blood vessel extraction, showing how it can highlight vascular features well. Finally section 4 draws conclusions and describes future work; expanding on vessel extraction and expansion into other related areas.

2 The Image Ray Transform

The image ray transform is a novel technique for extracting tubular and circular features that are not often found by other methods. It uses a pixel based ray tracing technique and a subset of the laws of optics to trace rays through an image. These then react to certain structural features, emphasising them. Whilst the transform is based on the principles of optics, the details of the technique can be adjusted to suit successful feature extraction rather than accurate simulation of the natural phenomenon. The implementation capitalises only on the basis of the analogy; we do not simulate intricate details of light propagation.

2.1 Laws of Optics

Rays are a method of modelling the propagation of waves, most often light. Specific regard is given for the direction of the wave as it interacts with its environment, and wave-like interactions such as diffraction are ignored. The path

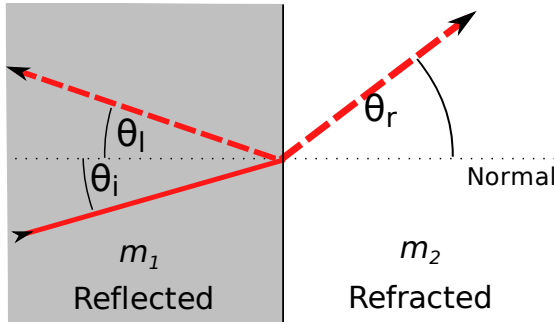


Fig. 1. Refraction and reflection of light at the boundary of two media

of a light ray will be altered when it crosses the boundary with a medium of different refractive index, refracting and/or reflecting (see figure 1). Light crosses the boundary between the media at an angle of θ_i to the normal of the boundary (the dotted line). If reflection occurs, then $\theta_l = \theta_i$. If refraction occurs, the light refracts at an angle of θ_r to the normal where θ_r is calculated from the refractive indices of n_1 and n_2 of the media m_1 and m_2 . Refractive index n is the ratio of the speed of light in a vacuum to the speed of light within the medium, in nature $1 \leq n \lesssim 4$. θ_r is found by Snell's Law:

$$\frac{\sin \theta_i}{\sin \theta_r} = \frac{n_2}{n_1} \quad (1)$$

If $n_1 < n_2$, light bends towards the normal, so $\theta_r < \theta_i$. If $n_1 > n_2$, light bends away from the normal, as long as it is below the critical angle θ_c . This is the angle for which θ_r would be equal to 90° and is calculated as:

$$\theta_c = \sin^{-1} \left(\frac{n_2}{n_1} \right) \quad (2)$$

Reflected angles above this are physically impossible. In this case, the light is totally internally reflected. In the natural world, the amount of refraction and reflection that occurs depends on the media and in most cases is a combination of the two, some part passing through, and some being reflected back. These rules form the basis of the image ray transform.

2.2 The Image Ray Transform

The image ray transform takes the laws of optics described previously and applies them to the problem of feature extraction. The transform operates by tracing a number of rays through an image. The paths of these rays are then used generate a new image, with tubular and circular features emphasised. The image is analogised to a matrix of two dimensional glass blocks, each representing a pixel whose refractive index is related to the intensity of the pixel in the image.

One method to assign a refractive index to a pixel with intensity i is shown in equation 3 (another is described in section 2.3), where n_{\max} defines the maximum possible refractive index. The indices are then spaced evenly between 1 and n_{\max} :

$$n_i = 1 + \left(\frac{i}{255}\right) \cdot (n_{\max} - 1) \tag{3}$$

The splitting of rays into reflected and refracted parts is not considered as in order to reduce computational complexity.

In this matrix of blocks, a ray is created with the position (x and y) and the direction (ϕ) drawn from a uniform distribution. For an image of size $w \times h$:

$$x \in [0, w), y \in [0, h), \phi \in [0, 2\pi) \tag{4}$$

ϕ is converted into a unit vector, \mathbf{V} , representing the direction of the ray as this is more convenient for calculation. The position vector \mathbf{p} initialises the ray at:

$$\mathbf{p}^{<0>} = (x, y)^T \tag{5}$$

and at iteration i :

$$\mathbf{p}^{<i+1>} = \mathbf{p}^{<i>} + \mathbf{V} \tag{6}$$

At each pixel through which the ray passes, \mathbf{A} is updated to show that the ray has passed through: $\mathbf{A}(\mathbf{p})$ is only increased once per ray, so as to prevent small loops in the ray's path repeatedly increasing a single pixel and excessively emphasising noise or other undesirable features. When crossing a pixel boundary between media of differing refractive indices, a new path must be calculated using a vector formation of the laws described in section 2.1. Using the direction of the ray \mathbf{V} , the normal of the boundary \mathbf{N} , and the refractive indices of the first and second media n_1 and n_2 the new direction can be calculated. If $n_1 > n_2$ we have to test whether we should internally reflect or refract. The critical angle θ_c must be found from equation 2 and if $n_2 > n_1$ or $\theta_i < \theta_c$ the ray refracts in direction \mathbf{R}_r , otherwise it reflects in direction \mathbf{R}_l . The equations for calculating these from \mathbf{V} and \mathbf{N} can be found in [15]. The ray is traced in this manner until it has either undergone d reflections or refractions or the ray exits the image. This is repeated for N rays or until a stopping condition (section 2.3) is reached, and the normalised accumulator gives the transformed image.

The values of the normals (\mathbf{N}) used to calculate the new directions of rays are always set to the normal of the edge direction found by the Sobel operator at that point. Currently, we only perform refraction when moving from higher to lower refractive indices as this tends to improve the quality of the transform result.

2.3 Refinements

In addition to the basic transform, a number of additional parameters and processes can be performed with the transform to give a wider range of results and to extend the range of circumstances in which the transform is useful.

Intensity and Refractive Index. The transform will only extract features that have larger refractive indices than the surrounding area because total internal reflection only prevents rays from passing from materials with higher to lower refractive indices, not vice versa. As the refractive index is proportional to intensity, this has the effect of extracting features that are lighter than their surrounding area. Depending on what is known about the properties of the features to be extracted different measures can be taken to improve the result of the transform.

If the approximate intensity of the desired features is known then the image can be easily transformed to make the target intensity have the largest value. This can be done by finding the difference from the target intensity t to the original intensity i_o for each pixel, as in equation 7:

$$i_t = |i_o - t| \quad (7)$$

It may also be the case that the difference in intensity between features and the surrounding area is not significant enough to extract it with the linear refractive indices calculated by equation 3. In such a case an alternative version can be used (equation 8) that assigns refractive indices exponentially, to ensure greater difference, and more refraction and reflection:

$$n_i = e^{\frac{i}{k}} \quad (8)$$

In this case it is k rather than n_{\max} that controls the scale of the refractive indices.

Automatic Stopping Condition. Finding the optimal number of rays that should be traced in order to ensure the result of the transform is of sufficient quality is a challenge that must be met. Rather than use fixed values of N , an improved method for automatically deciding when the transform should cease is to monitor the resultant image and stop when it no longer changes significantly between iterations.

The best method that has been found to do this is to measure the difference in the normalised accumulator image between iterations. We use the root mean squared (RMS) difference between the intensities:

$$D^{<t>}(\mathbf{I}^{<t>}, T) = \sqrt{\frac{1}{|\mathbf{I}'|} \sum_{i \in \mathbf{I}'} \left(\mathbf{I}^{<t>}(i) - \mathbf{I}^{<t-T>}(i) \right)^2} \quad (9)$$

where T is the number of iterations between each comparison. Rather than using successive images, the operation of a transform is sufficiently fast that large intervals should be used, that is, $T \gg 1$. The results produced by this method are consistent with the observed resultant images, with the size of D accurately reflecting the change that can be observed in the successive images.

The RMS difference measure is also cheap to calculate when T is set high enough, as expensive accumulator normalisations are calculated infrequently. It can be used as a stopping condition by setting a minimum value (D_S) to stop the process. From experimentation it has been found that a value for D_S of 1 when $T = 1000$ allows termination with a stable result.

3 Segmentation of Blood Vessels in Retinal Images

The ability of the image ray transform to highlight tubular curvilinear structures by turning them into optic fibres was shown in [14] and detection of blood vessels within eye fundus images is an appropriate application to further demonstrate this. As we do not use any advanced techniques for classifying the images that result from the transform, we do not expect accuracy superior to other techniques (although we try to establish that this is possible with superior classification techniques). We primarily intend to show that the transformed image is a suitable preprocessor for many retinal vessel extraction techniques, more appropriate than either intensity or edge detected images. Current techniques do not achieve accuracy of much more than 95%, and through use of an appropriate preprocessing technique such as the image ray transform we suggest that these results may be improved further.

3.1 Extraction Technique

We use the DRIVE database [4] to test our technique and compare to other techniques. The database consists of 20 test and 20 training fundus images, of which we only use the test images. The green channel of the retinal image is extracted for use as it provides the greatest contrast for blood vessels (figure 2(a)). The image is effectively inverted by the use of the target parameter (section 2.3), so that the lighter vascular structures are highlighted most strongly by the transform. Use of the transform also introduces additional problems. The edge of the fundus image acts as a circle and is highlighted by the transform; we expand the masks included in the database by 3 pixels in order to remove these unwanted features. The exponential transform also highlights the fovea strongly, which is undesired and so we used an automatic method (template matching with a Gaussian template) to find and remove them.

The image ray transform was applied to the green channel images (figure 2(b)) using both linear (referred to as RT-n) and exponential refractive indices (RT-k), as both gave results that differed significantly. Parameters used were $d = 256$, $t = 0$ and $D_s = 1$. For the linear refractive indices $n_{\max} = 40$ and for the exponential $k = 7$. We also tested on aggregated version of linear and exponential indices (RT-nk), so as to exploit the advantages of both techniques.

The results of the transform then had their histograms equalised (figure 2(c)) as this allows thresholds to be selected that are appropriate across all transformed images. Finally hysteresis thresholding (figure 2(d)) was performed to segment the image into vessel and background pixels. The upper threshold for hysteresis thresholding was set to 253 for all ray transform techniques, whilst the lower threshold was selected to give the highest accuracy across the database (RT-n: 234, RT-k: 235 and RT-nk: 230).

We compare pixels in the thresholded images with the ground truth, and calculate the performance of the technique. Two common metrics were calculated that are often used to test the strength of retinal blood vessel extraction: maximum average accuracy (MAA) and the area under the curve of a ROC graph (AUC).

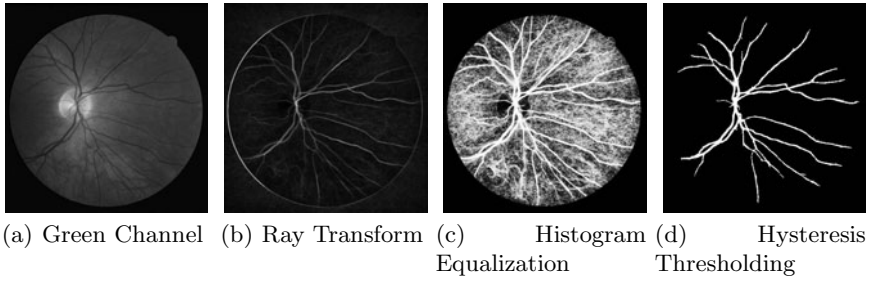


Fig. 2. Steps taken to extract retinal blood vessels

3.2 Results

This technique is capable of classifying pixels with accuracy comparable to a selection of contemporaneous unsupervised techniques. Figure 3 is a ROC graph showing the ability of each variant of our technique to classify vascular pixels correctly. Whilst all variants are significantly better than than the untransformed images, the aggregation of the two different types produces a marked increase in discriminatory ability, suggesting that different vascular structures are emphasised by each. The transforms themselves do not have prohibitive computational cost: the linear transform took 4.65s on average whilst the exponential transform took only 1.91s on a 2.53GHz processor. The transform with exponential refractive indices was significantly faster as rays adhered to the vessels more strongly and so the resultant image had less noise and converged more quickly. These times are, however, many times quicker than the rest of the classification process took.

In figure 4 a number of transformed images and their thresholded versions are displayed. The ray transform with linear refractive indices (figures 4(a) and (d)) is adept at highlighting larger features and those near the optic disc but introduces some noise across the image. In contrast, using exponential refractive indices (figures 4(b) and (e)) forces rays to adhere to small vascular features more strongly, introducing less noise across the whole image. However it highlights the fovea in the centre of the image and fails around the optic disc. The highlighting of the fovea was removed before classification. These complementary results allow the aggregated images (figures 4(c) and (f)) to have the strengths of both and negates some of the weaknesses. Most vessels highlighted by either version are present in the thresholded combined version, and it has been improved by automated removal of the highlighted fovea. Results for different lower thresholds retain more vascular pixels, but misclassified more noisy background pixels as well, a classification technique less vulnerable to noise would improve results considerably.

Table 1 shows the results for our technique as well as a range of others, also on the DRIVE database. Using our simple classification technique, the ray transformed images achieve superior MAA and AUC values than the original intensity images, implying that the ray transform has emphasised the vascular features to a greater extent than they had been before. Our method is comparable

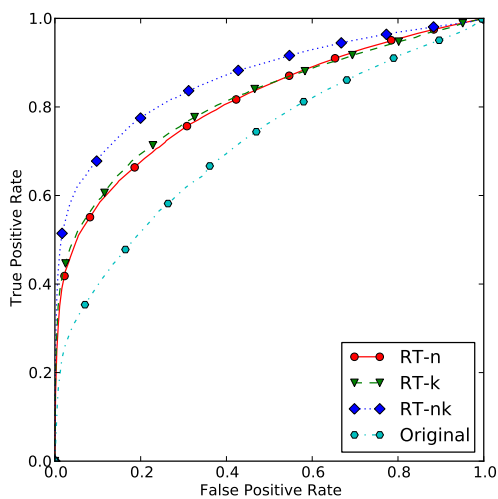


Fig. 3. ROC curve of the discriminatory ability of variants of our technique with the image ray transform

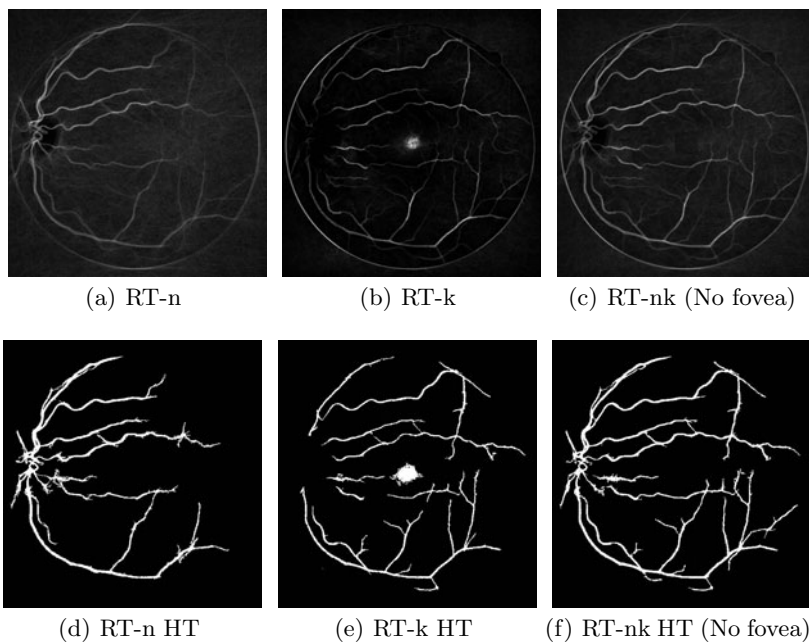


Fig. 4. Selecting retinal blood vessels by variants of the ray transform and hysteresis thresholding (HT)

Table 1. Maximum average accuracy (MAA) and area under the ROC curve (AUC) for our technique and others

Technique	MAA	AUC	Notes
Ray Transform-n	0.9082	0.8100	Linear Refractive Indices
Ray Transform-k	0.9118	0.8139	Exponential Refractive Indices
Ray Transform-nk	0.9237	0.8644	Aggregated RT-k and RT-n
Original Image	0.8863	0.7151	Original, our classification technique
All Background	0.8727	-	Most likely class
Chauhuri et al. [16]	0.8773	0.7878	Unsupervised
Zana et al. [26]	0.9377	0.8984	Unsupervised
Jiang et al. [36]	0.9212	0.9114	Unsupervised
Staal et al. [4]	0.9441	0.9520	Supervised
Soares et al. [5]	0.9467	0.9598	Supervised

to other unsupervised techniques, despite the simplicity of our classifier. As expected, supervised techniques produce superior results, but through use of the image ray transform as a preprocessor to a supervised learning method it should be possible to improve results further.

4 Discussion and Future Work

This work has shown that the image ray transform has considerable potential for use in enhancing detection of retinal blood vessels, and that performance comparable with other techniques can be achieved with use of a very simple classification technique. The transform is appropriate and successful at this task due to its inherent ability to detect tubular features such as blood vessels. Results show that the ray transformed images highlighted vascular features to a greater extent than they are in the original image, suggesting that use of the transform as a preprocessor of better classification methods will increase their performance. The transform is not computationally expensive in comparison with such classification techniques, and so does not increase their execution time significantly.

Future work will begin by applying the image ray transform as a preprocessor for current state of the art supervised and unsupervised methods, as it should be able to increase the performance of these techniques. Combining the two versions of the transform clearly improves results, but a better fusion would be able to use both results while excluding their weaknesses. Whilst the aim of this work was to detect blood vessels, it is clear that the ray transform also has potential to detect the fovea and the optic disc. This was a challenge in extracting vascular features, but detection of the fovea and optic disc are problems for which computer vision has been applied [16] and in the future the ray transform should also be evaluated for its ability to detect these features.

Acknowledgements

We gratefully acknowledge Alastair Cummings' EPSRC CASE studentship funded by the National Physical Laboratory (NPL).

References

1. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging* 8, 263–269 (1989)
2. Zana, F., Klein, J.: Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing* 10, 1010–1019 (2001)
3. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multi-threshold probing with application to vessel detection in retinal images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 131–137 (2003)
4. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging* 23, 501–509 (2004)
5. Soares, J., Leandro, J., Cesar Jr., R., Jelinek, H., Cree, M.: Retinal vessel segmentation using the 2-D Morlet wavelet and supervised classification bas. *IEEE Transactions on Medical Imaging* 25, 1214–1222 (2006)
6. Niemeijer, M., Staal, J., van Ginneken, B., Loog, M., Abramoff, M.: Comparative study of retinal vessel segmentation methods on a new publicly available database. In: *Proc. SPIE Medical Imaging 2004*, pp. 648–656 (2004)
7. Al-Rawi, M., Qutaishat, M., Arrar, M.: An improved matched filter for blood vessel detection of digital retinal images. *Computers in Biology and Medicine* 37, 262–267 (2007)
8. Zhang, B., Zhang, L., Zhang, L., Karray, F.: Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computers in Biology and Medicine* 40, 438–445 (2010)
9. Nixon, M.S., Liu, X.U., Direkoglu, C., Hurley, D.J.: On using physical analogies for feature and shape extraction in computer vision. *The Computer Journal* (2009)
10. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 629–639 (1990)
11. Liu, X.U., Nixon, M.: Medical image segmentation by water flow. In: *Proc. Medical Image Understanding and Analysis (MIUA 2007)* (2007)
12. Maragos, P.: PDEs for morphological scale-spaces and eikonal applications. In: Bovik, A.C. (ed.) *The Image and Video Processing Handbook*, 2nd edn., pp. 587–612. Elsevier Academic Press (2005)
13. Cummings, A.H., Nixon, M.S., Carter, J.N.: Circle detection using the image ray transform. In: *Int'l Conf. Computer Vision Theory and Applications, VISAPP 2010* (2010)
14. Cummings, A.H., Nixon, M.S., Carter, J.N.: A novel ray analogy for enrolment of ear biometrics. In: *4th IEEE Int'l Conf. on Biometrics Theory, Applications Systems, BTAS 2010* (2010)
15. Hill, F.: *Computer graphics using OpenGL*, 3rd edn., ch. 12, p. 678. Prentice Hall, Englewood Cliffs (2000)
16. Niemeijer, M., Abrāhoff, M., van Ginneken, B.: Fast detection of the optic disc and fovea in color fundus photographs. *Medical Image Analysis* 13, 859–870 (2009)

Automatic Liver Segmentation from CT Scans Using Multi-layer Segmentation and Principal Component Analysis

Hossein Badakhshannoory and Parvaneh Saeedi

School of Engineering Science, Simon Fraser University
Burnaby, BC, Canada

Abstract. This paper describes an automatic liver segmentation algorithm for extracting liver masks from CT scan volumes. The proposed method consists of two stages. In the first stage, a multi-layer segmentation scheme is utilized to generate 3D liver mask candidate hypotheses. In the second stage, a 3D liver model, based on the Principal Component Analysis, is created to verify and select the candidate hypothesis that best conforms to the overall 3D liver shape model. The proposed algorithm is tested for MICCAI 2007 grand challenge workshop dataset. The proposed method of this paper at this time stands among the top four proposed automatic methods that have been tested on this dataset.

Keywords: Liver segmentation, 3D organ reconstruction, mean shift segmentation, principal component analysis.

1 Introduction

An essential part of every computer assisted minimally invasive surgery is planning which is performed prior to the surgery. The planning process involves preparing a patient specific 3D model of the organ under surgery and its surroundings in order to provide the surgeon with a better understanding of the patient specific anatomy. This 3D model is based on image data of a patient that could be acquired from different modalities such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) or Ultrasound Imaging. In order to build the 3D organ model, various segmentation/extraction algorithms are applied on the pre-operative scans. A survey of segmentation methods for computer assisted surgery can be found in [1].

Liver is one of the most common human organs to undergo minimally invasive surgeries and therefore automatic liver segmentation/extraction from pre-operative scans for the purpose of 3D patient specific model reconstruction, is a highly needed task. In this paper an automatic liver segmentation algorithm is proposed. In this algorithm a multi-layer segmentation method that incorporates mean shift segmentation [2] is utilized to generate candidate boundary hypotheses of the liver. Also the shape of the liver is modeled by Principal Component Analysis (PCA). A training set of 20 liver mask volumes are used to

generate a PCA based liver space. The liver space is used to measure the similarity of volume mask candidates to the liver's overall shape. The best volume candidate that conforms to the PCA based model is the final result of our liver segmentation algorithm. The main contribution of this paper is an algorithm that encodes overall liver shape information using PCA to determine the proper parameter settings for a segmentation algorithm that leads to the best estimate of the reconstructed 3D liver model.

1.1 Previous Work

Various methods have been proposed in the literature for automatic and semi-automatic segmentation of liver. In general these methods either solely rely on the information in the input images to extract liver masks or they rely on training sets to incorporate shape information of the liver. Information available in the input images include: liver's texture/intensity image, spatial correlation of the 2D liver masks in consecutive slices, and location of the liver in abdominal area with respect to neighboring structures such as ribs. Methods in [3,4,5] are examples of the group of approaches that rely only on such information available in input images. There are also active contour model based methods such as [6] that rely on the information from the input images. In addition to the approach taken for utilizing input liver image information, methods in [7,8,9,10] use training sets to incorporate liver shape knowledge. These methods are in general more successful in extracting liver boundaries. Many of these methods use Active Shape Models (ASM) [11]. ASMs are statistical representations of the object's shape, which iteratively deform to fit to an example of the object of interest in a new image. ASM is based on PCA which has proven to be a strong tool to model organ shapes. The objective of the proposed work in this paper is to combine an intensity based segmentation method with a PCA based model approach to identify liver region in CT scan volumes. Such combination fuses strong characteristics of each one of these approaches to improve the overall quality of the liver extraction problem.

The rest of this paper is organized as follows; Section 2 describes the proposed algorithm in details. In Section 3 performance related issues and quantitative results of the proposed method are reviewed followed by conclusions in Section 4.

2 Proposed Method

The proposed method in this work incorporates liver intensity values, similarity of 2D liver masks in consecutive slices and a PCA based model.

The method has two modes: offline and online. In the offline mode, a PCA based model of the liver is generated from a training set including liver masks of 20 volumes. The output of the offline mode is a series of eigenvector (we name them eigenliver) that represent the liver space.

In online mode the PCA based model from the offline mode is used to assess and verify the quality of several liver mask that are generated through segmentation. The algorithm has two stages in this mode. In the first stage, several segmentation hypotheses are generated using an algorithm based on mean shift segmentation [2]. In the second stage of the proposed algorithm, the knowledge of the liver space is used to determine the similarity of generated liver hypotheses to the actual liver. The main idea at this stage is inspired by [12] where PCA was used to build a face space and measure the similarity of an input image to a face image. Here, every volume candidate is projected into the liver space (generated in the offline mode) and then reconstructed using the eigenlivers. The distance between each volume candidate and its reconstruction version by the liver space is a measure of shape fidelity of each volume candidate to overall liver shape and is used to identify the best candidate.

Details of the offline mode and stages of online mode are described in following subsections. In section 2.1 stage 1 of online mode is explained. In section 2.2 the offline mode where the PCA based liver model is created is described. In section 2.3 the second stage of online mode part of the algorithm is described.

2.1 Stage 1: Multi-layer Segmentation and Candidate Generation

In the first stage, the process starts from the middle slice of the CT scan volume where liver has its largest 2D surface. Mean shift segmentation [2] is used to extract the largest segment at this slice (our assumption in here is that the largest segment in this slice always corresponds to the liver). The remaining slices are then processed in two batches. Both batches start from the aforementioned middle slice but move in opposite directions. Every newly visited slice is segmented and the segment with the largest area overlap with the liver segment from the previous slice is marked as liver.

At each slice, the quality of segmentation is controlled by two parameters [2]: spatial resolution and intensity resolution. These parameters are usually set manually. Often a single set of values for each parameter will not be sufficient for a complete segmentation. Therefore in this stage the mean shift segmentation is applied over each slice several times using a range of different values for spatial and intensity resolution parameters. The range for spatial resolution is between 5 and 11 and the range for intensity resolution is between 3 and 25. We noticed that values outside these ranges would lead to either under segmentation or over segmentation of the liver region. We call each set of parameters for mean shift segmentation s . After applying the mean shift segmentation for each set, the boundary edges of all the segmented areas are extracted forming an edge map called EM . Different EM for each slice are added together to form one accumulative edge map (AEM) for each slice. The contrast of AEM is enhanced using Log transform. Contrast enhanced AEM is then thresholded to form an Enhanced Edge Map (EEM) which includes isolated connected regions. The threshold applied here is called β and is the variable parameter of the algorithm at stage 1. Naturally a fixed parameter would not provide the best results across

all volumes. Therefore, different values of β are utilized to generate a number of *EEM* images. This is described by equations 1 and 2:

$$AEM = \sum_s EM \tag{1}$$

$$EEM_{\beta}(x, y) = \begin{cases} 1 & \text{Log}(AEM(x, y)) > \beta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

At this point the segment (in *EEM*) with the largest area overlap with the liver segment from the previous slice is identified as liver segment. This segment is first morphologically opened by a small structuring element (a disk with radius of 4 pixels) to remove any excess small parts around its boundary. This procedure is followed by a morphological hole filling process to fill any small gaps within this segment. Sample *EEM* results with their corresponding segmented liver regions at different β values are shown in Fig. 1.

It must be noted that in the transverse direction, the 2D liver mask at each slice may consist of two or more pieces. Therefore, processing the CT volume at this direction could lead to missing liver components. For this reason the segmentation is performed at coronal and sagittal directions since it is observed that 2D liver masks at these two directions consist of one single piece.

After extracting all masks of a volume for a β value, these masks are stacked up together to form a candidate 3D liver mask volume. This implies that for each β value, one mask volume hypothesis is generated. The range of β used for

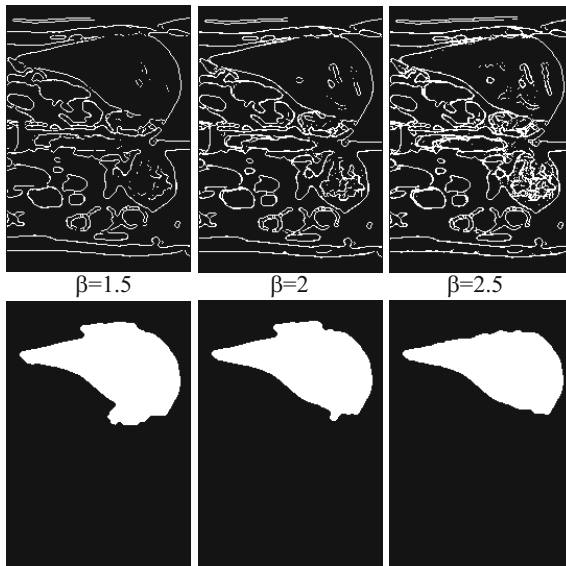


Fig. 1. Sample *EEM* for different β values (top row) with the corresponding extracted mask candidate (bottom row)

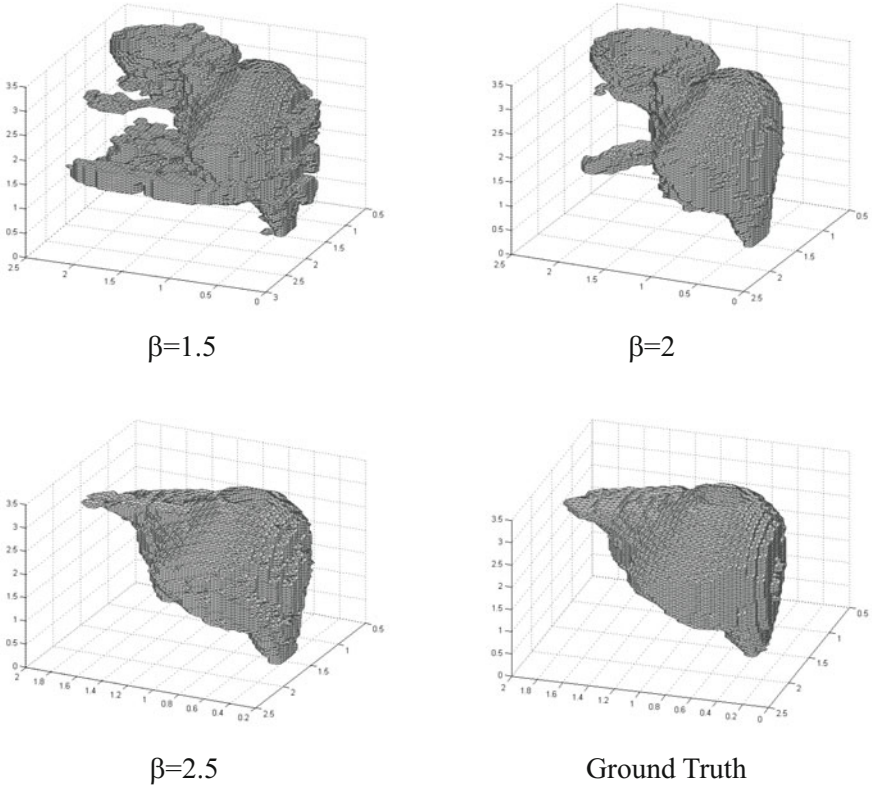


Fig. 2. Sample 3D volume mask representation for different β values

this stage is empirically chosen from the set 1.5, 1.6, 1.7, ..., 3.5. Fig. 2 shows three examples of 3D volumes generated for different values of β .

2.2 PCA Based Model Generation

The objective of this section is to create a model that represents the general shape of the liver volume. One approach for extracting shape information in a series of training liver mask volumes is to find the principal components of the distribution for the training set. This is equivalent to computing the eigenvectors of the covariance matrix of the set of liver mask volumes. Each volume contributes more or less to each eigenvector. Each eigenvector looks like a ghostly liver mask and therefore it is named eigenliver. Each new liver volume candidate can be approximated using a linear combination of the eigenlivers. The model is reconstructed as follows:

Let each 3D liver mask volume be a X by Y by Z array of 0s and 1s or equivalently a 1D vector of size $X \times Y \times Z$. If L_1, L_2, \dots, L_n , are 1D vectors that

represent liver mask volumes in the training set and ψ is their average then the distance of each volume to the average is defined by $\phi_i = L_i - \psi_i$. Here we look for the set of n orthonormal vectors u_i that best describe the distribution of the volume data. These vectors are the result of applying PCA over the entire training volumes and are eigenvectors of covariance matrix below:

$$C = \frac{1}{n} \sum_{i=1}^n \phi_i \phi_i^T = AA^T \quad (3)$$

Where $A = [\phi_1 \ \phi_2 \ \dots \ \phi_n]$. Matrix C is however very large and computing its eigenvectors is exhaustive. [12] introduces a computationally effective way to compute these vectors. Once eigenvectors/eigenlivers u_i are approximated, they are used to represent the liver space. This model is used (as explained in the next subsection) to measure the similarity of each volume candidate to the overall liver shape.

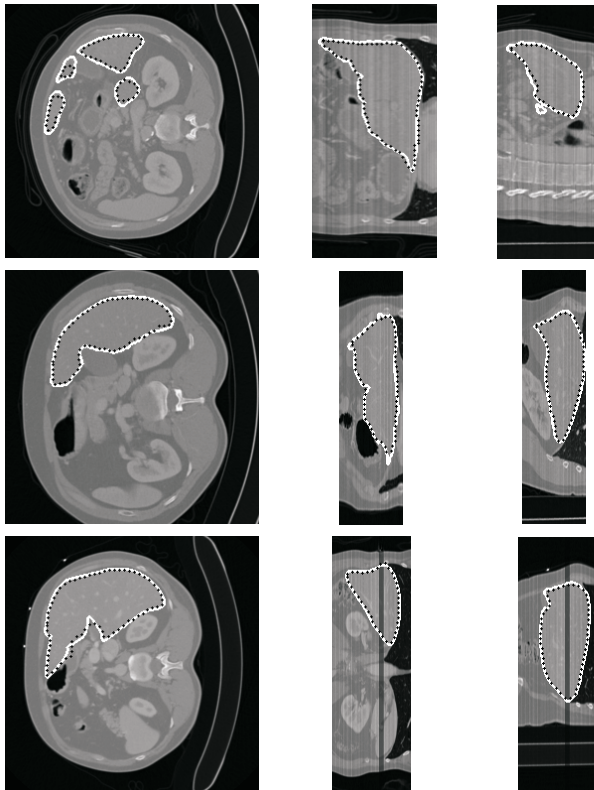


Fig. 3. Sample results of segmentation in transverse direction (left column), coronal direction (center column) and sagittal direction (right column), from training volumes of MICCAI 2007 dataset (white contours: ground truth boundaries, dotted contours: boundaries by the proposed algorithm)

2.3 Stage 2: Candidate Selection Based on Liver Shape Fidelity

As proposed by [12], where the face space knowledge is used to detect faces, we can use the liver space to measure the similarity of a liver mask volume to an actual liver shape. For this purpose, first mean adjusted input volume $\phi = L - \psi$ is projected onto the liver space. The result of this projection is vector $[\eta_1, \eta_2, \dots, \eta_n]$ where each η_i represents the contribution of each eigenliver in reconstructing the projected liver volume. The reconstructed liver is then computed as:

$$\phi_{rec} = \sum_{i=1}^n \eta_i u_i \quad (4)$$

For all candidate volume generated at stage 1, the Euclidean distance between their mean adjusted volume ϕ , and their reconstruction ϕ_{rec} is computed. The liver volume candidate with the minimum Euclidean distance is the volume with most fidelity to overall liver shape and therefore is chosen as the best representing liver mask by the proposed algorithm. Fig. 3 shows some 2D examples of the detected liver mask boundaries along with their corresponding ground truth segmentation.

3 Results and Discussions

To evaluate the proposed method, the datasets and evaluation metrics from MICCAI 2007 grand challenge workshop [13] are adopted. The dataset includes 20 training and 10 test volumes. Each volume consists of CT scans of size 512 by 512. For faster simulation the size of the images are rescaled to 256 by 256. The results for the test volumes are shown in Table 1. The definition of the metrics applied can be found in [13]. Brief descriptions of these metrics are as follows.

1. Volumetric Overlap Error, in percent. This is the number of voxels in the intersection of segmentation and ground truth divided by the number of voxels in their union, subtracted from 1 and multiplied by 100.
2. Relative Volume Difference, in percent. This is the total volume difference between segmentation and ground truth divided by total volume of ground truth multiplied by 100.
3. Average Symmetric Surface Distance, in millimeters. First the Euclidean distance between every bordering voxel in segmentation and the closest bordering voxel in ground truth is determined. Then the Euclidean distance between every bordering voxel in ground truth and the closest voxel in ground truth is determined. These two sets of distances are stored. The average of all these distances gives the Average Symmetric Absolute Surface Distance.
4. Symmetric RMS Surface Distance, in millimeters. This measure is similar to the previous measure but here the squared distances are used and the root of the average value is taken.
5. Maximum Symmetric Absolute Surface Distance, in millimeters. This measure is similar to the two previous measures but only the maximum of all the distances is considered.

Table 1. Quantitative results for the proposed method

Data set No.	Vol overlap error%	Score	Ave symm diff%	Score	Ave symm surface dist [mm]	Score	RMS symm surface dist [mm]	Score	Max symm surface dist [mm]	Score	Total
1	8.95	65.03	-4.69	75.04	1.32	66.98	2.52	65.06	18.37	75.83	69.59
2	9.90	61.32	-6.16	67.42	1.33	66.75	2.44	66.14	18.23	76.02	67.53
3	7.37	71.20	4.09	78.25	1.58	60.51	3.16	56.07	25.00	67.11	66.63
4	10.01	60.91	0.30	98.42	2.01	49.68	4.27	40.73	39.54	47.98	59.54
5	8.69	66.04	-5.28	71.92	1.45	63.71	2.83	60.73	23.73	68.78	66.24
6	8.12	68.28	-1.82	90.32	1.26	68.45	2.41	66.53	15.06	80.18	74.75
7	6.29	75.41	-2.81	85.08	0.86	78.41	1.67	76.85	12.99	82.91	79.73
8	7.83	69.40	-4.35	76.86	1.13	71.75	2.12	70.54	20.71	72.75	72.76
9	7.33	71.38	-1.03	94.54	0.90	77.39	1.78	75.23	20.77	72.67	78.24
10	10.90	57.41	-2.64	85.95	1.69	57.71	3.03	57.86	29.28	61.48	64.08
Mean	8.54	66.64	-2.44	82.38	1.35	66.14	2.62	63.57	22.37	70.57	69.86

The average runtime for extracting one liver mask using the proposed algorithm is 2 minutes using MATLAB 7.6.0.324 environment on a PC with an Intel Core 2 Duo (2 GHz) processor.

Presented results in Table 1 at this time stand among the top four automatic segmentation algorithms that have been tested on MICCAI 2007 dataset ([7], [9] and [10]) in terms of segmentation accuracy. Our method is also comparable to those by some of the interactive methods.

4 Conclusions

In this paper an automatic algorithm for extracting liver masks of CT scan volumes from abdominal area is proposed. The algorithm starts scanning the CT volumes in the coronal and sagittal directions to extract 2D liver mask candidates based on a multi-layer segmentation scheme that relies on mean shift segmentation. Multiple 3D liver mask candidates are generated as a result and the best candidate is chosen according to its similarity with its reconstructed version through a PCA based 3D liver model. This algorithm is novel in the way that it encodes overall liver shape information using PCA to determine the proper parameter setting for a segmentation algorithm that leads to the best estimate of the 3D reconstructed liver model. The idea of eigenliver can be utilized by previously proposed non-model based approaches to determine an optimal set of parameters that could potentially lead to better segmentation results.

References

1. Yaniv, Z., Cleary, K.: Image-guided procedures: A review. Technical report, Computer Aided Interventions and Medical Robotics (2006)
2. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transaction PAMI 24(5), 603–619 (2002)

3. Seo, K., Ludeman, L.C., Park, S., Park, J.: Efficient Liver Segmentation Based on the Spine. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 400–409. Springer, Heidelberg (2004)
4. Forouzan, A.H., Zoroofi, R.A., Hori, M., Sato, Y.: Liver Segmentation by Intensity Analysis and Anatomical Information in Multi-Slice CT images. Proc. of Int. Journal CARS 4, 287–297 (2009)
5. Susomboon, R., Raicu, D., Furst, J.: A Hybrid Approach for Liver Segmentation. In: 3D Segmentation in the Clinic - A Grand Challenge, pp. 151–160 (2007)
6. Pan, S., Dawant, B.M.: Automatic 3D segmentation of the liver from abdominal CT images: a level-set approach. In: SPIE Medical Imaging, vol. 4322, pp. 128–138 (2001)
7. Kainmuller, D., Lange, T., Lamecker, H.: Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: 3D Segmentation in the Clinic - A Grand Challenge, pp. 109–116 (2007)
8. Heimann, T., Meinzer, H.P., Wolf, I.: A Statistical Deformable Model for the Segmentation of Liver CT Volumes. In: 3D Segmentation in the Clinic - A Grand Challenge, pp. 161–166 (2007)
9. Wimmer, A., Soza, G., Hornegger, J.: A Generic Probabilistic Active Shape Model for Organ Segmentation. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 26–33. Springer, Heidelberg (2009)
10. Wimmer, A., Hornegger, J., Soza, G.: Implicit Active Shape Model Employing Boundary Classifier. In: ICPR, pp. 1–4 (2008)
11. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active Shape Models - their Training and Application. CVIU 61(1), 38–59 (1995)
12. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience (1991)
13. van Ginneken, B., Heinmann, T., Styner, M.: 3D Segmentation in the Clinic - A Grand Challenge. In: MICCAI Workshop Proceedings (2007)

Low Cost VR Meets Low Cost Multi-touch

Dane Coffey, Fedor Korsakov, and Daniel F. Keefe

Department of Computer Science and Engineering
University of Minnesota, Minneapolis MN 55455, USA
{coffey,korsakov,keefe}@cs.umn.edu
<http://ivlab.cs.umn.edu>

Abstract. This paper presents the design, implementation, and lessons learned from developing a multi-surface VR visualization environment. The environment combines a head-tracked vertical VR display with a multi-touch table display. An example user interface technique called Shadow Grab is presented to demonstrate the potential of this design. Extending recent efforts to make VR more accessible to a broad audience, the work makes use of low-cost VR components and demonstrates how these can be combined in a multiple display configuration with low-cost multi-touch hardware, drawing upon knowledge from the rapidly growing low-cost/do-it-yourself multi-touch community. Details needed to implement the interactive environment are provided along with discussion of the limitations of the current design and the potential of future design variants.

Keywords: VR, low cost, multi-touch, multi-surface, shadow widgets.

1 Introduction

Recently, the widespread availability of affordable 3D input and 3D display devices, such as the Nintendo Wii motion controller, NaturalPoint Optitrack optical tracking, and 3D DLP TVs, has opened up new applications for VR, making it increasingly practical for VR to be used in labs, offices, and classrooms without the significant expense that has been typical of VR hardware for so many years. In parallel to these advances, an exciting hardware revolution is occurring in the area of multi-touch display devices. From the large-scale Microsoft Surface to the hand-held Apple iPad, many multi-touch devices are now commercially available, and a growing community of researchers and hobbyists has established viable approaches (e.g. [4]) for building multi-touch displays using low-cost commodity components.

This paper explores the potential of novel VR hardware setups that are made possible by combining low-cost VR with low-cost multi-touch. In particular, we introduce a new multi-surface VR display configuration that combines a large, table-sized multi-touch surface with a VR stereoscopic display wall. We believe that the resulting hybrid environment can provide rich and valuable new modes of interacting with many VR applications without significantly increasing the cost of current affordable VR hardware solutions.

Compared to traditional VR input devices used with 3D applications (e.g. six degree-of-freedom trackers, VR wands), at first, multi-touch surfaces may appear limited, in that they fundamentally capture just 2D input on the surface. However, there are several reasons why multi-touch surfaces may be particularly well suited to VR applications. First, despite the fact that the touch events generated are 2D, multi-touch surfaces actually support very high-dimensional input because they support simultaneous tracking of tens to hundreds of touch points. Second, recent research has demonstrated that some 3D tasks (e.g. precise, relative positioning) are better performed using 2D views rather than 3D [6,14]; it follows that 2D multi-touch surfaces may be better suited for certain 3D interactions than six degree-of-freedom (DOF) devices. Finally, many of the most successful current multi-touch applications are built upon fluid, gestural, and direct-manipulation styles of input, and these are also common characteristics of many VR applications. Thus, we believe multi-touch input can be just as expressive as (or more expressive than) traditional VR inputs, be better suited to certain 3D tasks, and integrate seamlessly with many common VR interfaces and interaction metaphors.

This work makes several contributions centered around exploring the potential of combining low-cost VR with low-cost multi-touch hardware: 1.) We present the design and implementation of a multi-surface VR hardware setup combining a vertical stereo surface with a horizontal table multi-touch surface. 2.) We demonstrate an example 3D user interface technique for manipulating objects floating above the table, illustrating the potential of multi-touch as input for VR. 3.) We discuss the design lessons and insights learned through implementing the VR setup.

2 Related Work

Several previous systems have explored interaction techniques that utilize multiple display surfaces. (Lachenal and Dupuy-Chessa present an ontology of this design space [9].) Closely related to our work, Ajaj et al. [1] combined a horizontal table and vertical screen and used a map on the table to control the camera view of a 3D scene displayed on the vertical screen. Aliakseyeu et al. [2] use a similar smaller scale setup in which a tracked frame prop is moved above a tablet to control the view shown on a stereo screen. Spindler et al. [12] also explore above-the-table interaction using a tracked paper prop as an additional display surface. Hachet and Guitton [3] used a tracked tablet mounted on a pillar to enable interaction in large-display VR environments. LaViola et al. [10] used a pair of slippers to interact with the floor surface in a CAVE environment, freeing the user’s hands for other tasks performed above the floor in the 3D space of the CAVE.

Although these systems all utilize a multi-surface configuration and mix 2D inputs and displays with 3D displays, we know of no existing multi-surface configuration that combines multi-touch table input with a head-tracked, stereoscopic wall display. Our work has focused on this configuration because we believe it can be particularly effective for VR. For example, the configuration we propose

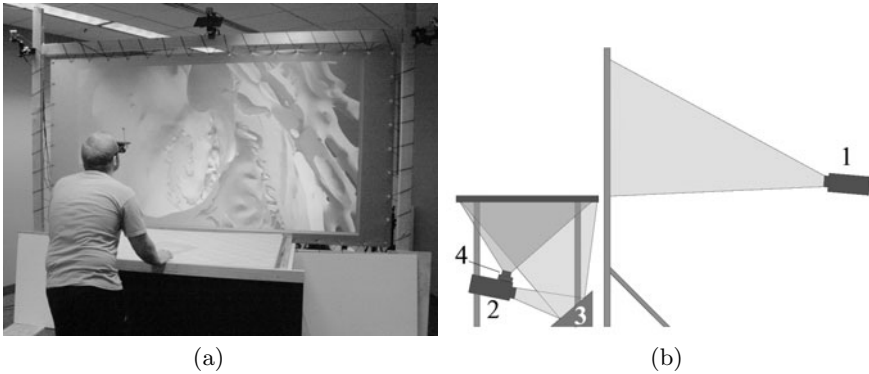


Fig. 1. Left: using the multi-surface VR setup to explore a biomedical environment. Right: the rear projected image on the vertical screen originates from a 3D projector (1) mounted behind the screen. The horizontal table’s image originates from a projector (2) mounted under the table and bounces off the front surface mirror (3). An IR camera (4) is mounted on top of the table’s projector and is used to track points of contact on the table.

can be used to make virtual objects appear to float in the air above the multi-touch table, which leads naturally to the interacting on the table with a shadow or projection of the 3D scene, a metaphor we explore in the interface described later in the paper.

Within the multi-touch research community, Han’s seminal work on frustrated total internal reflection (FTIR) [4] has spurred a variety of follow-on research and a growing community motivated to design and support low-cost multi-touch hardware. An important result from this community is the TUIO protocol for describing and transmitting touch events [8]. Servers and clients that implement this protocol are available for a variety of platforms [7], and we have found that this infrastructure makes it easy to integrate touch input within a typical VR application structure. For example, processing of TUIO events distributed to an application via a network connection can be integrated within a VR application alongside typical VR event handling routines, e.g. handling tracking data distributed via VRPN.

3 Low-Cost, Multi-touch, Multi-surface VR

Our multi-touch, multi-surface VR system consists of an FTIR multi-touch table working in combination with a stereo image rear-projected onto a vertical screen adjacent to the table, seen in Figure 1. FTIR-based multi-touch systems rely on IR light shining into the edges of a sheet of transparent material. The light is reflected internally, but the reflection can be disrupted if another object makes sufficiently close contact – an effect enhanced by using a thin, flexible material known as a compliant surface [4]. A camera with appropriate filters can then detect the touch as a blob of infrared light. A projector displays an image onto

the projection surface, which often serves as the base of the compliant surface. The main components of the hardware are described in the following sections.

3.1 Interactive Surface

The interactive surface consists of a sheet of acrylic approximately 4' x 3' in size, with edges enclosed in an aluminum channel. The channel contains a ribbon of surface-mounted 850 nm IR LEDs. The sheet of acrylic is covered with drafting film, which acts as the projection surface. In order to provide FTIR upon touch, the film is double-coated with a silicone-based compliant surface material on the bottom side [13]. The top side is laminated for protection. Early in the design process we observed that the laminated surface was too tacky for comfortable tactile interaction (as fingers may stick to it), so a layer of un-laminated drafting film was added on top.

3.2 Optical Touch-Tracking

Accurate and fast optical tracking of touches on the interactive surface is crucial to the overall performance. A monochrome Firewire camera with a fisheye lens (170° horizontal FOV) is mounted on the projector stand under the table (Figure 1) and provides 30 FPS at VGA resolution. In order to minimize the need for post-processing, factors such as stray visible light are best addressed at the hardware level, and for this purpose a Schott RG830 longpass filter is attached directly above the lens. Community Core Vision software (an open-source TUIO server) [11] processes the video feed and sends TUIO events to the VR application. The processing time per frame generally is in the vicinity of 6 ms, providing smooth and fluid interaction.

3.3 Table Projection

The majority of the optical components are positioned under the table in order to reduce the number of protrusions around the interactive surface. This makes image projection challenging, due to the need to achieve a stable, secure position and adequate ventilation for the projector. Our implementation uses a short throw projector, a ViewSonic PJD5351, mounted upside down. A first-surface mirror is used to increase the projector beam travel distance to fill the entire table surface. The resulting configuration of the table's projector and mirror are shown in Figure 1 (b).

3.4 Vertical Surface

The vertical surface displays a three-dimensional stereoscopic view of the VR application. A midrange stereo projector (DepthQ DQ-WXGA) mounted behind the screen generates the image, as seen in Figure 1. Shutter glasses are worn by the user, and the IR synchronization emitter is mounted behind the vertical

screen. A 9' x 4.5' Rosco Grey projection material is used for the display screen and is mounted in a simple wooden frame by suspending it with an elastic cord wrapped around the frame and through grommets placed around the edge of the screen. The user's head is tracked by 6 Optitrack NaturalPoint cameras mounted on both the wooden frame and the surrounding ceiling.

4 Shadow Grab: A Representative Interface Technique

This section describes Shadow Grab, an interaction technique that is representative of the type of VR interfaces that can be made possible by combining multi-touch with VR using the multi-surface configuration we describe. In Shadow Grab, the user interacts with a 3D scene that appears to float above the table by manipulating its shadow on the table surface. This concept builds upon early work by Herndon et al. [5], who showed using a mouse-based desktop system that 3D scene interaction and manipulation is facilitated through interacting with a set of 2D shadow widgets. This style of shadow interaction is ideal for the proposed multi-surface hardware setup because the rich 2D multi-touch input provides a direct way to manipulate the shadow.

Shadow Grab has two visual components. The first component is the floating 3D object displayed in stereo on the vertical screen, which, in our example, is an anatomical model of an aorta, as shown in Figure 2. The second component is

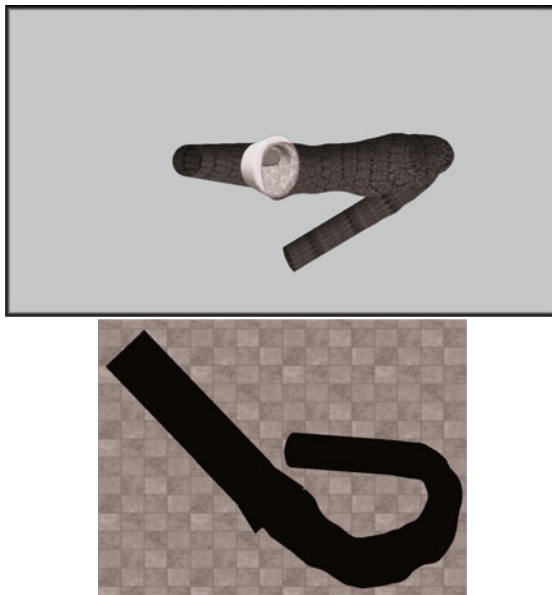


Fig. 2. The Shadow Grab interface is used to manipulate 3D objects floating above the tabletop by interacting with their shadow's projected on the table. Using multi-touch gestures, the objects can be translated, scaled and rotated.

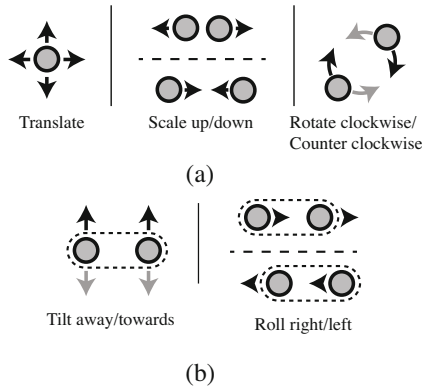


Fig. 3. The set of multi-touch manipulation gestures used in the Shadow Grab interface. Gestures that translate, scale, and rotate the object in the plane of the table are shown in (a). Gestures that tilt and roll the environment are shown in (b).

the shadow cast by the 3D object downward onto the table surface, also shown in Figure 2. The shadow projection provides a 2D view of the outline of the object, which the user can manipulate using multi-touch gestures. Using the gestures illustrated in Figure 3 (a), the object can be scaled, translated and rotated. A single point of contact allows translation in the horizontal and depth dimensions; the height of the 3D object above the table remains fixed. Two points of contact are used to scale the object and to rotate it about a vertical axis. Each of these interactions is similar in spirit to the 2D direct manipulation multi-touch interfaces that have become popular in photo browser and other similar applications. Figure 3 (b) illustrates an extension to these traditional gestures that we have implemented to support 3D interaction. Moving two points of contact together in the horizontal direction rolls the scene about an axis pointing into the vertical wall (the depth dimension), and moving two points of contact together in the depth dimension rotates the scene about a horizontal axis, tilting it toward or away from the viewer.

Initially, we implemented the shadow as a parallel projection of the scene downward along vertical projector rays. In practice, we found that when users interacted with the shadow positioned near the front of the table, this brought the 3D “floating” version of the scene very close to the user’s head, making the experience uncomfortable to many users. Our solution to this issue is to apply a slight offset to the shadow, moving it six inches closer to the user than it should be. This dramatically improves the usability of the interface, and surprisingly, none of the users we have worked with have noticed this offset.

5 Discussion

5.1 Manipulating Objects in Front of Low Cost Displays

We discovered as we built this VR system that the current generation of low-cost VR displays is quite limited in its ability to make the illusion of a scene floating

in front of the vertical display work. Our original design utilized a Samsung 60" DLP stereo television as the vertical stereo surface, but we found the significant ghosting that occurred with this display when objects were positioned more than a few inches out of the screenplane made the stereoscopic imagery very difficult to fuse, and hence, the technique was almost unusable with the Samsung display. For this reason, our current implementation uses a midrange stereo projector, which provides a higher quality image, albeit at an increase in cost. We anticipate that as 3D technology progresses, the quality of stereoscopic images produced by low-cost displays will improve considerably, and we submit that achieving high quality rendering for objects that appear significantly in front of a vertical VR display is a particularly important benchmark for low-cost display manufacturers and researchers to achieve, as it can enable a new class of multi-surface interfaces in the style of Shadow Grab.

5.2 Six Degree of Freedom Tracking

In the current hardware setup six OptiTrack NaturalPoint cameras are used for 6-DOF head tracking. Depending upon the application and the budget, 6-DOF tracking could play either an enhanced or a decreased role in the VR environment. With techniques such as Shadow Grab, 2D multi-touch interfaces could provide an affordable replacement for more traditional VR 6-DOF direct manipulation interfaces, thus, the optical tracking system used in our implementation could be smaller (so as to only capture limited head movement) or eliminated (if head tracking is not essential to the application). This would greatly reduce the overall cost of the VR environment.

On the other hand, it is also possible to make increased use of the 6-DOF tracking as compared to our current setup, and this provides an exciting direction for future research. For example, the multi-touch gestures could be combined with 6-DOF hand tracking above the surface.

5.3 The Choice of Touch-Tracking Method

FTIR is a fairly mature technology but by no means the only way to capture multi-touch input. Other methods, such as diffuse illumination (DI) and diffuse surface illumination (DSI), may also be considered. An important consideration in selecting the most appropriate technology is the size of the surface. Our design called for a rather large surface, and in this situation, FTIR is more cost-effective than DSI (which requires a more expensive type of acrylic material). For large surfaces, FTIR tends to provide more uniform performance than DI (which may suffer from uneven IR light distribution).

5.4 Infrared Interference Reduction

The current setup relies heavily on 850 nm IR lights. Touch-tracking, head-tracking, and the synchronization of the shutter glasses all use IR light close

to this wavelength, and cross-interference can cause spurious inputs to the interactive surface or disrupt the synchronization of the shutter glasses. This can be counteracted in 2 ways. First, the brightness of the Optitrack IR LEDs can be decreased, and the shutter glasses synchronization strobe can be carefully positioned. Second, an OptiHub device recently developed by NaturalPoint can be used to synchronize the cameras' IR LEDs with the synchronization strobe for the shutter glasses.

5.5 Seamless Projection

The metal frame of the multi-touch table used in our current implementation creates a small seam between the table and the vertical wall where they touch. An alternative design might be able to avoid this seam through engineering a seamless joint, perhaps inspired by the joints used in CAVE's and/or the Responsive Workbench.

5.6 Extending to Support Tangible Interfaces

Tangible interaction props marked with fiducials are more commonly associated with DI systems (e.g. Reactable [8](#)) as opposed to FTIR systems. Nevertheless, it is possible to augment the interaction with FTIR systems by using IR-emissive tangible objects, as well as narrow-beam IR sources. The table would provide a natural resting place for these interaction props, thus, an interesting area of future work is to explore the potential of prop-based user interfaces tailored to this multi-surface VR configuration.

6 Conclusion

We have presented a low-cost system that combines a 2D tabletop multi-touch interface with virtual reality. In addition to the implementation details and design lessons learned in creating this system, we provide an example 3D user interface that demonstrates the potential of the unique hardware configuration. Since multi-touch surfaces provide many simultaneous touch events, they are particularly rich (high-dimensional) input devices that may enable many new styles of interacting with VR applications. If appropriate mappings can be devised to interpret multi-touch input within the context of VR applications, then the rich input afforded by multi-touch surfaces may prove beneficial in a variety of VR applications. Combining multi-touch and VR technologies is becoming increasingly practical given the growing online community for both hardware and software support. Due to the combination of the relatively low cost, rich interaction, and existing open standards, the VR community is poised to take advantage of multi-touch surface computing.

References

1. Ajaj, R., Vernier, F., Jacquemin, C.: Navigation modes for combined Table/Screen 3D scene rendering. In: ACM International Conference on Interactive Tabletops and Surfaces, pp. 141–148 (2009)
2. Aliakseyeu, D., Subramanian, S., Martens, J., Rauterberg, M.: Interaction techniques for navigation through and manipulation of 2D and 3D data. In: Proceedings of the Workshop on Virtual Environments 2002, pp. 179–188. Eurographics Association, Barcelona (2002)
3. Hachet, M., Guitton, P.: The interaction table: a new input device designed for interaction in immersive large display environments. In: Proceedings of the Workshop on Virtual Environments 2002, pp. 189–196. Eurographics Association, Barcelona (2002)
4. Han, J.Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, pp. 115–118. ACM, Seattle (2005)
5. Herndon, K.P., Zeleznik, R.C., Robbins, D.C., Conner, D.B., Snibbe, S.S., Dam, A.V.: Interactive shadows, pp. 1–6 (1992)
6. John, M.S., Cowen, M.B., Smallman, H.S., Oonk, H.M.: The use of 2D and 3D displays for shape-understanding versus relative-position tasks. *Human Factors* 43(1), 79–98 (2001)
7. Kaltenbrunner, M.: Tuio implementations (2010), <http://www.tuio.org/?software>
8. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: Tuio - a protocol for table based tangible user interfaces. In: Gibet, S., Courty, N., Kamp, J.-F. (eds.) *GW 2005. LNCS (LNAI)*, vol. 3881. Springer, Heidelberg (2006)
9. Lachenal, C., Dupuy-Chessa, S.: Ontology for multi-surface interaction. In: Proceedings of IFIP Conference on Human-Computer Interaction: INTERACT 2003, pp. 447–454 (2003)
10. LaViola, J., Acevedo, D., Keefe, D.F., Zeleznik, R.: Hands-free multi-scale navigation in virtual environments. In: Proceedings of ACM Symposium on Interactive 3D Graphics 2001, pp. 9–15 (2001)
11. Moore, C., Sandler, S.: Community core vision (2010), <http://nuicode.com/projects/tbeta>
12. Spindler, M., Stellmach, S., Dachsel, R.: PaperLens: advanced magic lens interaction above the tabletop. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, pp. 69–76. ACM, Banff (2009)
13. “Tinkerman”: Tinkermans method - casting textured silicone (July 2008), <http://nuigroup.com/forums/viewthread/2383/>
14. Tory, M., Kirkpatrick, A.E., Atkins, M.S., Moller, T.: Visualization task performance with 2D, 3D, and combination displays. *IEEE Transactions on Visualization and Computer Graphics* 12(1), 2–13 (2006)

A. Appendix

A.1 Parts Listing

- Table hardware, including projector (Approximately \$2000)
 - Viewsonic PJD5351 projector
 - Unibrain Fire-i BW board camera with 107° horizontal FOV lens

- Acrylic sheet (4' x 3' x 3/8")
 - Reel with 5 m adhesive-backed ribbon of surface-mounted 850 nm IR LEDs
 - Power supply for LEDs
 - Rubber spacers for LEDs (4" setting block, cut as necessary)
 - Aluminum channel (2 pieces 3/8" wide, 3/4" deep, 98" long each)
 - Drafting film (two 3' x 4' sheets)
 - Clear silicone sealant and Xylene or comparable solvent
 - First surface mirror (408 x 608 mm, 4-6 wave)
 - Schott RG830 filter (1" diameter)
 - Adjustable table legs
- Stereoscopic wall hardware in original design (Approximately \$1500)
 - 60-inch Samsung 3D DLP TV
 - Synchronization device and shutter glasses
 - Replacement stereoscopic wall hardware in revised design (Approximately \$5000 for projector and glasses plus \$200 for the projection screen and frame)
 - DepthQ DQ-WXGA stereoscopic projector
 - Synchronization device and shutter glasses
 - Rosco Grey (9' x 4.5' sheet)
 - Wood beams for screen frame
 - Bungee cord (1/8" diameter) and grommets (1/2" inner diameter)
 - Computer Workstations/Servers (Approximately \$2000)
 - Community Core Vision and OptiTrack servers (Any desktop machine will suffice)
 - 3D application workstation (stereo capable graphics card needed in the revised design, but not in the original.)
 - Head-tracking system (Approximately \$5000 for our 6-camera system, but a smaller system for several hundred dollars would suffice.)
 - Optitrack camera server (any desktop machine will suffice)
 - Optitrack NaturalPoint 6 camera system

A.2 Construction Notes

It is strongly advised to prototype the placement of the interactive surface projector, the camera and the mirror using paper triangles (cut out to scale) that closely estimate beam shapes and focal ranges. The paper triangles can be folded to the required mirror size and angle for the projector beam.

Acrylic sheet edges may need to be polished. The channel should be cut to provide a frame for the sheet with additional space sufficient to place the LEDs on the inside of the channel. Pre-mounted LEDs greatly facilitate the assembly, but require rubber spacers between LEDs and the acrylic sheet to prevent the sheet from crushing the LEDs. The compliant surface material is a medium thickness mixture of clear silicone sealant and xylene painted onto the drafting film in two coats with a roller. Additional instructions and discussion can be found at NUI Group forums [\[13\]](#).

IQ-Station: A Low Cost Portable Immersive Environment

William R. Sherman¹, Patrick O’Leary², Eric T. Whiting², Shane Grover²,
and Eric A. Wernert¹

¹ Advanced Visualization Laboratory, Pervasive Technology Institute,
Indiana University

² Center for Advanced Modeling and Simulation, Idaho National Laboratory

Abstract. The emergence of inexpensive 3D-TVs, affordable input and rendering hardware and open-source software has created a yeasty atmosphere for the development of low-cost immersive systems. A low cost system (here dubbed an *IQ-station*), fashioned from commercial off-the-shelf technology (COTS), coupled with targeted immersive applications can be a viable laboratory instrument for enhancing scientific workflow for exploration and analysis. The use of an IQ-station in a laboratory setting also has the potential of quickening the adoption of a more sophisticated immersive environment as a critical enabler in modern scientific and engineering workflows. Prior work in immersive environments generally required special purpose display systems, such as a head mounted display (HMD) or a large projector-based implementation, which have limitations in terms of cost, usability, or space requirements. The alternative platform presented here effectively addresses those limitations. This work brings together the needed hardware and software components to create a fully integrated immersive display and interface system that can be readily deployed in laboratories and common workspaces. By doing so, it is now feasible for immersive technologies to be included in researchers’ day-to-day workflows. The IQ-station sets the stage for much wider adoption of immersive interfaces outside the small communities of virtual reality centers. In spite of this technical progress, the long-term success of these systems depends on resolving several important issues related to users and support. Key among these issues are: to what degree should hardware and software be customized; what applications and content are available; and how can a community be developed?

1 Introduction

Immersive environment (IE) systems, also known as virtual reality (VR) systems are a unique medium that offers the opportunity for natural interactions with a simulated world. The potential for natural interaction stems from taking into account the physical movements of the user when rendering the simulated, or virtual, world. This feature is referred to as “*physical immersion*” [1].

Since the early 1990’s, immersive environment systems have closely followed the predicted path defined by Gartner’s Hype Cycle for new technology [2] depicted in Figure 1. The technology trigger was the 1989 release of commercially

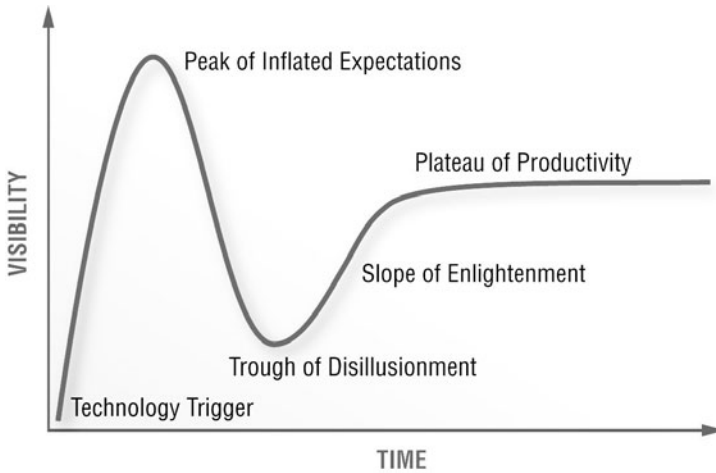


Fig. 1. Gartner's Hype Curve for new technology. Immersive technologies (aka VR) have moved beyond the *Trough of Disillusionment* onto the *Slope of Enlightenment*.

produced equipment enabling virtual reality — such as stereoscopic displays, glasses and three-dimensional position tracking systems. By the mid 1990's, immersive environments reached the peak of inflated expectations, where the technology was, and still is, frequently compared to Star Trek's Holodeck. Predictably unable to live up to these expectations, immersive environments fell into the trough of disillusionment. Weaknesses in the technology of the time included insufficient networking, rendering and computational power. Improvement in these areas was inevitable according to Moore's Law, but the computer gaming market and industry further accelerated advancements. The resulting improvements pushed the technology up the slope of enlightenment creating an "immersive environments renaissance" by the end of the 2000–2010 decade. One can argue that there is a substantial amount of work ahead to reach immersive environments' plateau of productivity for scientific and engineering workflows, but the technology seems to have turned the corner, putting it on the right side of Gartner's Hype Cycle.

1.1 Demonstrated Usefulness of Immersive Environments

Successes in using immersive environments to enhance scientific and engineering workflows have been demonstrated at several laboratories. Researchers at Brown University explored the effectiveness of microbiologists (graduate students, research associates and professors) in performing practical workflow tasks [3]. Their study compared the use of a CAVETM, an immersive fishtank display and a standard desktop environment in analyzing three-dimensional confocal microscopy data. The testing of common tasks for microbiologists exhibited a statistically significant improvement as immersion increased, moving from a desktop up through a CAVE.

Atmospheric researcher Gil Bohrer reports that the CAVE at Duke University was instrumental in aiding his research by acting as the catalyst for a discussion on how forest canopies affect local atmospheric air flow [4]. Once the initial insight was obtained, the CAVE had served its purpose and the workflow returned to traditional desktop visualization tools.

These efforts extend earlier work from the National Center for Supercomputing Applications (NCSA) where the Crumbs project used a CAVE as a visualization tool for a variety of volumetric datasets, including MRI and confocal instruments [5]. This early effort clearly revealed that for some data analysis, the immersive system provided both crucial insights and significant time savings in analyzing the data, and became a part of the research workflow.

1.2 Missing Links

The need for more natural and effective interfaces for immersive environments is real and growing. As described in the 2006 NIH/NSF report on visualization research challenges [6], “Fluid interaction requires that we create user interfaces that are less visible to the user, create fewer disruptive distractions, and allow faster interaction without sacrificing robustness.” What Johnson et al. were calling for is in fact the essence of immersive interfaces. This is the “missing link” that immersive environments provide. This need was echoed in an NSF sponsored workshop by presenters and attendees alike [7].

Another critical “missing link” is the availability of immersive displays. Low-cost VR workstations can bring immersion to the masses almost like the advent of the personal computer made computing available to the broad public. Projector-based immersive environments are generally costly and require a large amount of space. HMDs are often cumbersome, small, and focused on an individual. The 3D-TV based IQ-station represents a middle ground that provides a solution available and usable by a much wider audience.

2 Development of Low Cost Immersive Systems

The Desert Research Institute (DRI) was among the first to implement a low-cost VR workstation using the first generation of commodity 3D-TV screens. (Similar efforts were also taking place at the University of California, Davis.) Development of this system evolved over the course of the past three years. The earliest prototypes began as systems that would augment DRI’s primary immersive facilities that included a 4-sided FLEX/CAVETM with a 6-sided system in the works. The ability to purchase a large stereoscopic display at a local retail outlet was the catalyst around which the system was created. Combined with a reasonably priced turnkey 3D position tracking system, low-cost immersive systems were made viable. Indiana University (IU) built upon these initial experiences at DRI. Subsequently the system has been dubbed the Inexpensive Interactive Immersive Interface (*I-quaded-*, or *IQ-*) *station*.

The IQ-station is an instantiation of fishtank-style VR — a stereoscopic single screen display with head tracking. Our implementation of the IQ-station

also derives from the community building aspect of the early CAVELib and GeoWall communities [8]. Our project brings together the combined benefits of the larger screen size and community of the GeoWall, and the physical immersion of VR. By building on the larger 3D-TV screens, more users can gather around the IQ-station, enabling more collaborative discussions. By fostering a community around a particular recipe the IQ-station further differentiates itself from a generic fishtank system that any individual might construct. Providing a specific recipe enables the concept to proliferate more quickly by allowing new groups to benefit from the testing and analysis of individual components without duplicating these efforts. Thus, the Idaho National Laboratory¹ (INL) leveraged the IU design to deploy the technology in a research engineering environment. Of course, tweaking a recipe for local requirements or tastes is appropriate. For example, the default screen orientation for an IQ-station is vertical, but when using flat-screen displays, other choices are possible — such as drafting style (ala the ImmersaDesk™) or table-top (Responsive WorkBench™). Thus the advance, is both in form and increased functionality through community.

Integrating a turnkey system benefits from a formal prescription for hardware assembly, middleware software, and end-user applications. In the following subsections, we describe some of the available options, and choices made as an evolutionary outcome of usage, experience, and technology improvements of the past two years.

2.1 Hardware Recipes

There is a wide range of hardware options possible, depending on the desired ergonomics and budget constraints. On the very low-end, many home users may be able to assemble a VR system with hardware components already at home (the you-pick-it recipe). Stepping up from there, the next tier is at about \$10K–\$15K (the home-style recipe), and for the gourmand, \$25K is sufficient to produce a highly-capable, beefier system.

Given that physical immersion is a necessary feature of a VR workstation, the 3D position tracking system is crucial. This is the component that gives the system the ability to render the virtual world from the changing perspective of a user as they move. A visual display is also required of course, and usually for immersive experiences a stereoscopic display is preferred. For fixed-screen displays (such as monitors and projectors), the stereo effect usually requires a pair of specialized glasses for each viewer. Finally, a computing system capable of simulating and rendering the virtual world is needed. Of course, a way to mount or position the system is handy, so our recipes include this as a component as well.

There are a number of ways to produce low-cost 3D position tracking. A method popularized by CMU researcher Johnny Lee [9] uses the internal camera and software of a Nintendo Wii™ game controller (aka “Wiimote”) to calculate

¹ References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government, any agency thereof, or any company affiliated with Idaho National Laboratory.

the translational offset between a pair of infrared LEDs and the Wiimote. Similarly, one can take a video camera connected to a computer and use a software toolkit to analyze the view and make a position calculation. One freely available toolkit that uses fiducial markers to accomplish this is the ARToolkit [10] [11]. A third option that also makes use of video technology is the OptiTrack™ system from NaturalPoint that uses multiple infrared cameras to track clusters of retro-reflective markers. Marker reflections are used to calculate the location and orientation values. This is the most expensive of the listed technologies, but at about \$5K for a 6-camera set, it is still quite reasonable — a \$3K, 3-camera option is also available, but is less robust, working better in a ceiling mounted configuration, thus sacrificing mobility. As this software is restricted to Microsoft Windows, users of other operating systems should either add approximately \$1K to the cost of this system for a separate tracking PC, or allow some of their primary compute cycles to be used as a virtual machine for this software.

There are tracking technologies other than those using video, but they do not quite match the needs of a large screen-based VR station. For example, inertial devices can calculate relative movements and rotations, but while an HMD-based immersive system can tolerate the inexactness of these devices, the IQ-station cannot. Electromagnetic tracking systems are capable of producing sufficient position tracking, but are somewhat higher in cost for a comparable tracking volume. For a complete discussion on available tracking technologies, an introductory textbook on VR is recommended [1].

In the 2007/2008 time frame, commercial television manufacturers, in particular Samsung and Mitsubishi, began releasing models featuring stereoscopic output using Digital Light Processing™ (DLP®) projection technology. These systems functioned well, though were 16-24" deep to accommodate their projection nature. By the outset of 2010, it was quickly evident that 3D-TV technology had truly come of age. The Consumer Electronics Show in Las Vegas was awash with 3D. This explosion of new models and technologies aimed at the consumer 3D market has been driven by the re-invasion of the 3D theatrical movie. The result is a large selection of 3D-TVs to choose from, with the predictable price decreases and improvements in form factor, such as going from ~18" deep projection to only 3" deep plasma, and 1" LCD and LED devices. As these are not "auto-stereo" displays, stereo-glasses are still required. Fortunately, the cost of these glasses has also fallen from \$700 or more to under \$150 to accommodate the mass market.

The last major component of the system then is the simulation and rendering computer. Depending on the complexity of the virtual world, the cost for this system could feasibly range from \$1.5K to \$10K. A typical system will include a professional level GPU card (such as the nVidia Quadro™), which puts the cost of the computer in the \$5K range. But for graphically simple worlds this might be overkill.

Finally, there is the physical positioning and securing of all the components. This could be as simple as a TV display table or wall-mounting bracket. On the more elaborate end of the scale, one could consider a rolling flat-panel mount, or perhaps two, with a second mount for a display for the tracking



Fig. 2. The IQ-station, a mobile integrated immersive display system. This model uses two PCs, one for tracking and one for rendering.

computer which might also include other inputs, perhaps even screen-touch inputs. Another option is a multi-tiered or even a motorized elevating table, such as available from AnthroTM. Whatever the choice, if one chooses to use video technology for the position tracking, then a means of mounting the camera(s) must also be incorporated. By directly mounting the cameras to the platform, the system is made mobile. Accurate software-hardware calibration is essential for immersive environments and requires a stable physical relationship between the display screen and the tracking system. Table 1 presents the costs of the standard (IU), advanced (INL) and minimal IQ-station systems. All cases use an OptiTrackTM tracking system, however the minimal case uses fewer cameras. IU selected a 67" Samsung DLP, whereas INL opted for the 73" Mitsubishi DLP. On the computing side, INL has included a multi-CPU system with a robustly sized RAM. To accommodate larger audiences, INL also opted to include ten pair of the more robust style of stereoscopic glasses. This is all combined on the multi-tiered AnthroTM table (Figure 2).

2.2 Software Systems

Frequently, the selection of the underlying software will be made in conjunction with the system's hardware. The decision is intertwined because one must choose hardware that will operate with the selected software. Furthermore, the application software will often dictate what middleware libraries and device drivers are required. Here we describe the two VR integration libraries commonly used with the IQ-station. These two libraries are Vrui [12] developed at the University of California, Davis, and FreeVR [13].

As mentioned, our systems use the NaturalPoint OptiTrackTM system. OptiTrack includes a software suite that generates tracking data in common VR

Table 1. Approximate IQ-Station costs in USD

Component	IU	INL	Minimal
Optical tracking system	\$5,000	\$5,000	\$3,000
Optical tracking computer	\$1,000	\$1,000	\$500
3D TV display	\$2,000	\$2,000	\$2,000
3D glasses	\$500	\$3,000	\$500
Visualization computer	\$5,000	\$10,000	\$1,500
Table and mounting hardware	\$1,500	\$4,000	\$1,500
Total	\$15,000	\$25,000	\$9,000

protocols (VRPN and trackdTM), so from the VR library perspective, it receives a standard input stream.

Both the FreeVR and Vrui libraries are full-featured virtual reality integration libraries. That is to say, both handle the interface to the input and output systems, make the perspective rendering calculations, and then allow the end-user application to simulate and render the virtual world. Both required special adaptation to handle the “checkerboard” style of left-right eye renderings. There are pros and cons to each library, but fortunately both can peacefully coexist on the same system. Both work on Unix-style operating systems, including Linux and OS-X. One benefit of Vrui is that it includes a set of 3D widgets that can transition from a desktop interface to an immersive interface. Vrui also includes a specialized interface to the Wiimote. FreeVR allows for the quick porting of many existing VR applications. FreeVR also has a full-featured and formally defined configuration system.

In the end, one only needs to choose one or the other when developing a new application. For deploying an existing application, the choice will have been made by the application developer.

3 Applications

An integrated low-cost hardware scheme, and middleware software are necessary, but not sufficient to be a tool useful for the scientist, engineer, or other end-users. To complete the package, end-user applications must be included. Representative applications include a pre-existing volume visualization tool (“*Toirt Samhlaigh*”), and a world walk-through application suitable for training workers who need to become familiar with real-world physical operations.

3.1 Volume Visualization

Toirt Samhlaigh, our volume visualization tool, is built upon the virtual reality user-interface (Vrui) toolkit. Vrui enables visualization in an immersive environment by providing a collection of classes to facilitate the development of

immersive applications. To meet the unique performance constraints required for implementation in an immersive environment, we utilize an approach to accelerate GPU-based volume visualization, using a heterogeneous data structure [14]. Visualization of large volumes is facilitated through an empty-space leaping data structure traversal using tailored termination criteria.

The resulting application has been used in several imaging-based laboratories. Toirt Samhlaigh has been used successfully in a geology laboratory examining Green River shale for composition in oil-shale using data from an electron microscope; in a microbiology laboratory for examining micro organism ecosystems in a Tunicate from a confocal microscope; and has been used to manipulate medical imaging data from MRI and CT imaging equipment as depicted in Figure 3 (a).

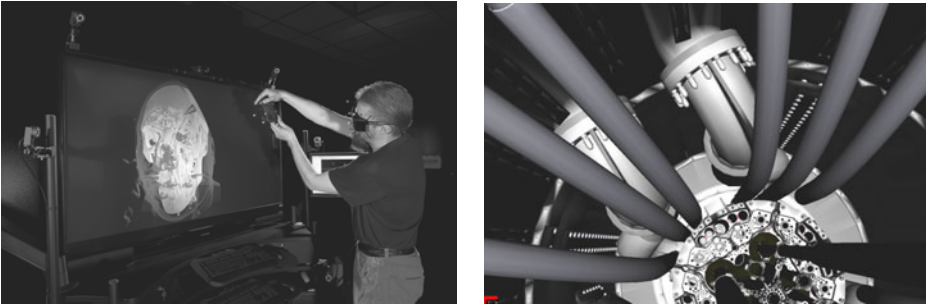


Fig. 3. Left: (a) Here a user manipulates a clipping-plane through computer tomography (CT) data using the Toirt Samhlaigh immersive application. Right: (b) A view looking downward in the ATR reactor vessel. Details shown include: serpentine reactor core, channels and tubes for experiments and large pipes for cooling water.

3.2 Training via World Walk-Through

INL's Advanced Test Reactor (ATR) operates a water cooled, high flux test reactor for scientific purposes. To support engineering, training, and the National Scientific User Facility mission of the ATR, a VR application was developed to use 3D CAD drawings of the facility and present a virtual tour, which is based on Delta3D[15] combined with either Vrui or FreeVR (Figure 3 (b)). The immersive environment provided by the IQ-station, coupled with this application, creates an effective training medium that communicates important technical information much more quickly and accurately than simply reviewing paper drawings or computer images with a typical 2D user interface.

Engineers, maintenance staff, subcontractors, and operators use the virtual ATR application running on an IQ-station for interactive exploration of the ATR components and facility. The combination of a portable immersive workstation and the easy-to-use application, create an ideal environment for discussion of features, design, and operating characteristics without the access and scheduling limitations of visiting the real facility. New staff paired with experienced staff in

front of an IQ-station are able to engage in highly effective knowledge transfer that might not be possible by traditional means. This project has been very successful and generates numerous requests for enhancements and additional immersive environment interactions such as interactive component change-out and reactor physics overlays.

4 Experiences

The value of even a relatively low-cost VR system can only be realized when the end-users find the system sufficiently useful that they are willing to overcome the barriers of an existing workflow and actively exploit it. On the flip-side, without a workable development environment (which includes a support community), the costs of maintaining and upgrading systems with low-cost hardware can outweigh the benefits.

4.1 The User Experience

One benefit of the VR libraries used by our applications is that both were designed to scale between large and small immersive display systems. Thus, users do not have to become familiar with new user interfaces as they transition between an IQ-station or a CAVE, for example. The Vrui-based applications also run reasonably well on traditional desktop workstations (though most users will likely be migrating from other analysis tools). We might consider the range of interface systems to be a “pyramid” ranging from the highly immersive but rare 6-sided CAVE all the way down to the non-immersive but ubiquitous desktop, with the IQ-station a step or two above the desktop. This ability to move up and down the pyramid can be beneficial in both directions — allowing sites with existing high-end immersive systems to spread their wealth and promote their immersive applications to a wider community; and for sites just entering the fray, an IQ-station can be a gentle introduction to the benefits of the physically immersive interface.

4.2 Development Experiences

There are many trade-offs to evaluate in determining a suitable system for one’s user community. By sharing the lessons learned through this process, we help others avoid rediscovering these same trade-offs and more quickly find the right technologies for them.

Hardware Development. Riding the wave of consumer technology certainly can bring us to the point where we can deliver functional immersive systems that improve our ability to work with large collections of data. However, there are hazards when we rely too much on technology that has a short model-life and targets general users rather than the special-purpose needs for immersive displays. Thus, the whims of the consumer-driven marketplace means that manufacturers

of 3D display hardware no longer need to focus on the issues of whether the systems will be more or less usable in a research setting. Whether for ease of use, or locking customers into proprietary environments (for repairs/replacement, or expansion) consumer options often mean less flexibility and in some cases less usability for systems aimed at improving scientific workflow. The flux of the consumer display models suggests that if there is a desire for a fleet of similar units, it is prudent to purchase a sufficient quantity of displays from the outset, whereas the tracking and computing technologies are less susceptible to the whims of the consumer marketplace.

Thus, during this period of great flux, there are many considerations to take into account when choosing a display technology. The DLP technology uses projection systems, resulting in deeper units that cannot be tilted. The positive side of DLP systems is that they are available in larger sizes (up to 82") and suffer less from stereo-separation interference (ghosting) and burn-in. LCD and Plasma systems can be very thin, opening up a variety of mounting options, including drafting or table-top styles, and are also somewhat more mobile. However LCD and Plasma systems are limited in size (up to 62") and do suffer from ghosting, and more noticeable burn-in effects. Unlike computer monitors, the modern HDTV specifications (1.4) require that the TV itself generate the stereo synchronization signal. The result of this requirement is that it becomes impossible to synchronize multiple screens together. Whether there is a problem with vendor lock-in is on a model by model basis. Overall, the DLP solutions are still the best options, but at least one manufacturer (Samsung) has already discontinued their line of DLP displays. A third option is the "prosumer" stereoscopic display from JVC (model GD463D10) which provides stereo separation through polarizing filter technologies (dubbed Xpol®). The benefits come from requiring only inexpensive glasses, and not suffering from the multi-screen synchronization problem. The problem with the display is that 46" is the only size available, and thus larger displays require the extra complications of tiling[16].

Software Development. Benefits of using highly-customized software include the ability to adapt for new hardware, such as when 3D-TVs and "Wiimotes" were hitting the marketplace, plus the ability to adapt to requests that arise from the user community. On the other hand, tools developed for a wide audience (i.e. "generic tools") benefit from their ability to be more widely deployed and therefore generate a larger user base. If a community is supported, then generic tools can become self-supporting through a user community that helps it's own.

The questions are: is there a middle ground, and does aiming at the middle ground reduce the potential user base? Perhaps some of the generic tools can be adopted to work with off-the-shelf components. This possibility is increased when the hardware vendors choose to make use of existing standards. For example, many of the early 3D-TVs worked with existing stereoscopic glasses and emitters. Also, some OpenGL drivers (such as the nVidia Quadro™) were adapted to convert the traditional left and right rendering buffers into a checkerboard stereo format. Both of these aspects eased the development process of the IQ-station.

Another example includes the use of the existing VRPN and trackdTM position tracking protocols by the OptiTrackTM rigid body tracking suite.

Application Development. Software plays an important role in the success of an IQ-station. The most significant predictor of project success with the IQ-station is the quality of the applications that are developed to utilize the immersive display.

Understanding user requirements and proper implementation of immersive user interfaces should be the priority of any organization considering deployment of an IQ-station. Of course, if the purpose of the IQ-station is to jump-start IE development then this system is an ideal place to start.

5 Conclusion

A new plateau has been reached in the realm of immersive interface technologies by taking advantage of recent product developments to produce an integrated system that performs as a useful tool, not merely a research prototype. Beyond integrating disparate hardware components, existing immersive software tools were adjusted to ensure their usability on these smaller-scale immersive systems. Ultimately, the achievement that is important is that there is now a fully functional immersive system that can be deployed where the research is taking place. By moving the equipment out of the computer science research lab and into the domain science labs, domain researchers have begun to fully incorporate this technology into their day-to-day operations.

Not all computing needs are met by portable smart phones and not all applications require a supercomputer. A desktop computer is a good compromise that serves the needs of a large segment of computer users. In a similar manner, the IQ-station has the potential to serve the needs of a large segment of users. As immersive environments grow in popularity and expand into everyday activities, a solution such as the one presented in this paper, effectively fills the need for many users. As a laboratory instrument, an architecture evaluation system, or as a scientific discovery tool, the use of these systems will quickly expand and support the next generation of immersive environment applications.

Acknowledgements. The authors would like to thank the Desert Research Institute (DRI), the Center for Advanced Energy Studies (CAES), the Idaho National Laboratory (INL), the Pervasive Technology Institute (PTI) and Indiana University (IU) for facilities and support. We would also like to acknowledge the system development work of Keith Wilson, of the Idaho National Laboratory. This work was supported partially through the INL Laboratory Directed Research & Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517, RDECOM-STTC under Contract No. N61339-04-C-0072, and by Kelly Estes, Paul Henslee, and Julie Huntsman of the Advanced Test Reactor Life Extension Program, Idaho National Laboratory.

References

1. Sherman, W., Craig, A.: *Understanding Virtual Reality*. Morgan Kaufmann Publishers, San Francisco (2003)
2. Fenn, J., Raskino, M.: *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time*. Harvard Business School, Boston (2008)
3. Prabhat, F.A., Katzourin, M., Wharton, K., Slater, M.: A Comparative Study of Desktop, Fishtank, and Cave Systems for the Exploration of Volume Rendered Confocal Data Sets. *IEEE Transactions on Visualization and Computer Graphics* 14, 551–563 (2008)
4. Bohrer, G., Longo, M., Zielinski, D., Brady, R.: VR Visualisation as an Interdisciplinary Collaborative Data Exploration Tool for Large Eddy Simulations of Biosphere-Atmosphere Interactions. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) *ISVC 2008, Part I. LNCS*, vol. 5358, pp. 856–866. Springer, Heidelberg (2008)
5. Brady, R., Pixton, J., Baxter, G., Moran, P., Potter, C., Carragher, B., Belmont, A.: Crumbs: a virtual environment tracking tool for biological imaging. *Biomedical Visualization* 82, 18–25 (1995)
6. Johnson, C., Moorhead, R., Munzner, T., Pfister, H., Rheingans, P., Yoo, T. (eds.): *NIH/NSF Visualization Research Challenges Report*. IEEE Press, Los Alamitos (2006)
7. Sherman, W.R., O’Leary, P., Kreylos, O., Brady, R.: *IEEE Visualization 2008 Conference Workshop on Scientific Workflow with Immersive Interfaces for Visualization*. In: Sherman, W.R., O’Leary, P., Kreylos, O., Brady, R. (eds.) *Proceedings of the IEEE Visualization 2008 Conference*. IEEE Press, Columbus, OH (2008)
8. Johnson, A., Leigh, J., Morin, P., Van Keken, P.: *GeoWall: Stereoscopic Visualization for Geoscience Research and Education*. *IEEE Computer Graphics and Applications* 26, 10–14 (2006)
9. Lee, J.: *Head Tracking for Desktop VR Displays using the WiiRemote* (2007), <http://www.youtube.com/watch?v=Jd3-eiid-Uw>
10. Kato, H., Billinghurst, M.: *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System*. In: *IWAR 1999: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, Washington, DC, USA, p. 85. IEEE Computer Society, Los Alamitos (1999)
11. Kato, H.: *ARToolKit* (1999), <http://www.hitl.washington.edu/artoolkit>
12. Kreylos, O.: *Environment-Independent VR Development*. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) *ISVC 2008, Part I. LNCS*, vol. 5358, pp. 901–912. Springer, Heidelberg (2008)
13. Sherman, W.: *Commodity-Based Projection VR: Software for Virtual Reality*. In: *SIGGRAPH 2004: ACM SIGGRAPH 2004 Course Notes*. ACM, New York (2004), <http://freevr.org/>
14. O’Leary, P., Sherman, W., Murray, A., Riesenfeld, C., Peng, V.: *Enabling Scientific Workflows Using Immersive Microbiology*. In: Sherman, W., O’Leary, P., Kreylos, O., Brady, R. (eds.) *Proceedings of the IEEE Visualization 2008 Conference*. IEEE Press, Columbus (2008), DVD
15. Darken, R., McDowell, P., Johnson, E.: *Projects in VR: The Delta3D open source game engine*. *IEEE Computer Graphics and Applications* 25, 10–12 (2005)
16. DeFanti, T., et al.: *The Future of the CAVE*. *Central European Journal of Engineering* (to appear, 2010)

A Fiducial-Based Tangible User Interface for White Matter Tractography

Steven R. Gomez, Radu Jianu, and David H. Laidlaw

Department of Computer Science
Brown University
{steveg,jr,dhl}@cs.brown.edu

Abstract. We present a method for interacting with 3D brain tract visualizations using a webcam and a fiducial marker that can be constructed cheaply in any home or office. Our contributions are a fiducial-based tracking architecture in the context of white matter tractography, and a preliminary evaluation with domain scientists providing usability and design insights. Expert feedback indicates that model positioning in our system is easier than in previous methods using traditional input devices or two-dimensional input interfaces, and that tract selection may be faster to execute using our tool, given training and practice.

1 Introduction

Scientists can now explore the shape and connectivity of fibrous tissues, such as muscle and brain white matter, through visualizations of Diffusion Tensor Magnetic Resonance Imaging (DTI) data that commonly render variations of streamlines, such as streamtubes and hyperstreamlines, in 3D. In the case of the brain, these streamline models are visually dense as a consequence of the brain's complex circuitry. As a result, typical interactions with white matter tracts, such as bundle selection or inspection of the model, may be difficult for a user to perform with conventional interface tools. In this paper, we present a new method for interacting with neural fiber tracts using a computer vision-based interface that allows for intuitive manipulation of the DTI model.

We use fiducial tracking to position the brain and perform 3D selection of fiber tracts. In lieu of a typical interface, e.g. keyboard and mouse, or specialized input devices, the user holds and moves a homemade marker object in front of a webcam to manipulate the model. This marker can be constructed inexpensively from a pattern and common household materials. In our experiments, we constructed a cardboard cube and decahedron whose faces are covered with paper Augmented Reality (AR) patterns, which can be printed on any printer.

Figure 1 shows a user interacting with a DTI brain model using our system. We have obtained feedback from experts in an anecdotal study for an initial prototype. Results suggest that this type of lightweight 3D interaction has the potential to enable faster interaction with dense fiber tract collections. Expert feedback indicates that our new representation is more intuitive – and may be easier to use and learn – than conventional DTI model interaction methods.

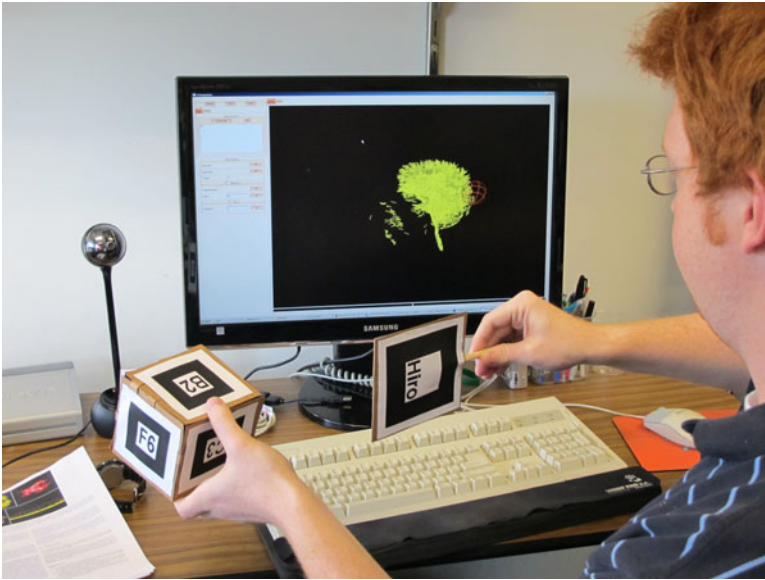


Fig. 1. The user holds both the brain marker and selection marker to identify tracts of interest in the visualization

2 Related Work

2.1 Interacting with DTI Models

White matter tracts in the brain are frequently visualized as tubes rendered from streamline traces in DTI data [1,2,3,4]. Common interaction tasks for these visualizations include exploration of the 3D model and tract of interest (TOI) selections in the tract bundles that make up the model. Tract selections are traditionally performed directly in 3D: regions of interest (ROIs) are placed in the volume along the presumed path of the desired tract bundle; the application then selects only fibers that intersect those ROIs [5,6,7]. At the same time, mastering the interactions required for 3D ROI manipulation using conventional input devices, e.g. mouse and keyboard, often slows inexperienced scientists in their exploration.

Due to these limitations, recent work has explored new input techniques that allow brain scientists to perform tract bundle selection with minimal overhead. In CINCH [8], Akers presents a sketch and gesture interface for identifying tract bundles by curve similarity and other heuristics. Jianu et al. [4] and Zhang et al. [3] present methods that use tract similarity to create 2D abstractions of the DTI space along the original 3D model, and use brushing and linking to map interactions performed in 2D onto the 3D model. While simplifying interaction, these methods require scientists to operate on abstract, unfamiliar representations and maintain a mapping between the different views of the same

data. Closer to our work is a technique introduced by Zhou et al. [9] for lassoing fiber tracts into bundles using a finger-mounted haptics device that provides force feedback and 3D tracking. Recently, Klein et al. [10] created a tract selection interface that allows a user to create 3D ROIs using a Nintendo Wii remote controller. Our system, which uses fiducial marker tracking with the ARToolKit [11] and a simple webcam, differs from these by offering an interface for the brain model that allows the user to interact in 3D space, as opposed to planar sketching, but requires no specialized hardware.

2.2 Fiducial Tracking

Fiducial-based tracking has grown in popularity with the release of systems like ARToolKit and ARTag [12,13], which provide pattern creation and recognition as library functions. Fiducials have been broadly applied in applications ranging from robot navigation and communication [14], games and learning [15], and the design of tangible user interfaces (TUIs) [16] where real objects are used as input devices. We find a novel application area for TUI design principles in interactive scientific visualizations.

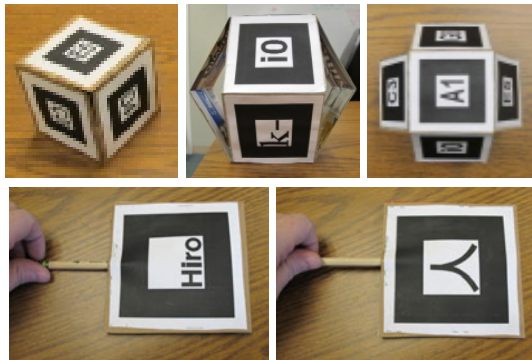


Fig. 2. Brain model markers constructed as polyhedra from cardboard and augmented reality (AR) patterns on the flat faces. Top, left to right: cube model; side view of decahedron; top view of decahedron. Bottom, faces A and B of the two-sided selection marker.

3 Methods

3.1 Brain Model Manipulation

We use the ARToolKit [11] for tracking fiducial patterns in the video feed from the webcam. Each pattern is a monochrome character sequence against a white background and inside a black square frame. We choose each sequence to be asymmetric and unique among the other patterns to avoid as much pattern

ambiguity in the detection phase as possible. The specification for each fiducial pattern we use is created from a toolkit utility and stored as a bitmap of pixel brightness intensities. During the main detection loop in our visualization software, the toolkit provides the orientation of any detected objects and a match-confidence score for each specific pattern we use. The detected object is labeled by the pattern that scores the highest confidence value.

We use the orientation of detected patterns to compute a coordinate frame that transforms the 3D model consistently with the user's manipulation of the tangible model. For a detected pattern in the frame, ARToolkit returns orientation parameters that determine the location and surface normal of that face. By the marker's design, we know where each pattern is oriented in object space in the prototypical marker shape. The rotations that move each specific face into a common position are pre-computed and stored in a hash table for quick lookup. As a result, we transform the model simply by composing the coordinate frame returned by the toolkit with the object space rotations stored for the respective pattern.

Because multiple patterns may be detected simultaneously from the surface of the cube or decahedron markers, the coordinate frame most similar to the existing model transformation is chosen. This similarity is determined as the arc length produced by projecting a common unit vector into both the current and detected coordinate frames. We choose the pattern that produces the minimum arc distance between these transformed points on the unit sphere. This ensures that the same detected face is used as the reference coordinate frame until it is no longer detected in the camera's view. This continuity check is aimed at keeping the 3D model as stable as possible during manipulation. As an added measure to mitigate camera jitter, we maintain a short history of detected marker orientations (up to 10 frames) and apply an average of these coordinate frames to smoothly transform the model over time.

3.2 Selecting Tracts

The selection tool we constructed is a two-sided marker with one unique pattern on each face and one handle that can be twisted to flip the marker to the reverse sign. Detecting the selection tool is similar to detecting the brain markers, but its orientation is used for positioning an on-screen selection sphere rather than moving the brain model.

When the A-side is visible to the camera, as the user moves the selection sphere through the white matter model, tracts intersecting the sphere are highlighted as a preliminary selection. Once the user is satisfied with the selection he can twist the tool in place so that the reverse B-side is detected. Toggling this face triggers the "selection" action, which commits the preliminary selection. Figure 3 illustrates this process; preliminary selections are marked in blue, committed ones in red. Three selection modes are possible: add tracts, remove tracts and intersect. The selection mode is used to construct preliminary selections by composing the set of tracts intersecting the selection sphere at each moment in time with the current selection. An existing selection can thus be cleaned by

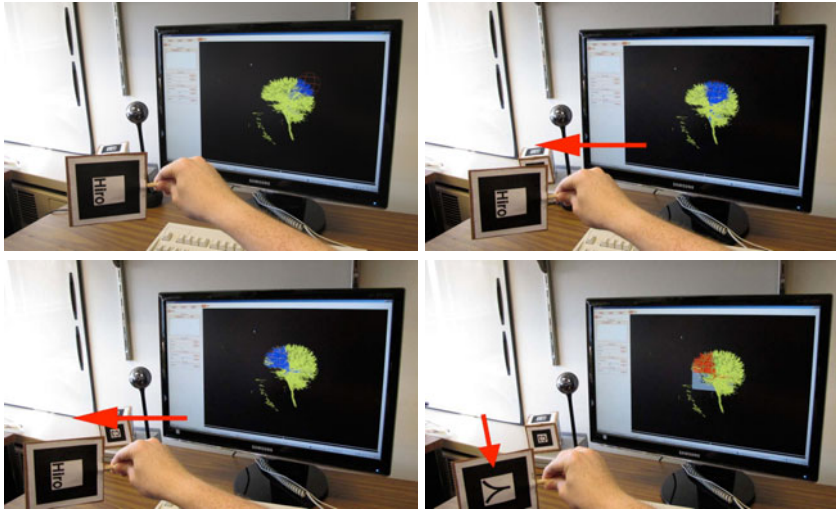


Fig. 3. Selecting tracts of interest in the brain model. The sequence shows the user positioning the selection sphere in model space by moving the physical marker, then flipping the marker to add the highlighted tracts to the selection. The marker can then be removed from the camera’s field of view and the selection persists. In the images above, the red arrows indicate movement from the previous frame.

brushing over tracts while the “remove tract” mode is active. The “intersection” mode provides a logical AND operation for selecting tracts crossing multiple areas of interest in the model.

3.3 Gesture Recognition

We designed our application to support simultaneous movement of the brain model marker and the selection marker. We hypothesized that using a mouse and keyboard for selection interactions (e.g. changing ROI radius or selection mode) might interrupt the user’s workflow, causing longer completion times for selection tasks. To overcome this limitation, we developed a set of gestures performed with the markers to execute commonly performed interactions in the system.

For flexibility, the interface also supports selection controls using traditional keyboard and mouse input, accommodating users who prefer to use only the brain model marker with the camera. When either marker leaves the field of view of the camera, interaction events are not generated by that marker. The user can effectively pause interactions by removing or obstructing a given marker, and can resume by reintroducing it into the scene.

Committing a selection can be performed, as previous described, by flipping the selection marker to show the opposing face. Toggling between selection modes (i.e. adding to, removing from and intersecting with the current selection) is performed by rotating the marker 90 degrees about its center in the image plane

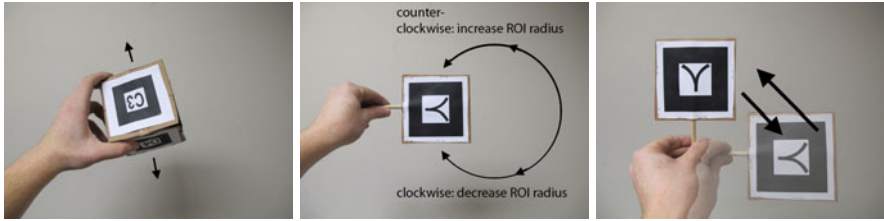


Fig. 4. Examples of gestures using the brain and selection markers. Left to right: shaking the brain marker slightly will toggle the 3D model from being frozen in position or unfrozen; creating a circular motion with the selection marker will increase (counter-clockwise) or decrease (clockwise) the radius of the selection sphere; rotating the selection marker 90 degrees and returning it back to its start position will switch selection modes between “add”, “subtract”, or “intersection”.

and then returning it to its initial orientation. The spherical selection ROI’s radius can be increased or decreased by moving the marker along a circular path clockwise or counter-clockwise. Finally, the 3D model can be frozen in place, and unfrozen, by moving the brain marker into the desired position and shaking it slightly. These gestures are illustrated in Figures 3 and 4.

Gestures are recognized by tracking a marker’s position and orientation over short periods of time – 1-2 seconds in our implementation – and computing gesture likelihood scores for the marker orientation data collected. In the case of shaking and circular movements, the scores incorporate distance traveled by the marker, displacement of the marker, and directionality of movement. A least-squares distance error function is used to assess whether a marker’s path is circular during the gesture.

3.4 DTI Model

Our test data included a diffusion-weighted MRI volume of a normal volunteer’s brain, acquired on a 1.5T Siemens Symphony scanner with the following acquisition parameters in 12 bipolar diffusion encoding gradient directions: thickness = 1.7mm, FOV = 21.7cm x 21.7cm, TR = 7200ms, TE = 156ms, b = 1000, and NEX = 3. The DTI volume was then obtained by fitting 6 independent parameters of a single second-order tensor at each voxel to the 12 measurements from diffusion-weighted MRI volume [17].

4 Results

We evaluated our prototype anecdotally with two neuropsychologists who had experience using white matter streamtube visualizations in clinical research. Each received a demonstration and brief training on the interface and gestures before exploring the tool and selecting TOIs. We gathered feedback using a think-aloud protocol during our demonstration and the users’ exploration of the tool.

The workflow that both our subjects suggested, and said they were likely to use with such an application, was to position the model in a specific pose, freeze it, and use the selection marker to perform a selection. They found that for certain types of interactions, such as model positioning using the fiducial marker, our method would be helpful in supplementing the mouse and keyboard. Both agreed that different users may have varying preferences for using the fiducials or a typical keyboard/mouse interface, for either coarse model manipulation or fine-scale TOI selection. For instance, one expert said he preferred the fiducial for coarse-scale rotation of the model, then favored switching to the mouse for precise tract selection; the other felt he could set the model position more accurately with a keyboard and mouse, but preferred the fiducial for tract selection because he could quickly select tract bundles in 3D and preview the selection.

Our subjects stated that while the selection gesture is useful, they preferred using the keyboard and mouse to alter other selection parameters (e.g. adding, removing, or intersecting modes; freezing and unfreezing the model). They noted that the selection marker was at times difficult to use with precision, given that executing the twist “select” gesture can change the marker’s position, causing different tracts to be selected once the gesture is recognized. One noted that this became easier after a few minutes of practice, and that more extensive initial training may give more confidence to the user. Both agreed that adding a new mode to hide non-selected tracts would allow the user to refine the selection more easily by reducing visual clutter. Additionally, one expert suggested that other medical applications requiring quick and easy inspection of a 3D model, such as a heart or tumor visualization, could benefit from an interaction method like ours.

5 Discussion

5.1 System Robustness

One finding in our work is that a vision-based approach to model manipulation introduces challenges from external factors. Fiducial recognition relies on some level of light and camera quality beyond our control. Occlusion of the fiducial patterns can also cause recognition problems, especially when concurrently manipulating both the selection marker and brain maker. One marker may block the camera’s view of the other as they try to occupy an overlapping position in model space. Even a user’s grip on a marker can introduce occlusion problems. In fact, we noticed that some improved usability of the decahedron over the cube was not due to the visibility of multiple patterns at the same time, as initially intended, but instead by the ease of holding and moving it while having at least one marker visible at all times.

5.2 Accommodating User Workflow

In the selection task, the largest challenge we identified was designing interactions that were appropriate for the user’s intended level of precision for selection.

For instance, a large ROI volume that moves relatively quickly in model space when manipulated may be ideal for a coarse tract bundle selection; however, for refinements of the selection, a smaller, slower moving selection volume may be easier for the user to navigate and use without error. Ideally, we would like our system to determine the user's tract selection goals in order to set these parameters automatically, to reduce manual specification that may slow the user down. We hypothesize that a coarseness estimate can be made from the level of model zoom in the 3D visualization; a user who is closely examining the model likely wants to make finer scale selections.

As revealed by our evaluation, the workflow suggested by the users was to position the model, freeze it, and then make a selection. We believe this is due to our subjects' familiarity with streamtube visualizations where model positioning and selection cannot happen concurrently. We hypothesize that new users, or those with training, will prefer to perform selections while manipulating both the selection marker and the model marker concurrently. We expect this would decrease the time required to execute a selection with minimal error because positioning the brain and selection markers can be done simultaneously. This may require further extension and refinement of the gesture set to obviate all keyboard and mouse interactions.

5.3 Reaching a Broad Audience

The power of this approach lies in its affordability. Many laptops are now sold with integrated cameras, and for other personal computers, external webcams are relatively inexpensive to purchase. In the released version of our application, we will make available marker templates that can be printed, cut, and folded into cubes. Users should be able to reproduce our setup in less than half an hour from downloading our tool.

Furthermore, we plan to design meaningful pictures, such as projections of typical brain views (i.e. coronal, sagittal, axial), to distribute as fiducial patterns on the marker. Each of these images offers a preview of the model's orientation before the marker face is presented to camera. We believe these fiducials will help new users more intuitively understand how manipulation of the marker object changes the brain visualization.

6 Conclusion

We present a lightweight, affordable interaction method for DTI brain visualizations using paper fiducial markers that are easy to create. This method allows scientists to manipulate 3D white matter models directly by manipulating a physical analogue in real space in front of a camera. Our contributions include a fiducial-based tracking architecture for interacting with white matter tractography models, and an evaluation revealing the advantages of this method and suggesting design guidelines. Furthermore, by using a physical marker that can

be printed, constructed easily, and used without specialized hardware, we provide a simple method for distributing the visualization tool to a wide base of users.

References

1. Basser, P., Pajevic, S., Pierpaoli, C., Duda, J., Aldroubi, A.: In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine* 44, 625–632 (2000)
2. Mori, S., Van Zijl, P.: Fiber tracking: principles and strategies—a technical review. *NMR in Biomedicine* 15, 468–480 (2002)
3. Zhang, S., Demiralp, C., Laidlaw, D.H.: Visualizing diffusion tensor mr images using streamtubes and streamsurfaces. *IEEE Transaction on Visualization and Computer Graphics* 9, 454–462 (2003)
4. Jianu, R., Demiralp, C., Laidlaw, D.: Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics* 15, 1449–1456 (2009)
5. Catani, M., Howard, R.J., Pajevic, S., Jones, D.K.: Virtual in vivo interactive dissection of white matter fasciculi in the human brain. *NeuroImage* 17, 77–94 (2002)
6. Wakana, S., Jiang, H., Poetscher, N.L.M., van Zijl, P.C., Mori, S.: Fiber tract-based atlas of human white matter anatomy. *Radiology* 230, 77–87 (2004)
7. Maddah, M., Mewes, A.U.J., Haker, S., Eric, W., Grimson, L., Warfield, S.K.: Automated atlas-based clustering of white. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3749, pp. 188–195. Springer, Heidelberg (2005)
8. Akers, D.: Cinch: a cooperatively designed marking interface for 3d pathway selection. In: *UIST 2006: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pp. 33–42. ACM, New York (2006)
9. Zhou, W., Correia, S., Laidlaw, D.H.: Haptics-assisted 3D lasso drawing for tracts-of-interest selection in DTI visualization. *IEEE Visualization 2008 Poster Compendium (Best Poster Nominee)* (2008)
10. Klein, J., Scholl, M., Kohn, A., Hahn, H.K.: Real-time fiber selection using the wii remote. In: *Proceedings of the SPIE*, vol. 7625 (2010)
11. (ARToolkit), <http://www.hitl.washington.edu/artoolkit/>
12. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 590–596 (2005)
13. Fiala, M.: Designing highly reliable fiducial markers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 1317–1324 (2010)
14. Dudek, G., Sattar, J., Xu, A.: A visual language for robot control and programming: A human-interface study. In: *ICRA*, pp. 2507–2513 (2007)
15. Kostandov, M., Schwertfeger, J., Jenkins, O.C., Jianu, R., Buller, M., Hartmann, D., Loper, M., Tsoli, A., Vondrak, M., Zhou, W., Fiala, M.: Robot gaming and learning using augmented reality. In: *SIGGRAPH 2007: ACM SIGGRAPH 2007 Posters*, p. 5. ACM, New York (2007)
16. Kato, H., Billinghamurst, M., Poupyrev, I., Imamoto, K., Tachibana, K.: Virtual object manipulation on a table-top ar environment. In: *International Symposium on Augmented Reality*, p. 111 (2000)
17. Basser, P.J., Mattiello, J., LeBihan, D.: Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance. Series B* 103, 247–254 (1994)

Immersive Molecular Visualization and Interactive Modeling with Commodity Hardware

John E. Stone¹, Axel Kohlmeyer², Kirby L. Vandivort¹, and Klaus Schulten³

¹ Beckman Institute for Advanced Science and Technology,
University of Illinois at Urbana-Champaign

² Center for Molecular Modeling, Temple University

³ Department of Physics, University of Illinois at Urbana-Champaign

Abstract. Continuing advances in development of multi-core CPUs, GPUs, and low-cost six-degree-of-freedom virtual reality input devices have created an unprecedented opportunity for broader use of interactive molecular modeling and immersive visualization of large molecular complexes. We describe the design and implementation of VMD, a popular molecular visualization and modeling tool that supports both desktop and immersive virtual reality environments, and includes support for a variety of multi-modal user interaction mechanisms. A number of unique challenges arise in supporting immersive visualization and advanced input devices within software that is used by a broad community of scientists that often have little background in the use or administration of these technologies. We share our experiences in supporting VMD on existing and upcoming low-cost virtual reality hardware platforms, and we give our perspective on how these technologies can be improved and employed to enable next-generation interactive molecular simulation tools for broader use by the molecular modeling community.

1 Introduction

Over the past decade, advances in microprocessor architecture have led to tremendous performance and capability increases for multi-core CPUs and graphics processing units (GPUs), enabling high performance immersive molecular visualization and interactive molecular modeling on commodity hardware. Stereoscopic visualization has been used in molecular visualization for decades, but it was previously a prohibitively costly technology. A growing number of commodity GPUs, televisions, and projectors now support stereoscopic display, enabling molecular scientists to afford stereoscopic display hardware for individual use for the first time. Inexpensive but powerful embedded microprocessors have also enabled a new generation of low-cost six-degree-of-freedom (6DOF) and haptic input devices suitable for molecular visualization in both desktop and immersive visualization environments. These immersive VR displays, 6DOF input devices, and powerful rendering and computing capabilities previously available only to well-funded laboratories are creating a new opportunity to greatly expand the use of immersive visualization and interactive simulation in molecular modeling.

Although the molecular modeling community has long had an interest in the use of immersive visualization and advanced input devices [1,2,3,4,5,6], these technologies have not seen much adoption outside of institutions with significant local VR infrastructure and expertise. One of the challenges that must be overcome in making VR technologies available to application scientists is to make them easy to configure and intuitive to use. Most users have minimal experience with installing and configuring visualization clusters, stereoscopic projection systems, or complex input devices; this presents a barrier to adoption of such advanced technologies, irrespective of cost.

This paper describes the software infrastructure supporting commodity virtual reality hardware and software within VMD [3], a full-featured software package for molecular visualization, analysis, and interactive molecular dynamics, used by tens of thousands of researchers worldwide. We describe our experiences developing and supporting VMD in both desktop and immersive virtual reality environments, and the unique challenges involved in supporting sophisticated immersive display and input technologies in software that is used by a broad community of life scientists that have little or no background in the technical disciplines related to installing and managing complex VR systems.

2 Software Architecture

Among similar molecular visualization tools, VMD is somewhat unique in that it was originally designed (under the initial name “VRChem”) for use primarily in immersive virtual environments such as the CAVE [7], ImmersaDesk [8], and non-immersive stereoscopic projection environments, but subsequently became a widely used desktop application supporting commodity personal computers. Today, VMD continues to support both desktop and immersive virtual environments, and has added expanded support for a wide variety of advanced input devices for use in both scenarios. Current releases of VMD support multiple VR toolkits including CAVElib [7], FreeVR [9], and VRPN [10]. The VMD user community has, at various times, also modified VMD to support VR Juggler [11], and a number of custom in-house VR systems and multi-modal input devices of various kinds [15,12]. Below we describe the design constraints and the hardware and software abstractions that VMD uses to support these diverse toolkits and their respective APIs, while maintaining ease of use for molecular scientists.

2.1 Design Goals and Constraints

One of the major problems that broadly deployed scientific applications must solve is to abstract the differences in hardware and software environments over a potentially wide range of usage scenarios. The bulk of daily molecular modeling work is done on commodity laptop and desktop computers that have no immersive display capability nor advanced input devices, so this usage scenario must be well-supported and must not be compromised in the effort to add support for immersive displays and interfaces.

The typical end users of molecular modeling applications today expect applications to be provided as binary or “shrink-wrapped” applications, just as most business and productivity applications are typically distributed. Similarly, as the separation between a tool or feature’s intended purpose and its technical realization has grown, some level of software automation is typically expected, particularly for common, repetitive, and well understood tasks. Such automation enables the user to focus on the science of their project rather than the technologies they are employing along the way. Our experience confirms that users are often averse or even unable to compile complex applications from source, and that they prefer not to go through complex installation and configuration processes and therefore are willing sacrifice some degree of sophistication, flexibility, and performance in exchange for convenience and ease of use. This observation places some constraints on the design of molecular modeling applications that are intended for broad use, and it gives guidance for designers of VR toolkits that support such broadly used scientific applications. The design of VMD supports the use of low-cost 6DOF VR input devices and stereoscopic display within an otherwise typical desktop application, as well as immersive virtual environments, enabling molecular scientists to incrementally evaluate and adopt VR technologies according their budget and needs.

2.2 Display Management and Rendering

Due to the diversity of display environments that VMD must support, the display management and rendering infrastructure within the program is designed to be flexible and extensible. Since the low-level rendering API, and windowing system or VR APIs for managing the display differ significantly, VMD uses a hierarchy of inherited classes to abstract the differences.

In molecular modeling, traditional scene graphs often incur significant performance penalties either during display redraws or during trajectory animation updates due to the fine-grained nature of the independent atomic motions and transformations that occur. VMD employs a full-custom scene graph structure that is designed specifically for the needs of molecular visualization, and in particular for visualization of dynamics of multi-million atom molecular complexes. VMD’s custom scene graph is also very memory efficient, which is particularly important when rendering structures with up to 100 million atoms. A consequence of the use of a custom scene graph is that VMD must also implement its own code for rendering the scene graph.

VMD implements a `DisplayDevice` base class to abstract the details of windowing systems, VR toolkits, rendering APIs, and hardware platforms. This base class underlies subclasses for both interactive displays and for batch-mode photorealistic ray tracing, scene export, and external rendering of various types. In order to support multiple interactive graphics APIs, the `DisplayDevice` class is subclassed according to the rendering API used, as illustrated in Fig 1. Although OpenGL is the only fully supported rendering API in current versions of VMD, this structure enabled earlier versions of VMD to support IRIS GL and Direct X rendering APIs as well. As we begin to transition from present-day OpenGL 1.1

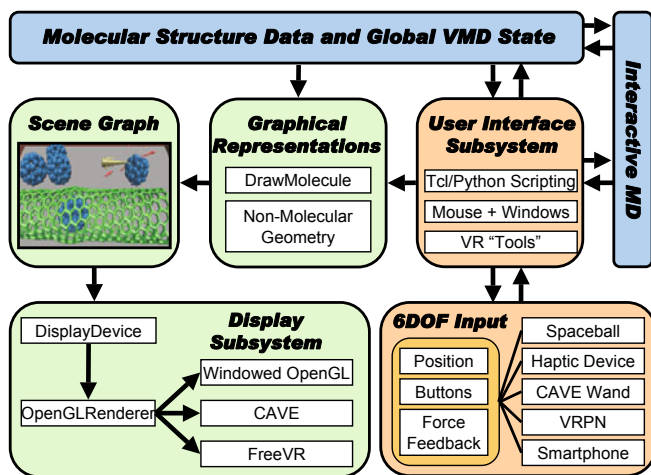


Fig. 1. VMD software subsystems responsible for managing input devices, user interactions, and interactive rendering on desktop and immersive displays

and 2.x rendering functionality towards OpenGL 4.x, the rendering API subclass capability will again prove to be beneficial.

The abstraction of windowing system and VR toolkit APIs is handled by further subclassing one of the renderers. Since the windowing system and VR APIs are largely independent of the underlying renderer code, compile-time macros determine which renderer class is used as the parent class for each windowing system, GUI toolkit, and VR API subclass. This design allowed VMD to support both IRIS GL and OpenGL within the same X-Windows and CAVE `DisplayDevice` subclass implementations. The rendering and display management classes require close handshaking during initial display surface creation, when the VR toolkit or windowing system requests features that must be specified during display initialization, such as a stereoscopic-capable visual context.

Beyond the abstraction of basic display and window management operations, the `DisplayDevice` class hierarchy also manages the details involved with parallel and multi-pipe rendering. The subclasses for each VR toolkit such as `CAVELib` and `FreeVR` provide the necessary data sharing or communication infrastructure along with the mutual exclusion or synchronization primitives required to share the molecular scene graph among each of the renderer processes or threads associated with the VR display system.

Unlike many VR applications, VMD contains a significant amount of internal program state aside from the scene graph, such that it is impractical to attempt to completely replicate and synchronize the full program state among rendering slaves in cluster-based VR systems. This is due in part to the fact that VMD includes internal scripting engines and a multiplicity of other interfaces that can modify internal state, and it would be prohibitive to attempt to maintain coherency of all such interfaces among rendering slaves. For this reason, the

multi-pipe VR implementations of VMD have largely been limited to one of two strategies: a tightly-coupled implementation that stores the scene graph and associated data structures in shared memory arena accessible to both the master application process and the rendering slaves, or a very loosely-coupled implementation based on a set of completely independent VMD processes running on a cluster of rendering nodes communicating over a network. The tightly-coupled shared memory approach is used by the CAVElib and FreeVR display class implementations in VMD, and allows full program flexibility and functionality. The loosely-coupled approach is typically used for ad-hoc display walls where a limited range of interactions are required and no on-the-fly scripting is used.

2.3 Multi-modal Input and Haptic Feedback

VMD abstracts VR input devices through a set of classes that allow 6DOF motion control input, buttons, and haptic feedback. As with the display and rendering classes, the base classes are subclassed for each device or VR toolkit API that is added. Many devices report events through the windowing system, some through VR toolkit API calls, and others through direct communication with low level operating system I/O calls. Since a given VMD session may use devices provided by disparate interfaces and coordinate systems, VMD provides a simple mechanism for applying an arbitrary transformation matrix to the position and orientation information associated with 6DOF motion control devices, enabling the users to customize the mapping from the physical workspace of the device to the virtual workspace. This enables devices to be used seamlessly in the environment, whether they are provided by CAVElib, FreeVR, VRPN, the host OS windowing system, or other sources.

VMD supports low-cost commodity input devices such as joysticks, Spaceball, and SpaceNavigator 6DOF input devices via input subclasses that handle windowing system events or direct device I/O. Devices supporting windowing system events can be used immediately for basic 6DOF motion control with no further configuration, making them an ideal choice for molecular scientists that have neither the time nor inclination to use the more sophisticated VR “tool” controls described below. Most VR-oriented 6DOF input and haptic feedback devices are supported through VR toolkits or libraries such as VRPN. VRPN [10] is an ideal counterpart to VMD for several reasons. VMD can be statically linked against a VRPN client library, which has few or no additional dependencies. In order to use VRPN-supported input devices with VMD, a VRPN server daemon runs on the host with the attached devices, and VMD connects to it over the network. This structure enables new devices to be supported without recompiling VMD binaries, it separates low-level input device management from the main VMD event loop (this work is done inside the VRPN server daemon), and it provides a simple mechanism for access to input devices that are not managed by the host windowing system. The use of VRPN in VMD also makes it easy to use devices that only have drivers for a single operating system with a VMD client instance running on any operating system [13].

In order to use VR input devices within VMD, a mapping is created between a physical device or devices, and an instance of a VMD “tool”. Tools are user interface objects that performs actions on molecules based on 6DOF motion control and button state inputs. This design is similar to the tool abstraction provided in recent VR toolkits such as Vrui [14], except that the tools in VMD are domain-specific, and they support multiple VR toolkits. Tools also create force feedback for haptic input devices, and they maintain a visual representation of the tool orientation within the virtual environment. VMD provides several standard tools for different visualization and interactive modeling tasks. The VMD “tug” and “pinch” tools apply forces to atoms or higher level molecular structure components within interactive simulations. A “grab” tool allows 3DOF and 6DOF input devices to arbitrarily translate and rotate the molecular scene. A “rotate” tool allows 3DOF input devices to rotate the scene. A “spring” tool enables precise visual placement of spring constraints used in molecular dynamics simulations. A “print” tool logs input device position and orientation data to a text console to assist the users with VR input device testing and calibration. Additional tools are easily implemented by subclassing a top level tool base class.

2.4 User Interfaces

A potential limitation to the application of immersive VR for molecular modeling is the need for alphanumeric input and simultaneous display of multiple properties of the molecular structure, often in other visualization modalities such as as 2-D plots of internal coordinates, timeline views of simulation trajectories, and tabular displays of alphanumeric information. Although such interfaces can be embedded within immersive environments, and some have been implemented in modified versions of VMD [12], users often find them inefficient relative to traditional 2-D interfaces and keyboard input. For this reason, VMD maintains the ability to display 2-D desktop graphical interfaces concurrently with an immersive virtual environment, by redirecting them to another windowing system console. The 2-D user interfaces are often most useful during exploration of large and unfamiliar molecular complexes when a user may wish to make many detailed queries about the model, not limiting themselves to 3-D interactions, visual representations, or an immersive environment.

Several groups have demonstrated the utility of incorporating 2-D GUI toolkits into VR applications as a partial solution to the need for auxiliary interfaces while working within immersive environments [15,16,17]. Since VMD incorporates Tcl and Python scripting interfaces, it is also possible to control the application or to display results through web browsers, graphical interfaces, gesture interfaces, and voice interfaces hosted on auxiliary desktop computers, tablet computers, or smartphones using a variety of network protocols. VMD’s scripting interfaces also enable creation of user-customized interfaces specific to the project that they are working on. All of the input device modalities supported in VMD (VRPN, CAVE, etc.) can be made to trigger scripting language event callbacks so they can be used in an arbitrary user-defined way.

3 Interactive Molecular Dynamics

A compelling use of the combination of immersive display and interaction features within VMD is interactive molecular dynamics (IMD) simulation [13]. Steered and interactive molecular dynamics simulations can be used to study the binding properties of biomolecules and their response to mechanical forces, or allow nanomechanical experiments, such as surface grafting or manipulation of simulated nano-devices or nanoscale objects like carbon nanotubes of fullerenes. The development of these ideas began with steered molecular dynamics techniques (SMD) that enabled runtime simulation visualization with limited support for interaction with the simulation [1,2] and has subsequently evolved toward fully interactive molecular dynamics simulation [13,18]. The fully interactive nature of IMD and similar techniques holds promise for both research [19,20] and teaching [6], but, until recently, the computational requirements for IMD hindered its applicability, limiting its use to relatively small molecular systems, or requiring the use of HPC clusters or supercomputers for IMD simulations of larger structures. Even under those restrictions, IMD has proved to be a valuable tool for education and outreach, giving non-scientists a captivating view – and, in combination with haptics, also a feel – of the world of computer simulations.

Recent advances in the use of GPU computing to accelerate molecular dynamics simulations have brought the performance of GPU-accelerated desktop workstations up to the level of small or mid-sized HPC clusters [21,22,23], largely eliminating the need for end-users to have expertise in using and managing HPC clusters, and making it possible to perform IMD simulations of moderate size molecular structures on a single GPU-accelerated desktop workstation. The advantage of GPU-acceleration is more pronounced in nano-mechanical and nano-chemical modeling because the many-body models used (e.g. Tersoff, Stillinger-Weber, AIREBO) have a much higher algorithmic complexity than potentials used in life sciences and, thus, benefit more from the GPU hardware architecture. Even for non-GPU accelerated applications, the overall performance of a multi-socket, multi-core desktop workstation can be as high as that of typical moderately sized HPC clusters of less than ten years ago.

3.1 IMD Software Design

VMD supports live display and interaction with running molecular dynamics (MD) simulations through two main software interfaces: a network channel to a molecular dynamics simulation engine, and an input device for motion control, ideally with 6DOF input and haptic feedback. Atomic coordinates, applied and resulting simulation forces, and global simulation properties such as energies, simulation volume, pressure and other quantities are continuously exchanged between VMD and the attached molecular dynamics simulation engine through a TCP/IP socket. This enables VMD to be coupled to a simulation running on the same workstation, or alternatively to a simulation running on a remote HPC cluster or supercomputer, enabling simulations of much larger molecular complexes while maintaining interactivity.

A special “Tug” tool (see description of VR input mechanisms above) allows groups of atoms to be linked to a haptic device. For the duration of time the user activates the haptic device, its effector follows the linked object and any force that is exerted on the effector will be translated by VMD to a force on the atoms and communicated to the ongoing MD simulation. A user can feel any objects that the linked atoms bump into, how much (relative) force is necessary to move the linked atoms to a specific position, and also get an impression of the linked atom’s inertia.

3.2 Physical Realism and Technical Challenges

Using IMD in combination with haptic devices creates new and unique challenges to molecular visualization applications. The length scales of all-atom classical molecular dynamics simulations are in the range of nanometers; at the same time, those simulations describe processes that happen on pico- to nanosecond time scales. In the visualization process this has to be converted to something that can be managed by the human perception, i.e. the objects are shown many orders of magnitude larger and processes have to be slowed down correspondingly.

Typically a molecular dynamics simulation is calculated “off-line” and often “off-site”, i.e. without directly visualizing simulation progress, and on a different machine from where the visualization is performed. During the simulation, configuration snapshots are regularly stored, transferred to the visualization site and then read into the visualization software to be viewed and analyzed. Typical problem sizes often require a substantial amount of computational power as provided by clusters or supercomputing centers. Consequently, no interactive manipulation is possible, although a number of biasing methods like steered molecular dynamics or meta-dynamics exist that can “guide” a system from one state to another via predefined recipes.

The alignment of time and length scales for non-interactive simulations is primarily a question of how efficiently the molecular visualization software can render the data. Adjustments can be made by either slowing down animations or by reducing the level of detail. In difficult cases even the visualization can be done “off-line” by producing movies via batch-mode rendering.

For interactive viewing of the MD simulation this flexibility is no longer available. The MD simulation must keep pace with the interactive visualization so that the molecular processes of interest can be viewed at interactive speeds. While less a problem for small systems, this is a big problem for large systems, as a powerful (parallel) compute resource will be needed. The remote computation resource also needs to be joined with the visualization resource via a high throughput/low latency link to enable smooth, stutter-free interaction.

Additional complications arise in using a haptic device to interactively manipulate the system in the ongoing simulation. The perceived inertia of the manipulated object depends on the MD simulation rate. The faster the MD simulation runs, the lighter an object “feels” and the more easily it can be manipulated. In principle, this can be adjusted by a scaling factor when computing the force to be sent to the simulation, based on the position of the effector of the haptic

device relative to the atoms it is linked to, but the larger this scaling factor, the less realistic the simulation. If the scaling factor is too large, the MD integration algorithm can become unstable.

This becomes even more difficult if one is interested in manipulating a large and slowly moving object immersed in many small and fast ones (e.g. a protein solvated in water, or a cantilever of an atomic force microscope). The visualization time scale would have to follow this slow object, but the simulation must be run at an appropriate resolution and timescale for the small objects. Thus, the demands on the MD simulation performance are extremely high, yet the frame rate of the visualization software and human perception limits how quickly frames can be processed and need to be communicated, potentially resulting in a jumpy representation of such small objects. This applies as well to the force feedback, where an overly-soft coupling of the haptic device to atoms will result in an indirect feel, while a strong coupling will make an object linked to the effector feel jittery or lead to unwanted and unphysical resonances in the force-feedback coupling. Filtering of high frequency molecular motions within the molecular dynamics simulation engine is likely the best method for addressing such timescale-related haptic interaction issues.

4 Future Direction

The increasing computational capabilities of commodity hardware and the availability of affordable and commonly available advanced input devices will allow more realistic interactive modeling and more intuitive and flexible interaction with the visualization. Although VMD supports many 6DOF input modalities, the technical difficulty involved in configuring and maintaining such input devices remains a hurdle for the general user community. For mainstream adoption by non-VR-experts, such devices need to move from being a niche item to something that the majority of users have and can regularly use. In addition, a slight reduction in “immersiveness” and a significant reduction in the (perceived) complexity of VR components, in combination with the use of commodity hardware, would lead to affordable interactive molecular simulation “appliances”. Such appliances could be preconfigured with VR scenarios that would only require that the user provide the molecular model and would automate configuration, launching, and connecting the different hardware and software components.

The perceived value of a given component of a VR or immersive display system directly impacts the amount of use of that component for a given session. We have observed that users often utilize a subset of the available VR components (usually choosing those that are least disruptive to their workflow). They might use stereo visualization, but not an enhanced input device, or vice versa. Ultimately, input devices and their interactions with the VR environment must correspond to well understood metaphors in order to gain acceptance among application scientists. Devices that operate in a familiar way are more apt to be used than unique or special purpose VR devices. This makes a strong case

for adoption of new input technologies such as multitouch user interfaces that have become common in recent smartphones (e.g. the iPhone, Android, etc.) and tablets (e.g. iPad) and are slowly becoming a standard feature of mainstream desktop operating systems. As multitouch input devices become more broadly deployed and their associated programming APIs become more standardized, many new opportunities for use will arise in the domain of molecular modeling, particularly for workbenches, walls, and other display modalities of particular interest for small-group collaborative visualization.

While the standardization of input conventions, gestures, and APIs for multitouch input is an area of ongoing effort, all modern smartphones include accelerometers and cameras, and new phones such as the iPhone 4 include gyroscope instrumentation. Together, these provide the necessary data for various kinds of 6DOF motion control, text input, auxiliary 2-D graphical interfaces, and voice input, all in a familiar package that users already own [15,16,17,24]. This will encourage everyday use of 6DOF motion control, and in an ideal case, will enable a convenient means for collaborative visualization and modeling among small groups of researchers, wirelessly, and with none of the typical burdensome installation required by traditional 6DOF VR input devices.

We have developed a prototype user interface for VMD that allows a smartphone to be used as a wireless touchpad or 6DOF wand, using the touch sensitive surface of the phone display and 6DOF data obtained from on-board accelerometer and magnetometer instrumentation, respectively. We envision smartphones being particularly useful for wireless control during interactive presentations, and in multi-user collaboration scenarios. In our prototype implementation, the smartphone communicates with VMD via datagram (UDP) packets sent over a local IEEE 802.11 wireless network, and a VMD input device subclass listens for incoming motion control messages, potentially from multiple phones. Our initial experiments have shown that current smartphones hold great potential both as 6DOF input devices, and potentially for various other input and program control modalities. The responsiveness of our prototype implementation has already demonstrated that smartphone 6DOF processing overhead and local WiFi network latencies are qualitatively low enough to be tolerable for typical molecular modeling tasks. Much work remains to improve the quality of 6DOF orientation data, particularly on smartphones that lack gyroscopes. We found that the use of magnetometer data in the orientation calculation could lead to erratic results in some workspaces due to proximity to metal furniture, etc. We expect that smartphones incorporating on-board gyroscopes will provide higher quality 6DOF orientation data and will be less susceptible to errors arising from the physical workspace. If successful, these developments will pave the way for smartphones to be used as ubiquitous multi-modal input devices and auxiliary displays in both single-user and collaborative settings, all without the installation, wiring, and other hassles that have limited the use of traditional VR input mechanisms among molecular scientists.

Acknowledgments

This work was supported by the National Institutes of Health, under grant P41-RR05969 and the National Science Foundation, under grant no. 0946358. The authors wish to thank Justin Gullingsrud, Paul Grayson, Marc Baaden, and Martijn Kragtwijk for their code contributions and feedback related to the VR and haptics interfaces in VMD over the years. We would also like to thank Russell Taylor for the development and ongoing maintenance of VRPN, and many useful discussions related to haptic interfaces. A.K. thanks Tom Anderson of Novint Inc. for donation of two Falcon devices for implementing VRPN support and Greg and Gary Scantlen for stimulating discussions and more.

References

1. Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kalé, L., Skeel, R., Schulten, K., Kufirin, R.: MDScope – A visual computing environment for structural biology. In: Atluri, S., Yagawa, G., Cruse, T. (eds.) *Computational Mechanics 1995*, vol. 1, pp. 476–481 (1995)
2. Leech, J., Prins, J., Hermans, J.: SMD: Visual steering of molecular dynamics for protein design. *IEEE Comp. Sci. Eng.* 3, 38–45 (1996)
3. Humphrey, W., Dalke, A., Schulten, K.: VMD – Visual Molecular Dynamics. *J. Mol. Graphics* 14, 33–38 (1996)
4. Ihlenfeldt, W.D.: Virtual reality in chemistry. *J. Mol. Mod.* 3, 386–402 (1997)
5. Sharma, R., Zeller, M., Pavlovic, V.I., Huang, T.S., Lo, Z., Chu, S., Zhao, Y., Phillips, J.C., Schulten, K.: Speech/gesture interface to a visual-computing environment. *IEEE Comp. Graph. App.* 20, 29–37 (2000)
6. Sankaranarayanan, G., Weghorst, S., Sanner, M., Gillet, A., Olson, A.: Role of haptics in teaching structural molecular biology. In: *International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, p. 363 (2003)
7. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A.: Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In: *Proceedings of SIGGRAPH 1993*, Anaheim, CA., pp. 135–142. ACM, New York (1993)
8. Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G.L., Brown, M.D.: The ImmersaDesk and Infinity Wall projection-based virtual reality displays. *SIGGRAPH Comput. Graph.* 31, 46–49 (1997)
9. Pape, D., Anstey, J., Sherman, B.: Commodity-based projection VR. In: *SIGGRAPH 2004: ACM SIGGRAPH 2004 Course Notes*, p. 19. ACM, New York (2004)
10. Taylor II, R.M., Hudson, T.C., Seeger, A., Weber, H., Juliano, J., Helser, A.T.: VRPN: a device-independent, network-transparent VR peripheral system. In: *VRST 2001: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 55–61. ACM, New York (2001)
11. Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., Cruz-Neira, C.: VR Juggler: a virtual platform for virtual reality application development. In: *Proceedings of IEEE Virtual Reality*, pp. 89–96 (2001)
12. Martens, J.B., Qi, W., Aliakseyeu, D., Kok, A.J.F., van Liere, R.: Experiencing 3D interactions in virtual reality and augmented reality. In: *EUSAI 2004: Proceedings of the 2nd European Union Symposium on Ambient Intelligence*, pp. 25–28. ACM, New York (2004)

13. Stone, J., Gullingsrud, J., Grayson, P., Schulten, K.: A system for interactive molecular dynamics simulation. In: Hughes, J.F., Séquin, C.H. (eds.) 2001 ACM Symposium on Interactive 3D Graphics, New York. ACM SIGGRAPH, pp. 191–194 (2001)
14. Kreylos, O.: Environment-independent VR development. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 901–912. Springer, Heidelberg (2008)
15. Angus, I.G., Sowizral, H.A.: Embedding the 2D interaction metaphor in a real 3D virtual environment, vol. 2409, pp. 282–293. SPIE, San Jose (1995)
16. Watsen, K., Darken, R.P., Capps, M.V.: A handheld computer as an interaction device to a virtual environment. In: Proceedings of the Third Immersive Projection Technology Workshop (1999)
17. Hartling, P.L., Bierbaum, A.D., Cruz-Niera, C.: Tweek: Merging 2D and 3D interaction in immersive environments. In: Proceedings of the 6th World Multiconference on Systemics, Cybernetics, and Informatics, Orlando, FL, USA, vol. VI, pp. 1–5 (2002)
18. Férey, N., Delalande, O., Grasseau, G., Baaden, M.: A VR framework for interacting with molecular simulations. In: VRST 2008: Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology, pp. 91–94. ACM, New York (2008)
19. Grayson, P., Tajkhorshid, E., Schulten, K.: Mechanisms of selectivity in channels and enzymes studied with interactive molecular dynamics. *Biophys. J.* 85, 36–48 (2003)
20. Hamdi, M., Ferreira, A., Sharma, G., Mavroidis, C.: Prototyping bio-nanorobots using molecular dynamics simulation and virtual reality. *Microelectronics Journal* 39, 190–201 (2008)
21. Phillips, J.C., Stone, J.E., Schulten, K.: Adapting a message-driven parallel application to GPU-accelerated clusters. In: SC 2008: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, Piscataway, NJ, USA. IEEE Press, Los Alamitos (2008)
22. Anderson, J.A., Lorenz, C.D., Travesset, A.: General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Chem. Phys.* 227, 5342–5359 (2008)
23. Stone, J.E., Hardy, D.J., Ufimtsev, I.S., Schulten, K.: GPU-accelerated molecular modeling coming of age. *J. Mol. Graph. Model.* 29, 116–125 (2010)
24. Hachet, M., Kitamura, Y.: 3D interaction with and from handheld computers. In: Proceedings of IEEE VR 2005 Workshop: New Directions in 3D User Interfaces. IEEE, Los Alamitos (2005)

Multi-institutional Collaboration in Delivery of Team-Project-Based Computer Graphics Studio Courses

Tim McLaughlin¹, B. Adán Peña¹, Todd A. Fechter², Anton Markus Pasing³,
Judith Reitz³, and Joseph A. Vidal⁴

¹ Department of Visualization, Texas A&M University

² Arts & Technology, The University of Texas at Dallas

³ Akademie für Internationale Bildung, Bonn, Germany

⁴ Design and Technology Academy, San Antonio, Texas

Abstract. Effective use of computer graphics for technical and artistic exploration often requires the participation of multiple teams representing specific knowledge domains though these teams may be separated by both geography and time zones. This paper reports on the introduction of a project organized by four academic institutions oriented around collaborative technical and visual problem solving among non-co-located students. The project was developed to match the curricular requirements of existing courses. Participants included undergraduates at two U.S. universities and U.S. students studying in Western Europe, as well as a group of U.S. high school students. This paper specifically details the organizational issues, curricular alignments, and employment of affordable information technology for both workflow coordination and communication among team members. The results indicate that the project economically utilized course time, contributed to learning objectives aligned with work force trends in the animation industry, and leveraged commonalities of existing computing infrastructure along with commodity computing services for positive effect.

1 Introduction

Information technology enables collaboration among world-wide project participants, including the delivery of visual computing related courses and projects. Students pursuing education in visual computing fields are likely to eventually contribute as domain-area specialists to projects involving distant collaborators separated by both physical distance and time zones. Effective multi-disciplinary collaboration includes the ability to communicate synchronously and asynchronously between individuals and among groups. Effective collaboration also requires organizational components such as scheduling, workflow coordination, and specificity of data types and information to a degree of shared understanding significantly beyond the requirements of individual work or group projects among co-located team members.

As educators with the goal of preparing students for the world outside of the academy, we want to expose students to the issues surrounding distance collaboration. We want to provide them with an understanding of the differences between working in the physical presence of our collaborators and using technology to allow us to work at a distance [1]. We want to broaden the students' exposure to the thoughts and ideas of both students and instructors at other programs. We also want to take advantage of the commodification of information technology by broadening opportunities to practice cross-disciplinary collaboration across institutions.

This paper focuses on the organizational components and technical infrastructure required to extend participation in group projects beyond students at a single university. Collaborative projects provide opportunities for students to learn from one another and contribute their knowledge and skills to a larger goal than is possible in independent projects. When structured as a reflection of industry practice, collaborative projects also create a fluency for students about processes and considerations that are important to future employers. This project focused on how a group of institutions can partner with one another for the benefit of their respective students through exposure to distance collaboration and creative problem solving with distant partners. In this paper we show that multi-institutional collaborative student projects can be managed effectively without significant changes to curricular or infrastructure requirements. We assume that the effectiveness of the existing courses is already validated. The results include specific organizational requirements, technical requirements, and expectations for faculty and student participants. We have defined the conditions through which remote collaboration projects can be setup that match conditions within a targeted industry, and provided the students with an experience that promotes fluency with issues they are expected to encounter in their careers following formal education.

Our approach differs from other uses of distance education in that the students are participating as area specialists with non-co-located partners. The project did not consume the entire calendar for each location. The work was coordinated to include overlaps and periods in which there was little or no overlap. We also used existing commercial technology to facilitate the IT requirements of the project. This project brought together academia and industry to address the issue of how creative problem solving is facilitated among non-colocated team members. Industry participants, acting in an advisory role, included artists, engineers, and managers from Lucasfilm Ltd., DreamWorks Animation, ReelFX Creative Studios, Digital Domain, and Adobe Systems. These companies are interested in this project because it directly addresses issues they face in the global market of development of technically and creatively complex products and intellectual properties. Educators and administrators at each institution are interested in the project due to the potential to extend the reach of creativity enhancing and technically challenging curricula outside the brick-and-mortar classroom.

2 Prior Work

2.1 Collaborative Projects

The studio approach for instruction in undergraduate computing-intensive courses has been used in practice for only about 20 years, as documented by Tomakyo at Carnegie Mellon University in 1987 [2]. Cooperative learning environments are able to achieve significantly better student performance through peer-to-peer mentoring [3], and high levels of collaborative learning increase student satisfaction with a courses [4]. Identified benefits of the studio approach for computing-intensive work are the extension of the skill set of the group, the capacity to employ technology attuned to the individual needs of groups members, and ownership of responsibility for outcomes assumed by the group members [5].

In 1990, Rensselaer Polytechnic Institute in Troy, New York, instituted change across the curriculum embracing studio-style instruction [5]. Such substantial change was undertaken to both improve the educational experience in science and engineering, and to adapt a model for delivery of education that was extensible through distance learning. Among the conclusions drawn from the experience was the importance of highly interactive computing and communication tools [6]. The embrace of studio-style learning for computing-intensive subjects follows over 100 years of studio-style education in art and design and repeated empirical analysis of the connection between studio learning and creative thinking [7]. However, information technology is changing the way studio courses are taught, both in computing and design [1]. Rapid integration of new technology is required if educational institutions intend to continue to supply skilled workers to industry and maintain effective delivery of computationally intensive education experiences for growing numbers of students.

2.2 Technology Development for Synchronous Collaboration

Studio style learning is dependent upon tight communication and cooperation among participants. When collaborators are not co-located the success of their partnerships can be in large part determined by information technology. Participants in large projects who are not co-located rely on both synchronous and asynchronous exchanges of information. Verbal interaction is recognized for its superiority over asynchronous information exchanges such as email and message boards for both efficiency and qualitative reasons. Awareness of a collaborators progress and intentions, as supported by verbal exchanges, are critical to successful collaboration. Tools that provide awareness through visual means have been shown to promote efficiency, increase accuracy, and contribute to qualitative improvements when used over time [8]. When partners are problem solving using visual imagery, synchronicity of verbal communication and collaborative viewing of the imagery is essential. [9] developed a video architecture facilitating synchronization of video presentation through the use of a video server, several video clients, an autonomic controller, and a common communications

infrastructure. [10] developed a web-based system for collaborative virtual science lab environments using Java applets within the open source tool Easy Java Simulations.

3 Our Contribution

The primary focus of this project was the enhancement of undergraduate learning in undergraduate design studios that require both artistic and technical problem solving. The design of this project reflects a project management issue faced by many companies in the animation industry where high levels of technical and visual creativity and collaboration are required from a global workforce. Our approach is fundamentally dependent upon solving administrative, technical, and pedagogy delivery issues to facilitate the participation of students from multiple learning institutions. In this arena we have produced findings that are instructive both in terms of the conditions that promoted our primary objective of enhanced student learning and some issues that interfered with enhancement of learning. Our findings are grouped in three sub-topics discussed in detail below: (a) institutional organization, (b) alignment of pedagogy, and (c) technology.

3.1 Project Funding

Significantly, funding to facilitate the technology support for the project was very low relative to the standards for computer graphics oriented projects. To execute the project described in this paper expenditures on technology totaled 696 USD. Later in this paper differentiation will be made between technology specific to this project and technology already existing at the collaborating institutions. This project was phase one of a three year investigation funded by the National Science Foundation's CreativeIT Program within the Division of Information & Intelligent Systems (NSF Award #0855908). This program explores cross disciplinary research in creativity, computer science, and information technology with a particular focus on creative problem solving. Funding from NSF, in addition to supporting the Principal Investigator's work on the project, included support for a graduate student at the PI's institution, and travel assistance funding for the faculty and industry advisors to gather at a professional conference prior to the initiation of the project.

3.2 Institutional Alignment

In the Spring of 2010, faculty from four academic institutions embarked on a joint effort to engage students at each location with partners at each of the other three institutions in the development and execution of short animation projects requiring cooperative problems solving in both aesthetic and technical issues. Prior to the initiation of this project, the four institutions involved were all connected in some way not directly related to the project to the institution

initiating the project. Formal planning and organization for the project began approximately five months prior to the first day of participation by the students involved.

This project initiated within the Department of Visualization at Texas A&M University (for clarity hereafter called AU). AU is a state funded research-oriented university in College Station, Texas, U.S.A. that serves a large population of both undergraduate and graduate students. Initiating and participating in projects of this kind is an expected contribution for faculty at AU. The curricular goals of this project match the learning outcomes expected for undergraduates pursuing the Bachelor of Science in Visualization degree, a studio-based program that mixes equal parts of math, science, and logic competency with art and digitally composed visual media competency.

The European participants were students participating in a study abroad program sponsored by AU but administered by the Academy for International Education located in Bonn, Germany and managed by German educators (hereafter called EU). The main campus for EU is approximately 8,200 km (5,100 miles) and six time zones ahead of AU. EU custom designs programs to provide international experience, cross-cultural competence, and professional skills. EU is not-for-profit and coordinates with multiple US-based universities to conduct programs in the academic fields of Business, Liberal Arts, Language and Cultural Studies, Film and TV, Theater, Political Science, Engineering, Architecture, and Landscape Architecture. Participation in a project of this kind is uncommon for EU, but falls within the expectations of responsible partnership toward curricular goals that EU and its faculty share with their affiliated US-based institutions.

The other US-based university participating in this project was the University of Texas at Dallas (hereafter called BU). BU is also a state funded research-oriented university located in Richardson, Texas U.S.A. It is in a separate university system from AU. The two campuses are roughly 300 km (190 miles) apart, but in the same time zone. Students participating from BU were enrolled in the Arts & Technology undergraduate program that focuses on the intersection points of art, design, technology, and science. Participation in projects of this kind is an expectation for faculty at BU.

The participating high school was the Design and Technology Academy (hereafter called HS), a community funded school located in San Antonio, Texas, U.S.A. HS is roughly 260 km (160 miles) from AU and 450 km (280 miles) from BU and shares the same time zone. HS has a four year (9th through 12th grade) magnet program focused on design and technology. Participation in projects of this kind is uncommon for faculty at HS, but falls within their scope of responsibilities for contributing to the currency of the academic program and contributing to outreach and external profile of the institution.

Initiation of this project required the interest and agreement of the faculty members directly involved at each institution as well as the administrators of the academic programs in which the student participants were enrolled. An important feature of the project setup is that no additional resources were expected

to be required by any of the participating institutions. This did not prove to be precisely true (as is discussed in the Technology portion of this paper) but remained true in large part. The participating faculty taught the courses and course load they would have normally taught even if the project did not exist. However, the scheduling of weekday and meeting times of the classes at AU, BU, and EU were coordinated to provide overlapping times when all students would be in class simultaneously.

Unfortunately, the scheduling of classes at HS did not permit a course time overlap. The absence of course time overlap with the participants from HS led to an unexpected and significant lack of coordination between the work of students at HS and the work of students at AU and BU. The factor was not felt as heavily between HS and EU since students from EU contributed primarily at the beginning of the project workflow and students at HS contributed primarily at the end of the project workflow. The term, or semester, calendars at the four locations did not line up precisely. The greatest negative impact of imperfect alignment stemmed from the fact that the students at EU were primarily responsible for the first phase of the project pipeline yet were the last group to begin the school term. The difference was only a matter of one week, but within a 15 week semester the impact was significant.

In the five month period prior to the first day of student involvement, the faculty collaborating on the project held one face-to-face meeting at a professional conference. This meeting also included the project's industry advisory board. At this occasion the general goals and organization of the project were discussed and a variety of technical solutions were considered for further investigation. Following the face-to-face meeting, and approximately one month prior to the start of student participation, the faculty participated in a group conference call. During this call the final details for the calendar of the project, roles of participants, and outstanding technology questions were addressed. Email was used by the faculty and graduate assistant on the project to plan, discuss, share documents, and determine solutions for issues that did not get answered during the face-to-face meeting or conference call.

3.3 Curriculum Alignment

In the case of AU, BU, and EU, the project was conducted as a part of existing regularly taught courses. For HS the project replaced a professional study, or internship, component of the curriculum that is a requirement for graduation. As such, the project was new in the form in which it was delivered, but did not increase curricular requirements. At each location students were informed about the project prior to the start of the term in which the project took place and were offered alternative curricular equivalents if they did not wish to participate.

This project involved four institutions and four courses with separate, though related, course requirements and expected learning outcomes. A primary goal in organizing the project was to preserve the existing curricula at the participating locations. We reasoned that if this project succeeded in creating a connection

between courses at the participating institutions, the significant impact of the project would be greatest if the courses were part of the existing curricula. The academic programs involved in this project at each institution are aligned with the study of topics in computer graphics that have application to the technical aspects of animation, visual effects, and video game development. As such, the workflow of this project was organized to roughly match processes and disciplines found in the production of 3D computer graphics animation. Within each academic program the learning objectives for students include awareness of industry practices, skills specific to the generation of computer graphics imagery, visual design problem solving, programing and technical problem solving.

In the preparation time prior to student involvement in the project, the faculty determined the scope of the project to be undertaken and how the work would be divided among the collaborating locations. It was determined that the students would be divided into two teams at each location with each team responsible for producing a 3D animated short of approximately 30 seconds in length. Each project was required to include two characters, one or two environments, and one or two effects such as fire, smoke, sparks, or an explosion. The specifications for the project did not include title design or sound design. Division of work was based upon discipline responsibilities and workflow common to the animation industry. The work was divided between the schools based upon the learning objectives of the specific courses at each location.

Contribution of the students to the group project was arranged so that the time required did not consume the full 15 weeks of the semester. See Table 1 for the project calendar. This created time within the semester to concentrate on topics and projects that were not directly tied to the group project. For example, students at AU began story and concept development beginning on their first day of class (Week 2 for the project). For much of the Week 4 and Week 5 they were exploring general topics related to scripting languages, motion, control, and deformation systems through lectures and short experimental projects. In Week 6 through Week 12 they were deeply invested the collaborative project. In Week 13 through 15 they worked on individual projects that were unconnected to the group project. Similarly, at BU the students were not heavily involved in the group project for Weeks 1, 5, 6, or 7. The students at EU completed their work on the project in Week 7. A positive effect of this schedule is that the overall project was large in scope for a 15-week schedule yet did not consume all of the time in any of the courses. A drawback to this schedule is that the students ranged between being intensely connected to the project and being disinterested. Their investment in solving problems related to the project was directly related to the calendar. For example, the students at EU and AU, whose work contributed near the beginning of the project, were intensely engaged in the Story and Look Development pitches for which all participants were to take part. Conversely, the students at BU and HS, whose primary responsibilities to the project did not occur until later were not as engaged in the Story and Look Development pitches.

3.4 Student Participants and Team Structure

In total, 42 students took part in this project. There were 18 undergraduates spread between the two US institutions (AU & BU), 16 students at the European-based program (EU), and eight high school juniors at the participating high school program (HS). English was the native spoken and written language for all of the students. The undergraduate students were all already proficient in the use of computing technology including both the software required to perform tasks to contribute to the project delivery and email, which was required to communicate with team members at the other locations. Students in the high school program spent a significant portion of the semester learning software leading up to their direct contribution to the work.

At each location the students were divided into two teams. For purposes of this paper we will call them *Team A* and *Team B*. Each team, therefore, was composed of 21 members: eight members from EU, four members from AU, five members from BU, and four members from HS. During the Story Development and Pitch phase of the project, each team at each location developed and pitched

Table 1. Calendar of collaborative project activities during the 15-week semester

Week	Activities	Locations
Week 1	Course & project introduction	BU
Week 2	Course & project introduction	AU & HS
Week 3	Course & project introduction	EU
	Story development begins	AU, BU & EU
	1st virtual meetings between student team members	AU, BU & EU
Week 4	Story ideas pitched	AU, BU & EU
	Story ideas selected via voting	All
Week 5	Modeling, layout & visual style development begins	EU
Week 6	Development of anim. control & deformation systems begins	AU
	1st virtual project review: rough models, layout & vis. dev.	EU & AU
Week 7	Character animation tests begin	AU
	Mid-project virtual meeting of instructors	All
	2nd virtual project review: final models, layout & vis. dev.	AU, BU & EU
	Modeling, layout & visual development completed	EU
Week 8	Effects animation tests begin	HS
	Surfacing begins	BU
Week 9	Animation control systems work completed	AU
Week 10	3rd virtual project review: rough animation	AU & BU
	Lighting tests begin	BU
Week 11	4th virtual project review: surfacing & lighting tests	AU & BU
	Animation review	AU & BU
Week 12	Animation completed	AU
Week 13	Continuation of lighting	BU
	Compositing tests	BU
	Continuation of effects animation	HS
Week 14	6th virtual project review: final lighting	AU & BU
Week 15	Final review	AU, BU & EU

a story idea. Other than this phase, the participants on each team worked collectively on the same project. The teams were purposefully organized with no clear leadership structure. No location was specified as the lead on the project and no individual was identified as the director by the faculty. During email exchanges and virtual project reviews strong opinions emerged, but the students were left on their own to devise solutions to the conflicts. In the end there was some deference to the opinions of the originators of the story pitches, but each location internally solved the problems over which they had local control as they deemed fit. This could have resulted in something akin to the *telephone game* in which a secret is whispered into the ear of a succession of children and by the end the secret that is told by the last child bears little resemblance to what the first child initially said. However, perhaps surprisingly, both team projects emerged from the production process bearing a significantly strong correlation to the original story pitch, including visual style and emotional tone. Even without defined leadership the collaborators accommodated differences of opinion and managed to successfully stay on course.

3.5 Technology

Each academic institution involved in this project had both commonality and differences in their respective approaches to information technology, hardware, and software. Overall, there was more commonality than differences. One does not have to look very far into the past to find a time when the prospect of connecting students at different schools together on a computer graphics related project would have been extremely difficult due to the significant differences between levels of computing and incompatibility between systems and file formats. One of the driving forces behind embarking on this project was recognition that the capacity to connect electronically has exceeded our knowledge and skill at doing so for collaborative effectiveness. Another goal for this project was to minimize the universality requirement between collaborators. Universality is the degree to which workflow -systems, directory structures, software, interface setups, naming conventions and file formats, is consistent. In commercial production universality is a key factor in contributing to or impeding productivity. Universality within a single project and as teams move from project to project is desirable feature. A high degree of universality is possible in top-down structures such as commercial production where technology and artistic leads jointly determine workflow standards. Education, particularly when multiple non-affiliated institutions are involved, has many features that prohibit top-down driven universality. To develop the technology requirements for this project we divided the issue into two separate areas: communications technology and production technology. Collectively, these two areas comprise the project's information technology (IT).

3.6 Cross-Site Tool and Workflow Alignment

This project was structured so that students at each location would contribute a specific portion of the work making up the completed 30-second short animation.

Such a workflow requires that data from preceding portions of the production pipeline can be loss-lessly incorporated by the succeeding pipeline steps. Specific to our project, models and camera scenes from EU were used by AU, BU, and HS; animation from AU was used by BU and HS; lighting from BU was used by HS; and effects animation from HS was used by BU. We also worked to keep the pipeline as open as possible to software and workflow preferences at each location. Forcing the use of a specific piece of software potentially dictates pedagogical and budgetary decisions. To facilitate the lossless transfer of data this project we determined that all models would be delivered in *.obj* format. We determined a unit size and global orientation standard within 3D software packages, for example: $1 \text{ unit} = 1 \text{ foot}$; and all character models should face down the *positive z-axis*, for example. These guidelines were distributed to all students at the beginning of the project. As it turned out, Autodesk's Maya 3D animation software was the preferred choice of all of the students at each location. Maya is a standard tool within animation and visual effects. At BU Maya was already licensed by the academic institutions and available for the students to use. Students at AU, EU, and HS took advantage of Autodesk's free trial opportunity available only for students.

Computing power and connectivity varied from location to location among the participating schools. At BU and HS students performed their work in computer tutorial classrooms on graphical workstations provided by the institutions. These machines were connected in a local area network and possessed, optimally, 1 Gbps ethernet connection to the outside world. Students at AU performed their work in a studio classroom, not specialized for computing, using personal laptops or personal desktop machines. They communicated to the outside world using a VPN controlled wireless network with approximately 54 Mbps connections. Students at EU also worked in a studio classroom using their personal laptop machines. They connected to the outside world either through 18 Mbps data ports or 3 Mbps wireless connections. At EU the institution purchased higher performance routers once the administrators recognized the extent to which connectivity was going to play a role in the execution of the course and project.

3.7 Cross-Site File Sharing

Sharing large files containing models, animation, and image sequences among a large team is a hallmark of animation production. To facilitate this facet of the project we utilized a cloud-based service with controlled access provided by Dropbox. Published by Evenflow and made publicly available in 2008, Dropbox offers cross-platform compatibility for Windows, Mac, and Linux. Users place files into a designated folder on their own machine and synchronization with the Dropbox cloud server occurs automatically when changes are detected. Basic 2 GB service is free and each student on the project set up his or her own Dropbox folder. A home folder for the project on server partition at AU provided a 50 GB file storage location for which Dropbox charged 99 USD for a year. Access to the files on in Dropbox folders is password protected. A one-month revision history is included in the service. The file structure in the primary project folder

was originally left to the students to organize. By mid-project however, there was a great deal of confusion over where the latest versions of files were located and what file naming convention should be used. With agreement of the faculty the graduate assistant instituted a file structure and attempted to enforce its use throughout the second half of the project.

3.8 Synchronous and Asynchronous Communication

To provide synchronous viewing of visual media and verbal communication between team members at different locations we combined the use of two commercial software applications: cineSync and Skype. The former is a synchronized media player permitting control at multiple locations. CineSync includes drawing and text annotation capabilities. It is built on the Apple QuickTime player. Media files can be constructed into a playlist and played back in perfect synchronisation at any location invited to participate in the review session. A particular advantage of cineSync for animation review is frame specificity and control. CineSync is downloadable for Mac and Windows. The cost for five users for six months was 597 USD. A version with additional features, including support for Linux, is available at a higher price but was not required for this project. Though cineSync includes audio support for verbal discussions we chose to use Skype running alongside cineSync. Skype utilizes VOIP technology and is free to non-commercial users. In addition to VOIP Skype offers instant messaging. This feature was highly useful to the faculty and graduate assistant on the project for quickly sending small snippets of information while organizing cineSync virtual reviews (dailies) between locations.

To provide a mechanism for detailed communication we provided three project specific email accounts: one was an overall project email and was intended for students to be able to contact the faculty and graduate assistant with questions about the project. The other two email accounts were team specific. Everyone on Team A was connected to one account while everyone on Team B was connected to another. We used Google's mail service, *gmail* because it was free and, significantly, it was not the domain of any of the institutions involved. For a variety of reasons, including security, institutions are reasonably averse to providing email accounts and access to individuals who are not either faculty, staff or students of that institution. The use of an outside email service avoided this bureaucratic problem. At the beginning of the project the lead institution created a publicly accessible website for the project. On this website initial instructions for getting started on the project, such as how to setup a Dropbox account, were posted. Following initial setup the majority of communication was handled through email.

4 Conclusion and Future Work

This project succeeded in bringing the faculty and students of four academic institutions together to execute a large scale collaborative project while preserving curricular autonomy. The workflow employed followed standards in the

animation industry, however the collaboration did not introduce any special IT requirements that had a significant impact on existing resources. The programs involved and their students are all left-brain/right-brain engaged. They are straddling the division between the art of computer graphics and the science of computer graphics. It is conceivable that this project could have been organized to include participation from students in programs that are either strongly computer science oriented or strongly art and design oriented. It is our opinion that because each location shared responsibility for both technical and aesthetic problems there was a shared empathy and understanding among the student participants and faculty regarding time, resources, knowledge and skills required. We intend to pursue the project further through augmentation of the IT used to maintain synchronous communication among the participants.

References

- [1] Maher, M.L., Simoff, S., Cicognani, A.: *Understanding Virtual Design Studios*. Springer, London (2000)
- [2] Tomayko, J.E.: *Teaching a project-intensive introduction to software engineering*. Software and Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1987, Tech. Rep. SEI-SR-87-I (1987)
- [3] Tsai, P.J., Hwang, G.J., Tseng, J., Hwang, G.H.: A computer-assisted approach to conducting cooperative learning process. *International Journal of Distance Education Technologies* 6(1) (2008)
- [4] So, H.J., Brush, T.A.: Student perceptions of collaborative learning, social presence and satisfaction in a blended learning environment: Relationships and critical factors. *Computers & Educations* 51, 318–336 (2008)
- [5] LaPlante, P.A.: An agile, graduate, software studio course. *IEEE Transactions on Education* 49(4), 417–419 (2006)
- [6] Wilson, J.M., Jennings, W.C.: Studio courses: how information technology is changing the way we teach, on campus and off. *Proceedings of the IEEE* 88(1), 72–80 (2000)
- [7] Hasirci, D., Demirkan, H.: Understanding the effects of cognition in creative decision making: a creativity model for enhancing the design studio process. *Creativity Research Journal* 19(2-3), 259–271 (2007)
- [8] Nova, N., Wehrle, T., et al.: Collaboration in a multi-user game: impacts of an awareness tool on mutual modeling. *Multimedia Tools and Applications* 32, 161–183 (2007)
- [9] Phung, D., Valetto, G., Kaiser, G.E., Liu, T., Kender, J.R.: Adaptive synchronization of semantically compressed instructional videos for collaborative distance learning. *Journal of Distance Education Technologies* 5(2), 56–73 (2007)
- [10] Jara, C.A., Candelas, F.A., Torres, F., Dormido, S., Esquembre, F., Reinoso, O.: Real-time collaboration of virtual laboratories through the Internet. *Computers & Education* 52, 126–140 (2009)

A Workflow Based Process Visual Analyzer (ProVisZer) for Teaching and Learning

Nathaniel Rossol¹, Irene Cheng¹, and Mrinal Mandal²

¹ Computing Science Department, ²Dept. of Electrical and Computer Engineering
University of Alberta

Abstract. Making use of *visualization* in education, integrating graphics, animations and other visual cues, to present curriculum content, in order to inspire and engage learners has become increasingly attractive to educators, curriculum designers, application developers, researchers and administrators. Not only the interactivities available to learners, but also the visual concepts complementing the comprehension of subject topics, have proved successful in enhancing learning performance. In this paper, we introduce a workflow based **Process Visual Analyzer** (ProVisZer), which can be used for self-tutoring, as well as evaluating learning and teaching performance. Based on the recorded data, teachers will understand the skill levels of students, and adjust the curriculum or teaching methodologies accordingly. An authoring tool for defining the work flow behind a curriculum question is implemented with the Analyzer. The proposed system provides an effective visualization tool for education.

1 Introduction

Computing-based education has opened up promising opportunities to develop engaging and motivating curriculum contents. Graphics and animations often provide interactivities and additional expressive power to capture complex ideas. Incorporating more visual components to help improve learning performance has become a popular trend in education. For example, computer games, played by an individual or collaboratively, have been widely used to teach concepts [1], [2], [3]; visual effects are used to assist learner understanding [4], [5]; and virtual laboratories are used to simulate medical and engineering experiments [6], [7]. While visual computing in education enhances student learning capabilities, its full benefits in education have not been adequately explored. While most research focuses on designing visually appealing interfaces or gameplay to motivate students to learn, not much has been developed to assist teachers to visualize their teaching effectiveness, and adapt accordingly.

In this paper, we propose a **Process Visual Analyzer** (ProVisZer), which can be used not only by students as a self-tutoring tool, but also by teachers to assess the progress of a student or a classroom of students. Evaluations of both learning and teaching performance can be obtained using ProVisZer. Instead of evaluating students' understandings of a topic by simply looking at the final answers to complex problems, teachers can trace the students' way of thinking in solving various steps of a question. This is done by examining graphs which depict students' thinking patterns. We apply workflow based methodology to construct these bi-directional graphs

with the flexibility of recording corrective steps performed by students when solving a problem. These corrective steps also provide teachers clues about student skill levels in the corresponding subject area. We use an earlier developed Process Analyzer prototype to create curriculum examples to illustrate the functionalities of the proposed ProVisZer.

The rest of the paper is organized as follows: Section 2 reviews related work including a brief overview of our earlier developed Process Analyzer prototype; Section 3 introduces the proposed ProVisZer, explains how the Process Tree represents the underlying workflow associated with each subject question, and ProVisZer can be used to evaluate performance; and finally the conclusion and future work is presented in Section 4.

2 Related Work

Process analysis is an effective means for analyzing complex, multi-step process-based educational problems. We implemented a prototype of such an analyzer. As the name implies, this prototype records the processes (or steps) students take to solve complex problems and provides teachers useful information to assess student skill levels. Evaluation by examining intermediate steps is more accurate than simply evaluating the final answer. Students solve individual processes associated with a complex problem via a graphical user interface (GUI). The GUI has three major panels: the Graphics Panel displays graphics and animations and the question text to be solved; the Sub-Process Panel displays the different possible steps that can be chosen; and the Working Panel displays input/output boxes where the student can key in text or numbers to compute the required variable. Hints and user instructions, as well as intermediate results are also displayed on the GUI. The proposed ProVisZer is built upon this prototype with a major component – Visual Analyzer, added. ProVisZer also relaxes the question type, which was restricted to only math based questions in the prototype.

A variety of approaches using visualization tools in education have been undertaken and some major ones are discussed here.

2.1 Parson's Programming Puzzles

Parson's Programming Puzzles [8], is similar to ProVisZer in that it is also a teaching and learning system designed to engage students through the use of visualization. Parson's Programming Puzzles also comes complete with an authoring tool that instructors can use to author their own content. Students (who are typically learning programming for the first time) use a single Selective Response Item in the form of a drag-drop Item Type to assemble a working program out of several code segment objects. The system also uses graphics to display the intended function of the system as a UML activity diagram. It allows students to follow incorrect paths for a while, (giving them the opportunity to correct themselves when they realize this) and also supports multiple paths to the correct solution with some paths being better than others (i.e. some make better use of proper coding style and software engineering design

principles). However, as the name suggests, Parson's Programming Puzzles system is solely designed for teaching introductory programming and is not general enough to be applied to another subject area. Furthermore, while students do in fact go through the process of constructing a program incrementally, step-by-step, only the final result is evaluated, and the intermediate steps the students used to construct the solution (including back-tracking and self-corrections) are not recorded. This means that instructors cannot visualize or examine, in any way, the thought process that students used to solve the problems.

2.2 SIETTE

The "System of Intelligent Evaluation using Tests" (SIETTE) [9] is a web-based adaptive testing system that attempts to use Computer Adaptive Assessment (CAA) in order to adaptively assess and evaluate student performance. Instructors populate a question bank with a large selection of questions on the subject that they are testing and define a large number of parameters to help tune how the system operates. These parameters include the difficulty level of the questions, time limits, the length of the test, etc. There is also the option to allow the system to pose as many questions to the student as necessary in order to estimate their knowledge level within a defined level of statistical certainty. One important limitation of the system is that the various parameters for each question must be set manually by instructors. What an instructor believes is the correct difficulty level of a question may not be accurate compared to what students find difficult. Another limitation is that the system assumes each question is independent of the next, which is contradictory to the adaptive principle of CAA, which selects the next question based solely on the student's responses to previous questions. Unlike the design of our ProVisZer, instructors here do not have the ability to decide what question to load next based on a student's skill level.

2.3 AutoTutor

The AutoTutor system as described in [10] is one of the many systems that attempts to learn about a student's psychological state (i.e. Affective State) in order to adapt appropriately to tutor better. For example, if the system believes the user is becoming bored or uninterested, it could take adaptive action by providing a more engaging question next or perhaps switching to a different topic for a short while. The AutoTutor system described in [10] uses natural language and questions in a conversation-based format to mimic a real human tutor. Arrays of visual, audio and tactile sensors focused on the learner are the way in which the system attempts to infer the emotional state of the learner. The version described was implemented to teach both Newtonian Physics and computer literacy. Although ProVisZer was not designed to specifically deal with the emotional or Affective states of the students, it is flexible enough to allow for this data to be inputted into the system when a question is loaded. Developers can implement Process Nodes to provide this data to the system, and Question Type Nodes can be implemented that would use this data to adapt in response to the data regarding the student's affective state.

3 Process Visual Analyzer (ProVisZer)

We have designed and implemented a visualization ProVisZer (Fig. 1) that is capable of modeling any process-based question that can be broken down as a series of simpler sub-processes. Students can visualize concepts expressed in graphics and animations. Also, the system is complete with a Multimedia workflow-based authoring tool that allows instructors to create and preview Multimedia process-based questions, rather than requiring XML code be written by hand. This is a critical design factor because teachers may not have the technical skill or have time to learn such skills due to other teaching responsibilities. It is important for teachers to feel comfortable and in control of the tool when creating and revising the questions otherwise, they may prefer to stay with paper and pencil, or simple fill-in-the-blank and multiple choice type question formats, instead of supporting the more vibrant and motivating visual computing elements.

ProVisZer uses a modular approach. The process-based problem is built-up as a series of Item Type nodes (both Selective Response Items and Constructive Response items) and branches differently depending upon student responses. Not all nodes in a process question are a Multimedia Item Type Node, however. Other nodes (named Process Nodes) exist which perform various other tasks such as processing data, displaying information to the user, and providing adaptive behavior. In this sense, Item Type nodes are simply one of the many types of Process Nodes that make up a process question. ProVisZer begins at the “Start” Node and proceeds to the next Process Node that it is connected to, branching if appropriate according to student responses or other conditions. Process Nodes (including Multimedia Item Type Process nodes) are able to accept data in the form of input parameters and also output data to pass it to another Process Node. In this manner, Item Type Nodes can pass along any data they wish to each other. Item Type Nodes are designed using a plug-in style interface approach. This allows brand new item types to be implemented and integrated into the system with ease. A programmer needs only create an Item Type class that implements the provided abstract class interface methods, and will be available for use immediately.

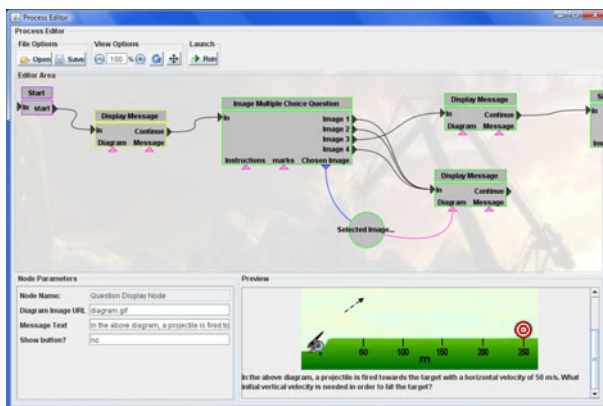


Fig. 1. ProVisZer with a sample question loaded. Teachers can modify individual questions by selecting them and adjusting their parameters inside the parameter list. They can also test out the questions by clicking the “Run” button.

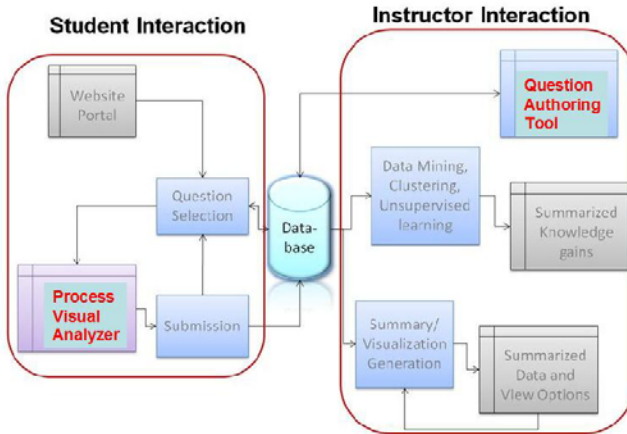


Fig. 2. ProVisZer is designed to fit into a general framework for Computer Adaptive Assessment

A novel feature of our ProVisZer is when a student solves a question, their solving process is visualized and displayed in a tree format. This Process Tree visualization is interactive in the sense that the student can click on any node of the tree to restore a previous process state. This essentially allows a student to “backtrack” if they believe they have made a mistake on a previous step, or to switch back to a different path if they believe it might have been a simpler approach to solving the problem. The Process Tree is submitted to the server along with the student’s answer so that instructors can inspect a student’s process tree and visually identify where a student (or group of students) ran into difficulties whilst attempting to solve the problem. The Process Tree can also help give instructors insight into the decision-making process a student used when attempting to solve the problem.

ProVisZer is capable of both online and offline use. In the case of offline use, it can be used for tutoring purposes and will read the problems from .xml files off the local hard disk. In the case of online use, ProVisZer can run inside a web browser and can receive the question XML data from a database. The student solutions and Process trees are submitted to the server to be evaluated and visualized using a visualization module. A general framework integrating ProVisZer is suggested in Fig. 2 which shows students interact with the system by answer problems through ProVisZer. Once a problem is completed and submitted by the student, a potentially adaptive server-side process updates the database and decides which problem to retrieve next and give to the student. The right hand side of the diagram illustrates how instructors can use more sophisticated data visualization software to further analyze the collected data. The potential also exists to use data-mining, clustering, and/or Unsupervised Machine Learning techniques to try gaining useful knowledge regarding re-occurring patterns in student achievement regarding the problems. In this paper, we focus on the analyzer and authoring components.

3.1 Workflow Based Design

Process-based problems often have multiple branches all leading to a correct solution. This is one of the main rationales behind the workflow-based design. As discussed in [11] workflows are an ideal way to model processes in general and user interactions in particular. Fig. 3, shows that ProVisZer can branch differently depending upon the choices a student makes whereas most computer-based tutoring systems allow only a single path to the correct answer.



Fig. 3. A Sample Question in which the problem branches into two possible correct paths to the solution. The green colored box-shaped nodes represent Question Nodes whereas circles represent data nodes.

Each Question Item Node in ProVisZer has the ability to branch differently depending upon the student's response. The system can also branch based on some logic described by the author (for example, comparing the current response with a past one). It is important to note, though, ProVisZer is not modeling an FSM (Finite State Machine) because in our case the state is not dependent solely upon which node in the Question the system is currently on. Process Nodes inside these workflow-based Questions are capable of saving data in variables for use by other nodes later in the process. The combination of the current state and all saved variables together define a state in ProVisZer. It is also important to note that our Process Tree can be DAGs (Directed Acyclic Graphs) containing cycles. This feature allows the system to provide corrective action as needed. For example, if a student branches away from any of the defined paths that can lead to the correct answer, the instructor authoring the Question can have the system branch to a "Hint node" that will display a corrective hint. After the hint is displayed, the instructor can have the workflow loop-back to the Process node the student made the mistake on. This feature would naturally be ideal for use in the system's Tutorial Mode.

3.2 The Process Tree and Data Visualization of Student Interaction

As the authors of VisTrails [12] explain, data without understanding is meaningless. Data needs to be interpreted in order to be of any use to humans and data visualization

is seen as an effective means by which to summarize data for easy interpretation. Effective data visualization can give the user an accurate overview of the data at a glance, whilst providing more in-depth details as needed. ProVisZer achieves such goal through the implementation of Process Trees.

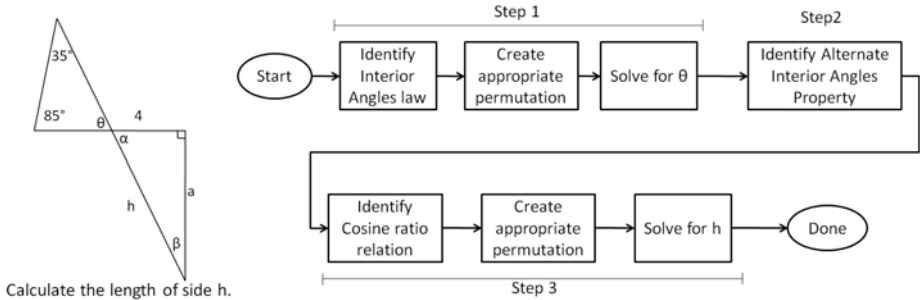


Fig. 4. (Left) A simple geometry problem. (Right) A workflow representing the sample solution to the geometry problem.

As mentioned previously, ProVisZer records every action a student takes in solving a problem, including any dead-end branches where the student deviated from any of the correct possible paths. Consider a trivial geometry problem (Fig. 4 (left)) as an example. The student might solve this problem using the following steps:

Step 1: The Interior angles of a triangle always sum up to 180 °, therefore:

$$\begin{aligned}
 35 + 85 + \theta &= 180 \\
 \theta &= 180 - 35 - 85 \\
 \theta &= 60
 \end{aligned}$$

Step 2: Opposite interior angles are always equal, therefore:

$$\begin{aligned}
 \alpha &= \theta \\
 \alpha &= 60
 \end{aligned}$$

Step 3: And finally, because we are dealing with a right angle triangle:

$$\begin{aligned}
 \cos(\alpha) &= 4/h \\
 h &= 4 / \cos(\alpha) \\
 h &= 8
 \end{aligned}$$

In this example, the student broke the problem down into three smaller sub-problems, namely: finding θ , finding α , and finally solving for h . Steps 1 and 3 both had three sub-steps each, namely: identifying the appropriate law to use (and the appropriate permutation of that law), substituting in the appropriate values, and finally solving the resulting expression. Step 2 simply involved recognizing the correct geometry law.

Given that each step (and sub-step) was dependent upon the step before it (as is typical of these types of problems) it is natural to model it as a workflow. A possible model is shown in Fig. 4 (right).

However, in the geometry question, there exists another route to the final solution that the student may take. Suppose that the student is not aware of the relationship between the cosine of an angle and the sides of a right-angle triangle, the student may instead use tangent relationship to compute the length of side 'a' instead after step 2:

$$\begin{aligned} \tan(\alpha) &= a/4 \\ a &= 4 * \tan(\alpha) \\ a &\approx 6.928 \end{aligned}$$

Afterwards the student could use the Pythagorean Theorem to compute the length of side h:

$$\begin{aligned} h^2 &= a^2 + b^2 \\ h &= \sqrt{((6.982)^2 + 4^2)} \\ h &= 8 \end{aligned}$$

If we call the previous solution A and this latter solution B, a workflow that accounts for both of these acceptable paths to solve the problem might look like Fig. 5.

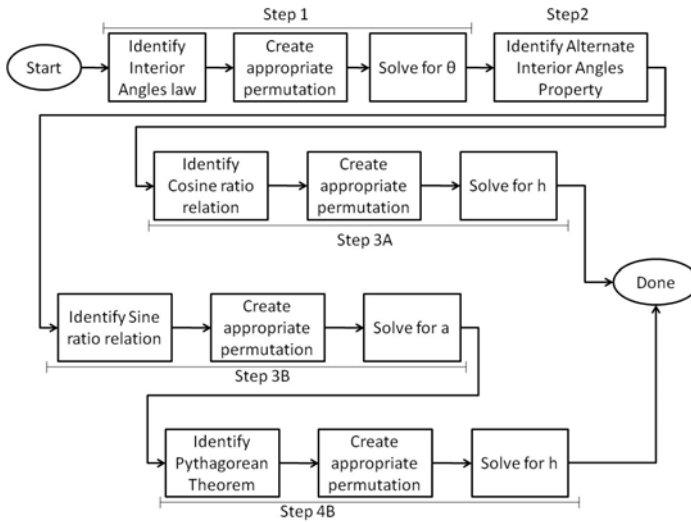


Fig. 5. A workflow that accounts for both possible means by which to solve the problem. The paths branch after step 2 depending upon the way the student wishes to solve the problem.

Now suppose student A completed each step flawlessly, Student B completed steps 1 and 2 flawlessly, but was not able to immediately identify all the variables needed to complete the cosine relationship, so instead used the sine relationship and the Pythagorean Theorem. Also consider a third student, C, who first tried to apply the Sines law in step 1 and then the Cosines law before finally realizing the need to use the Interior Angles Law. Using a tree format, we could model each of the student’s problem solving processes. As we can see in Fig. 6, student A completed one step after another without any issue and thus has no branching. Student B started on one approach for step 3, but became “stuck” at 3.1 and abandoned it. The student then took a different approach to step 3 which led to a fourth step before arriving at the final answer. Student C attempted three strategies for the first step before finally settling on one that worked. From this visualization, much can be gained at an initial glance. Instructors can easily recognize that student with little or no branching in their Process Tree obviously had little difficulty with the problem. However, students with a great deal of branching (i.e. Process Trees with greater “breadth”) clearly had much

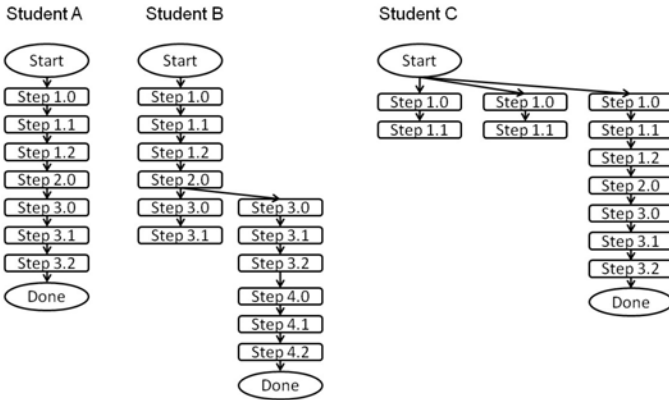


Fig. 6. A Process Tree visualization of three students completing the geometry problem. The nodes indicate the state of the student (i.e. which Question sub-step they are on).

more difficulty in solving the problem. In paper-and-pencil style exams, such repeated failed attempts at solving a step would often be erased by students and lost. However, a great deal can be learned from failed attempts, and as such, ProVisZer records every action and attempt that the student takes.

We believe that the Process Tree visualization is not only useful to instructors after the answer is submitted, but also of great use to the student whilst they are solving the problem. In ProVisZer, the Process Tree is dynamically generated as the student works on the problem. Each node in the tree represents a state the student was in for a certain step. Students are free to return to any past state they were in simply by clicking on a node in the Process Tree. The Process Tree Visualization module will then create a new branch as the student continues solving from that state. This way, no student work is ever lost; if a student abandoned an approach but then later realized that it was the correct approach, the student can click on that state left off and continue working from there.

Finally, after the student has completed the question, the entire Process Tree is submitted to the server for automated assessment by server-side scripts. If a CAA module is implemented on the server, the next question will be selected adaptively. Instructors can then use data visualization tools to visualize the Process Trees of a single student or perhaps summarized Process Tree data from the entire class. For example, if step 1.0 had thirty branches coming from it but step 2.0 had eighty, it would indicate to the instructor that students are struggling a great deal with either the concepts tested for in step 2.0, or perhaps the way the question is worded or set up. The instructor can quickly narrow-in on these issues at a glance and take corrective action immediately.

4 Conclusion

We propose a Process Visual Analyzer (ProVisZer) to support visual computing in teaching and learning for complex problems involving multiple intermediate steps to

arrive at a final solution. By modeling such process-based questions as workflows, and by creating an interactive visual feedback mechanism via the Process Tree visualization, users of the system can visualize the decision path(s) and adapt accordingly. The system allows corrective actions navigating backwards or forwards through the Process Trees. ProVisZer is not only an effective self-tutoring tool for students, but also an efficient mechanism for teachers to assess the performance of a student, or a class of students. The assessment results also help teachers to judge whether curriculum or teaching methodologies need to be adjusted. Future potential exists for integrating ProVisZer with large scale learning and management systems such as Moodle, LAMS, and SCORM.

References

1. Feng, K.C., Chang, B., Lai, C.H., Chan, T.W.: Joyce: A Multi-Player Game on One-on-one Digital Classroom Environment for Practicing Fractions. In: Proc. of the 5th IEEE Int. Conf. on Advanced Learning Technologies (ICALT 2005) (2005)
2. Galvao, J.R., Martins, P.G., Gomes, M.R.: Modeling Reality with Simulation Games for a Cooperative Learning. In: Proc. of the Winter Simulation Conf., pp. 1692–1698 (2000)
3. Mitchel, A., Savill-Smith, C.: The Use of Computer and Videogames for Learning. Learning and Skills Development Agency (2004)
4. Lyvers, R.F., Horowitz, B.R.: A Unique Instructional Tool for Visualizing Equipotentials and its Use in an Introductory Fields Course. *IEEE Trans. on Education* 36(2), 237–240 (1993)
5. Stubbs, K.: Kana no Senshi (Kana Warrior): A New Interface for Learning Japanese Characters. In: CHI, pp. 894–895 (2003)
6. Duarte, M., Butz, B., Miller, S., Mahalingam, A.: An Intelligent Universal Virtual Laboratory (UVL). *IEEE Trans. on Education* (2007)
7. Hernandez, A., Mananas, M., Costa-Castello, R.: Learning Respiratory System Function in BME Studies by Means of a Virtual Laboratory: RespiLab. *IEEE Trans. on Education* (2007)
8. Parsons, D., Haden, P.: Parson’s Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In: Proc. of the 8th Australian Conf. on Computing Education, vol. 52, pp. 157–163 (2006)
9. Guzman, E., Conejo, R., Perez-de-la-Cruz, J.: Improving Student Performance Using Self-Assessment Tests. *IEEE Intelligent Systems Magazine* 22(4), 46–52 (2007)
10. D’mello, S., Picard, R., Graesser, A.: Toward an Affect- Sensitive AutoTutor. *IEEE Intelligent Systems Magazine* 22(4), 53–61 (2007)
11. VisTrails available online at <http://www.vistrails.org>
12. Callahan, S., Freire, J., Santos, E., Scheidegger, C., Silva, C., Vo, H.: Managing the Evolution of Dataflows with VisTrails. In: IEEE Workshop on Workflow & Data Flow for Scientific Applications 2006 (2006)

Teaching Geometric Modeling Algorithms and Data Structures through Laser Scanner Acquisition Pipeline

S. Gueorguieva¹, R. Synave¹, and Ch. Couture-Veschambre²

¹ UMR5800, Laboratoire Bordelais de Recherche en Informatique

² UMR5199 PACEA, Laboratoire d'Anthropologie des Populations du Passé, Université Bordeaux1, France

Abstract. Experience from geometric modeling course based on a specific teaching medium, namely trochlear surface reconstruction from laser scans, its evaluation in terms of shape feature measurements and finally its instantiation through 3D printing, are presented. Laser scanner acquisition, reconstruction and 3D printing lend well to teaching general concepts in geometric modeling for several reasons. First, starting and ending with real physical 3D objects (the talus and its 3D print) provide in addition to the classical visual feedback a material feedback for correctness of treatments all over the pipeline. Second, the notion of error during each step of the pipeline is illustrated in a very intuitive way through length measurements, manual ones with callipers on the tali, and numerical ones with arc and chord lengths on the numerical reconstructions. Third, students are involved with challenging scientific problems and produce semester-long projects included in larger scaled project of cultural heritage preservation. Our believe is that this approach gives a deeper understanding of both theoretical and application issues in geometric modeling.

Keywords: geometric modeling, digital paleoanthropology, cultural heritage preservation, laser scanner acquisition, image registration, image reconstruction, 3D printing.

1 Introduction

Geometric modeling evolves into a great variety of visual computing applications as 3D shape matching and recognition, medical image analysis [1] and cultural heritage preservation [2,3,4,5,6]. Geometric models capture spatial aspects of the objects of interest for an application [7,8,9] and necessitate rigour mathematical foundations [10,11,12,13,14,15,16].

Unfortunately, the mathematical prerequisites are often unappealing to college students and the visual representation of 3D objects is usually considered as a final goal. Often, facet (planar face) models correspond to “soup” of polygons with degenerate polygons, holes and self-intersecting faces. Such models could be acceptable for rough graphical representation but when looking for surface feature estimation as the surface normal for example, these models lead to

erroneous calculation. A possible way to handle such irregularities is to locate artefacts through geometric intersection tests [17][18], to refine the underlying mesh in singular points and then to produce a manifold surface subdivision. These treatment is a commonly employed operation when integrating different scan views in a laser scanner acquisition based surface reconstruction. The advantage in making use of the laser scanner acquisition pipeline is that theoretical notions as *r - sets* and *3D manifolds*, topological invariants and Euler characteristics, have an imminent impact in the results. Another example of geometric model prevailing property is the validity to argue if the model corresponds to (at least) one object. In the terms of the three level hierarchical view of modeling [9][19], “physical object \rightarrow mathematical models of objects \rightarrow representations of objects”, a valid geometric model could correspond to a mathematical model but not to a physical object. A new intuition on validity is brought to an active state through the 3D printing. The pipeline supports a three level graph view, “physical object \leftrightarrow mathematical models of objects \leftrightarrow representations of objects”. Once the object boundary surface is reconstructed, a STL file [20] could be produced and a synthetic physical object reproduction, a 3D print, could be output. The distinction between valid and “printable” geometric model allows to close up the cycle “physical object \leftrightarrow mathematical models of objects \leftrightarrow representations of objects \leftrightarrow physical object”. Each level of these modeling view is clearly identified through the pipeline and practitioner students can themselves evaluate the quality of all intermediate results and the final product. The control is done through manual measurements on the physical supports (bone specimen or its 3D print) and through numerical measurements on the numerical object reconstruction. The purpose of the present research is to show our experience in how to teach the algorithms and data structures underlying the pipeline. Our objective is to emphasize the practice interest of geometric modeling theoretical foundations for both computation implementation and anthropology investigations. Our believe is that this approach will engage students in learning geometric modeling through the development of solutions to scientific relevant problems. In the following, the main steps are explicated with the corresponding accompanying references. This program is proposed as a MS degree course of geometric modeling in the Department of Computer Science at the University of Bordeaux

1. Examples of the student project final results are also provided.

2 Laser Scanner Acquisition, Registration and Integration

There are several examples of 3D model acquisition pipelines [21][23][5][22][23][24]. In our case, the acquisition is done by a non-contact 3D digitizer VIVID 300/VI-300 with a laser wavelength $\lambda = 690nm$ and maximal optical power $P_{Max} = 7mW$, object length range is in the limits of $[0.55m, 1.2m]$, field of view is in the limits of $[185mm, 395mm]$ and output data points 400×400 . An illustration of talus acquisition with VIVID 300/VI-300 and Polygon Editing Tool is given in Fig. 1. Two views of the talus in Fig. 1(left) are registered in Fig. 1(center).

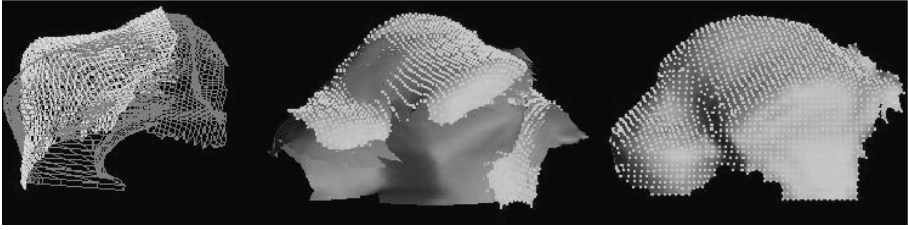


Fig. 1. Polygon Editing Tool: (left) Two scans (center) Scan registration (right) Scan integration

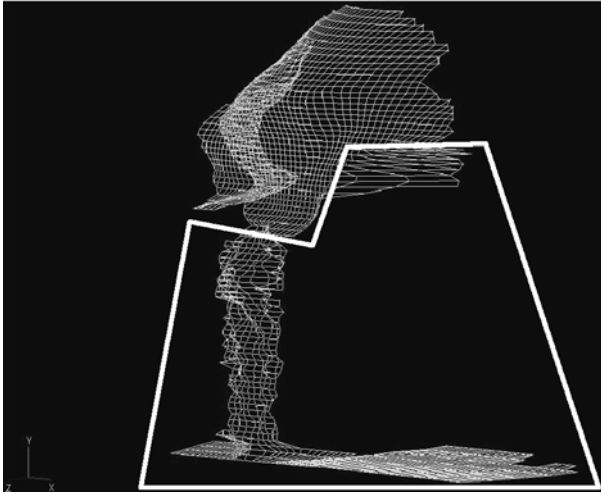


Fig. 2. Polygon Editing Tool: erroneous acquisition

It should be noted that the rough acquisition for a view corresponds to the mesh model shown in Fig. 2. The human operator is supposed to clean up those parts that are out of the region of interest and to produce the model given in Fig. 1(left). The final step is the integration of all scans and bringing forward for consideration the reconstructed surface as shown in Fig. 1(right). Acting with the system for a short time is sufficient to identify crucial points as:

- The optimal values for the laser power (% of P_{Max} ?) and the object distance. A few protocols including technical data sheets for the laser scanner measurements are known [25].
- The initial alignment and the partial overlap between sequential scans. The widely used in range image registration Iterative Closest Point(ICP) algorithm and its derivatives [26,27,28] strongly depend on both characteristics.
- The extent and the constraints to interpolate and/or approximate cloud points during the integration. Using Polygon Editing Tool for example, one can easily observe surface artefacts as undesired smoothing and lost of details in the reconstructed surface.

Along this first stage in the training program, students are mostly supposed to act as final users of industrial software. As long as such software is a “black box” no technical details are provided and thus personal investigations are encouraged in parallel with an introductory course in geometric modeling fundamentals.

3 Object Boundary Reconstruction and Evaluation

3.1 Programming Project

At this period, students start with a programming project dealing with surface reconstruction from 3D range images [29,30] obtained through the first stage. The range images could represent individual or integrated views of the studied objects and the goal is to construct valid surface mesh models. The following requirements are implied:

- The data structures in used [31] should support efficient topological queries as variable sized vertex neighbourhood construction referring to the quad-edge [32], the winged-edge [33,34] or the half-edge [35] data structures, and evaluated boundary representation for rapid visualisation;
- The models ought to fulfil mesh element quality criteria [36] and geometric coherence of data [37].

In order to broach the geometric computations students are urged to experiment with examples of robustness problems [38,39] and in particular the *Core* [40] and *LEDA* [41] libraries. Along with geometric intersection tests essential in quality mesh improvement, a set of mesh query based operations are also required:

- Calculation of Euler characteristics of the underlying object boundary surface [42,43];
- Geodesic path construction between a pair of source and destination vertices [44,45,46,47].

Finally, as an optional functionality, the discrete curvature estimation [48] at mesh model vertices is required. This feature is helpful to indicate the characteristic vertex corresponding to chosen landmark as long as it is often situated either on spherical or hyperbolic point. In case when this functionality is not implemented the choice is made by visual appreciation.

These programming projects are developed as a collaborative work within two or three person team. Weekly, each team present the advancement of the project. The implementation is done in *C++* using either *Qt4* or *GTK* graphical user interface.

3.2 Validation

Once the student software becomes operating the validation of the boundary surface reconstruction starts with geometric features evaluation. First, manually on the studied specimen and then by simulating the same measurements on the numerical representation.

For our experience we study osteological specimens issued from an archaeological series emerging from the “Soeurs Grises”’s cemetery (medieval sample,

XV-XVIII° century) located in Beauvais (Oise, France). In particular we are interested in the (trochlear) articular facet on the superior surface of the talus's body involved in the ankle joint and consequently in the foot position during locomotion [49]. An illustration is given in Fig. 3.

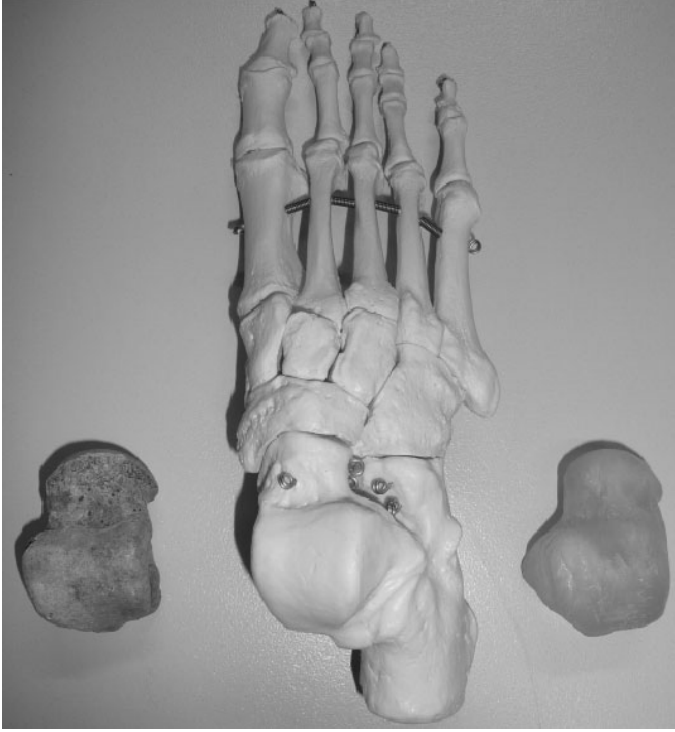


Fig. 3. Left talus: (a) Superior view (b) Anatomical position (c) 3D print

Three pairs of landmark points for the trochlear shape are defined following [50, 51, 52]:

$(P_{apmr}, P_{aplr}), (P_{ipmr}, P_{iplr})$ and (P_{ppmr}, P_{pplr}) . Each pair corresponds to one position: the anterior, the intermediate and the posterior with respect to the frontal plan. A point of the pair is either on the medial or on the lateral process of the trochlea with respect to the sagittal plane, denoted as medial or lateral ridge as illustrated in Fig. 4. Anthropometric chord and arc length measurements are performed manually with calliper and millimetre ruler band on the specimens.

For the validation of the trochlear surface reconstruction each manual measurement is replicated on the mesh model. Chord measurements are similar to euclidean distances between the pair of characteristic vertices, each one corresponding to an anatomical landmark point. Arc measurements agree with geodesic distances between the characteristic vertices. Statistical analysis [53] is performed over all samples in order to evaluate the range interval for the

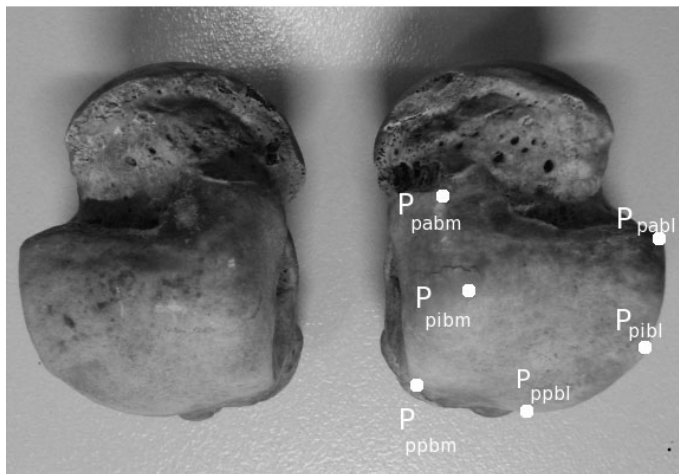


Fig. 4. Anatomical landmark points on the trochlear surface

standard deviations for the measured entities. All results are presented and appreciated by an expert in the application field namely an anthropologist. According to our experience these discussions are particularly fruitful as long as the visual and the numerical feedback of the trochlear surface reconstruction are commented by a potential final user of such kind of software packages. In this way the developer deals with the technical trade specifications.

4 Offsetting Object Boundary and 3D Printing

Creating of 3D print models becomes very popular as research and training tool [54,55,56]. The 3D printer put to student's disposal is an Objet Eden250 3D printing system. The input of the 3D printing system is the STL format [20] that does not guaranty the correctness of the underlying surface namely to correspond to an oriented 2D manifold without boundaries. The result of the reconstruction on the previous stage is a triangulated surface but with possible existing boundaries as for example when the object boundary is partially reconstructed and does not enclose a bounded volume (shell surface). For such cases initial surface should be offset [57,58]. The offset direction is chosen opposite to the normal direction in order to preserve the chord and arc metric of the initial shell surface. The offsetting and the STL output are added as new functionalities to the programming project. Finally, each student team produces and evaluates the 3D print of the studied talus specimen. Evaluating the precision of the complete pipeline from the acquisition through the reconstruction to the 3D printing is an innovative approach to validate the whole treatment channel and the supported data structure and algorithm implementations. A few similar works are known related to vision system based on computed tomography and commercial software [59]. According to the technical specifications the typical tolerance of

Eden250 is of $0.1mm$. The student experimental results show that the variance of manual measurements on the 3D print is similar to the one of manual measurements on the original specimen but that the absolute values of chord and arc lengths increase for the 3D print. It is observed that the interchangeability of both supports is limited to a resolution of $1.5mm$.

5 Discussion and Experimental Results

The elaborated training program is successfully applied as a graduated course in the master of Computer Science at the University of Bordeaux 1. It should be noticed that mathematical prerequisites and namely notions from algebraic topology of surfaces and differential geometry are difficult to apprehend as long as in the previous Computer Science degree courses students do not broach these subjects. Often, problems are identified but the mathematical apparatus to resolve them takes too long time to become familiar with enough in order to be used. Dealing with the laser scanner acquisition pipeline as a whole permits to remain flexible with respect to the pedagogical objectives that can be adapted according to individual skills of the members in a programming project team.

For the basic pipeline processing steps:

registration, reconstruction, offsetting, geodesic path computation and curvature estimation, rough algorithms are proposed as initial guess to be improved as students go along. See for example, in Fig. 5(b) the illustration of the shortest path calculation between the source (P_{pabl}) and the destination (P_{ppbl}) characteristic vertices. The rough algorithm is the Dijkstra algorithm. Very quickly a shortest path computation could be implemented and visualised but the limits of a naive implementation are obvious even for mesh models with a total number of vertices under the range of 10^4 . Further, the problem of non-uniqueness of the shortest path and the lack of a measurement direction as in physical measurements with calliper or millimetre ruler band could be seen in Fig. 5(a). Finally, the choice of the path extremity in a specified mesh vertex position could be tedious and erroneous task so that the possibility of interaction with vertices in a variable sized neighbourhood of source (destination) vertex is useful as shown in Fig. 5(c).

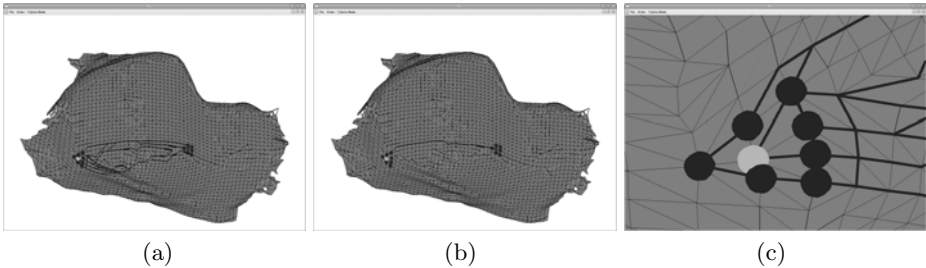


Fig. 5. M. Fichoux and D. Sabary's programming project : geodesic path calculation

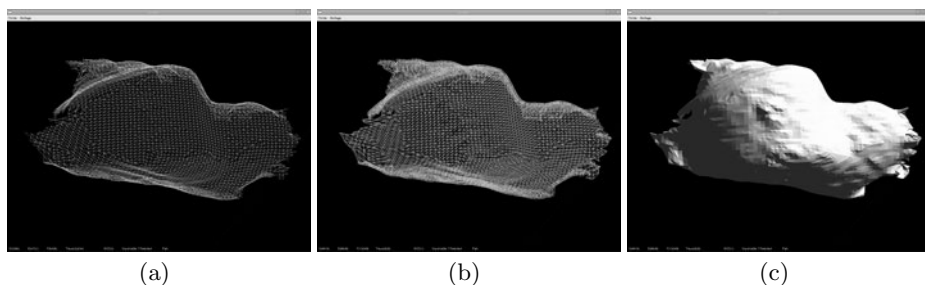


Fig. 6. Ch. Mary and Ch. Delmotte's programming project : offsetting for 3D printing

The offsetting of a mesh model from Fig. 6(a) is illustrated in Fig. 6(b) and Fig. 6(c). The rough algorithm consists in a vertex based translational extrusion. As long as for 3D printing of shell surface we need a narrowly banded extruded surface, the offset vector is of small magnitude and in almost all tested models self-intersections are avoided.

According to our experience discrete curvature evaluation stands up as one of the most apprehended problem. Starting with [60] and the *GSL* GNU Scientific library for matrix operations, an initial evaluation could be reported as shown

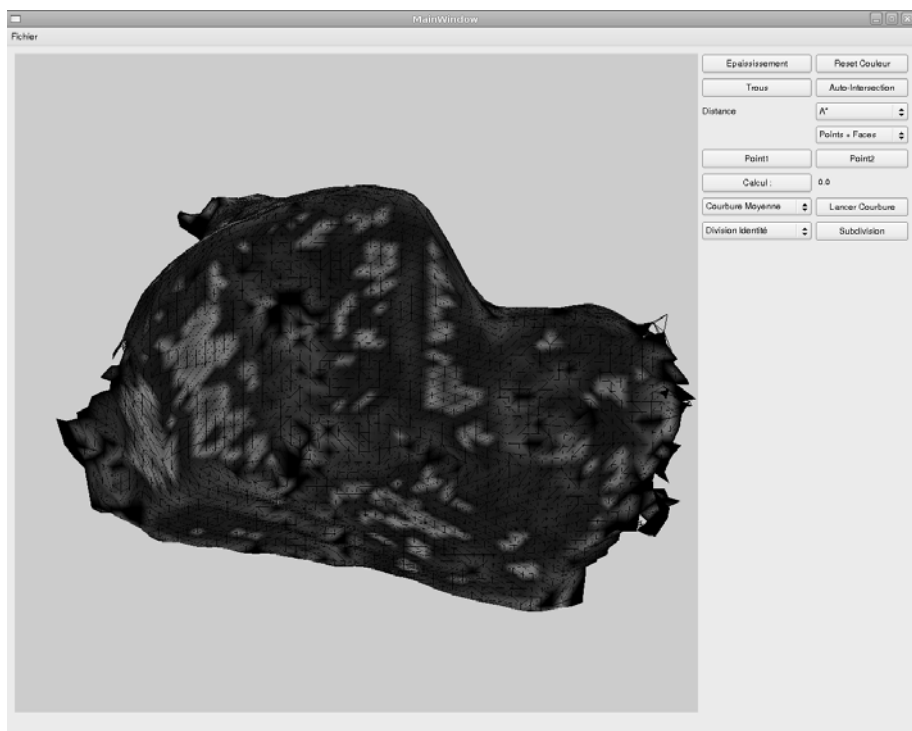


Fig. 7. J. Laviolle and R. Leguay's programming project : discrete surface curvature evaluation

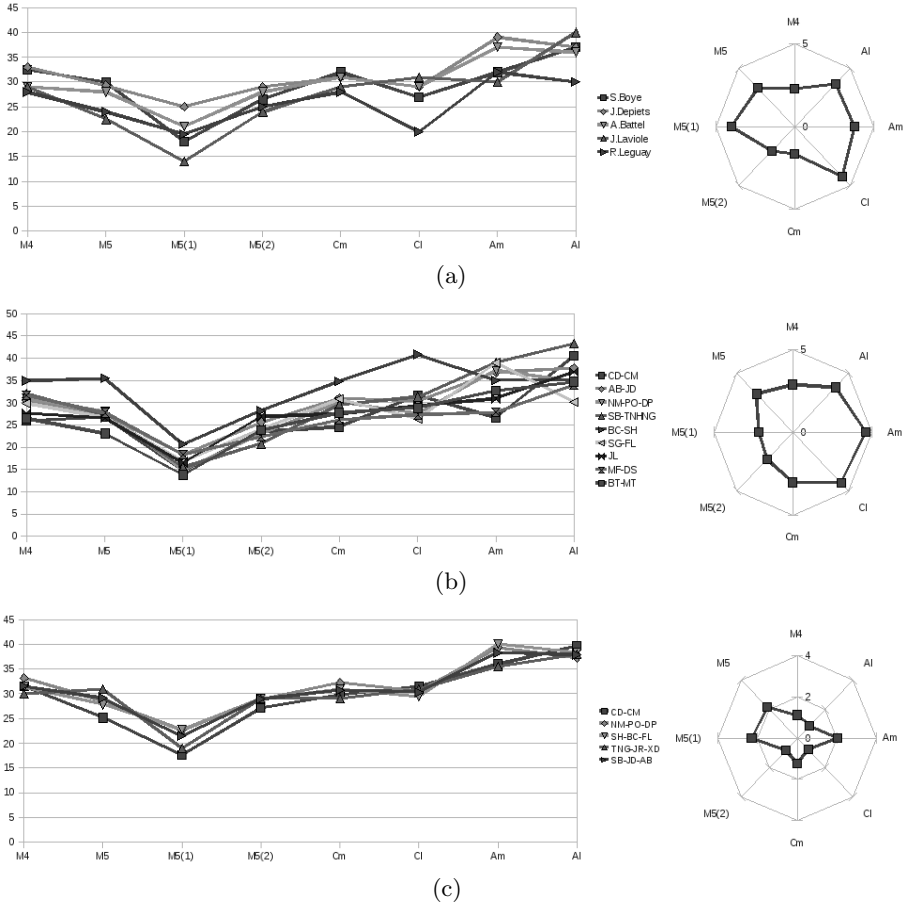


Fig. 8. Chord and arc measurements on BCT92-S380-G: (a) Original specimen (b) Numerical reconstruction (c) 3D print

in Fig. 7 where flat like regions are displayed in black colour and the curved ones in different graduations of the grey colour.

There are two checkup points for the developed software:

First, evaluate the reconstructed object boundary in terms of chord and arc measurements between characteristic vertices on the underlying mesh model, and second, estimate the same geometric features on the corresponding 3D print. The computed values are compared with the manual measurements on the original specimen. The goal is to establish the degree of interchangeability along the different object reproductions, the numerical representations and/or the synthetic ones. The results are summed up for the specimen the BCT92-S380-G, for the original individual in Fig. 8(a)(a), for the numerical reconstruction in Fig. 8(b)(b), and for the 3D print in Fig. 8(c)(c). In the figure, on the left side, computed values are provided. On the right side in the figures, the standard

deviation distribution for the sampled values is shown. The notations in use are as follows:

- The length of the trochlea, $M4$, is the length of the chord joining the crossing points of the anterior and the posterior ridges with the medial lengthwise curvature of the trochlea;
- The width of the trochlea, $M5$, is the length of the chord $P_{ipmr}P_{iplr}$ joining the medial and the lateral ridges;
- The posterior width of the trochlea, $M5(1)$, is the length of the chord $P_{ppmr}P_{pplr}$ joining the medial and the lateral ridges;
- The anterior width of the trochlea, $M5(2)$, is the length of the chord $P_{apmr}P_{aplr}$ joining the medial and the lateral ridges;
- The medial chord of the trochlea, Cm , is the chord length between P_{ppmr} and P_{apmr} ;
- The lateral chord of the trochlea, Cl , is the chord length between P_{pplr} and P_{aplr} ;
- The medial arc of the trochlea, Am , is the length of the curve joining P_{ppmr} and P_{apmr} along the medial ridge;
- The lateral arc of the trochlea, Al , is the length of the curve joining P_{pplr} and P_{aplr} along the lateral ridge.

These chord and arc measurements provide the support for the morphometric analysis performed at the reconstruction and the prototyping stage of talus repository reproduction.

It could be seen that range intervals of the standard deviations decrease for measurements on the 3D prints. In fact, evaluate original specimen with partially damaged ridges is subjective depending on the human operator interpretation. While, after the reconstruction, the corresponding parts on the boundary are “repaired” and thus the positioning of the characteristic vertices is less uncertain. Moreover, range intervals of the chord and arc lengths on the tali tend to increase for their counterparts on the 3D prints in average with $1.5mm$. On the contrary, length estimations on numerical models oscillate around manually measured values within $1mm$ absolute value interval. One can conclude that the dominant of the error in the pipeline occurs during the reconstruction and consequently further improvement should be supplied.

6 Conclusion

The present work relates our experience in teaching geometric modeling fundamentals through the laser scanner acquisition based pipeline and with a particular application domain of cultural heritage preservation. We choose to cover the pipeline as a whole starting with the original specimen, through the acquisition, the registration and the reconstruction of the object boundary, and finally, printing it. The major advantage of this approach is that in addition to the classical visual feedback in the visual systems, a material feedback is also produced and in this way the correctness of treatments all over the pipeline is evaluated. The notion of error is illustrated in a very intuitive way through length measurements:

manual ones with calliper and millimetre ruler band on the tali and their 3D prints, and numerical ones on the numerical reconstructions. All over the covered subjects, students are involved with challenging scientific problems looking for efficient computation solutions. Our believe is that this approach gives a deeper understanding of both theoretical and application issues in geometric modeling.

Acknowledgements. This work was supported by our student classes during the autumn semesters of the years 2008-09 and 2009-10, and with a particular contribution of Ch. Delmotte, Ch. Mary, N. Mellado, B. Orsini, D. Palanchon, S. Boye, T.N.H. N’Guyen, J. Laviole, G. Simon, F. Lepretre, S. Damien, M. Fichoux, B. Tricoire and M. Troale.

The authors wish to thank Mr. Patrice Courtaud for providing the collection from Pessac osteological repository, and Mr. J.-M. Femolan in charge of the complete collection from Archaeological Department of Beauvais city council.

References

1. Ayache, N.: Epidaure: A research project in medical image analysis, simulation and robotics at inria. *IEEE Trans. Med. Imaging* 22, 1185–1201 (2003)
2. Levoy, M.: The digital michelangelo project. In: *3DIM*, pp. 2–13 (1999)
3. Bernardini, F., Rushmeier, H.E.: The 3d model acquisition pipeline. *Comput. Graph. Forum* 21, 149–172 (2002)
4. Bernardini, F., Rushmeier, H.E., Martin, I.M., Mittleman, J., Taubin, G.: Building a digital model of michelangelo’s florentine pietà. *IEEE Computer Graphics and Applications* 22, 59–67 (2002)
5. Rusinkiewicz, S., Hall-Holt, O.A., Levoy, M.: Real-time 3d model acquisition. In: *SIGGRAPH*, pp. 438–446 (2002)
6. Coqueugniot, H., Couture, C., Dutailly, B., Gueorguieva, S., Desbarats, P., Synave, R.: Range image processing for paleoanthropology heritage preservation. In: *Proc. of the 2nd IEEE International Workshop on Digital Media and Its Applications in Museum and Heritages, DMAMH 2007, Chongqing, Chine, December 10-12 (2007)*
7. Requicha, A.: Representation for rigid solids: Theory, methods and systems. *ACM Computing Surveys*, 437–464 (1980)
8. Requicha, A., Rossignac, J.: Solid modelling and beyond. *IEEE Computer Graphics & Applications*, 31–44 (1992)
9. Requicha, A.: *Geometric Modeling: A First Course*. Copyright 1996 The University of Southern California (1999)
10. Mortenson, M.: *Geometric Modeling*. John Wiley & Sons, Chichester (1985)
11. Faux, I., Pratt, M.: *Computational Geometry for Design and Manufacture*. Ellis Horwood (1979)
12. Farin, G.: *Gurves and Surfaces for Computer Aided Design*. Academic Press, London (1988)
13. Fomenko, A., MATEEV, S.V.: *Algorithms and Computer Methods for Three-Manifolds*. Kluwer Academic Publishers, Dordrecht (1997)
14. Farin, G., Hoschek, J., Kim, M.S.: *Handbook of Computer Aided Design*. North-Holland, Amsterdam (2002)
15. Agoston, M.: *Computer graphics and geometric modeling: mathematics*. Springer, Heidelberg (2005)

16. Agoston, M.: Computer graphics and geometric modeling: implementation and algorithms. Springer, Heidelberg (2005)
17. Coxeter, H., Greitzer, S.: Geometry revisited. Mathematical Association of America (1967)
18. O'Rourke, J.: Computational geometry in C, 2nd edn. Cambridge University Press, Cambridge (2001)
19. Hoffmann, C.: Geometric and Solid Modeling: An Introduction. Morgan Kaufmann, San Francisco (1989)
20. Szilvasi-Nagy, M.: Analysis of stl files. Mathematical and Computer Modeling, 945–960 (2003)
21. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH 1996: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 303–312. ACM Press, New York (1996)
22. Jaeggli, T., Koninckx, T.P., Gool, L.V.: Online 3d acquisition and model integration (2003)
23. Pauly, M., Mitra, N.J., Giesen, J., Gross, M.H., Guibas, L.J.: Example-based 3d scan completion. In: Symposium on Geometry Processing, pp. 23–32 (2005)
24. Synave, R., Desbarats, P., Gueorguieva, S.: Toolkit for registration and evaluation for 3d laser scanner acquisition. In: Proc. of the 16th International Conference in Central Europe on Computer Graphics, Visualisation and Computer Vision 2008, WSCG 2008, Plzen, Czech Republic, Plzen, Czech Republic, pp. 199–204 (2008)
25. Synave, R.: Reconstruction de solides à partir d'acquisitions surfaciques. PhD thesis, Université de Bordeaux1 (2009)
26. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14, 239–256 (1992)
27. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: 3DIM, pp. 145–152 (2001)
28. Synave, R., Desbarats, P., Gueorguieva, S.: Automated trimmed iterative closest point algorithm. In: Proc of the 3rd International Symposium on Visual Computing, ISVC 2007, Lake Tahoe, Nevada, California, November 26–28 (2007)
29. Bernardini, F., Mittleman, J., Rushmeier, H.E., Silva, C.T., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. IEEE Trans. Vis. Comput. Graph. 5, 349–359 (1999)
30. Nehab, D., Rusinkiewicz, S., Davis, J., Ramamoorthi, R.: Efficiently combining positions and normals for precise 3d geometry. ACM Trans. Graph. 24, 536–543 (2005)
31. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann Publishers, San Francisco (2006)
32. Guibas, L., Stolfi, J.: Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. ACM Transactions on Graphics 4, 74–123 (1985)
33. Baumgart, B.: A polyhedron representation for computer vision. In: National Computer Conference, pp. 589–596 (1975)
34. Braid, I.: The synthesis of solids bounded by many faces. Communications of the ACM 18, 209–216 (1975)
35. Mäntylä, M.: Geometric and Solid Modeling: An introduction. Computer Science Press (1988)
36. Pébay, P.P., Baker, T.J.: Analysis of triangle quality measures. Math. Comput. 72, 1817–1839 (2003)
37. Bischoff, S., Kobbelt, L.: Towards robust broadcasting of geometry data. Computers & Graphics 26, 665–675 (2002)
38. Mehlhorn, K., Yap, C.: Robust Geometric Computation. under preparation (2004), <http://www.cs.nyu.edu/cs/faculty/yap/book/egc/>

39. Kettner, L., Mehlhorn, K., Pion, S., Schirra, S., Yap, C.K.: Classroom examples of robustness problems in geometric computations. *Comput. Geom.* 40, 61–78 (2008)
40. Li, C., Yap, C., Pion, S., Du, Z., Sharma, V.: Core library tutorial, pp. 1–46. Courant Institute of Mathematical Sciences, New York University (2003)
41. Mehlhorn, K., Näher, S.: LEDA. Cambridge university Press, Cambridge (1999)
42. Alexandroff, P.: Elementary Concepts of Topology. Dover, Inc. (1961)
43. Henle, M.: A Combinatorial Introduction to Topology. Herman, Paris (1979)
44. Mitchell, J., Mount, D., Papadimitriou, C.: The discrete geodesic problem. *SIAM J. Comput.* 16, 647–668 (1987)
45. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S., Hoppe, H.: Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.* 24, 553–560 (2005)
46. Polthier, K., Schmies, M.: Straightest geodesics on polyhedral surfaces. In: SIGGRAPH 2006: ACM SIGGRAPH, Courses, pp. 30–38. ACM, New York (2006)
47. Synave, R., Gueorguieva, S., Desbarats, P.: Constraint shortest path computation on polyhedral surfaces. In: Proc. of the 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008 (2008)
48. Surazhsky, T., Magid, E., Soldea, O., Elber, G., Rivlin, E.: A comparison of gaussian and mean curvatures estimation methods on triangular meshes. In: ICRA, pp. 1021–1026 (2003)
49. Wood, B., Aiello, L., Wood, C., Key, C.: A technique for establishing the identity of "isolated" fossil hominin limb bones. *Journal of Anatomy* 193, 61–72 (1998)
50. Martin, R., Saller, K.: *Lehrbuch der Anthropologie*. Bd. 1. Fischer G Verlag, Stuttgart (1957)
51. Scheuer, L., Black, S.: *The development of juvenile osteology*. Academic Press, London (2000)
52. De la Villetanet, B.: Utilisation d'outils de mesures en 3d dans le cadre d'une etude comparative morphofonctionnelle de tali d'hominoides actuel et du hominide fossile. Technical Report Octobre, Univ. Bordeaux I, MS,spécialité anthropologie (2005)
53. Slice, D.: *Modern morphometrics in physical anthropology*. Kluwer Academic Plenum Publishers (2005)
54. Allard, T., Sitchon, M., Sawatzky, R., Hoppa, R.: Use of hand-held laser scanning and 3d printing for creation of a museum exhibit. In: 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (2005)
55. Gill-Robinson, H., Elias, J., Bender, F., Allard, T., Hoppa, R.: Using image analysis software to create a physical skull model for the facial reconstruction of a wrapped aghmimic mummy. *Journal of computing and information technology* 14, 45–51 (2006)
56. Carter, Y., Allard, T., Collin Moore, N., Goertzen, A., Klonisch, T., Hoppa, D.: The role of printing in teaching and education in human skeletal anatomy. *The Journal of the Federation of American Societies for Experimental Biology* 23 (2009)
57. Maekawa, T.: An overview of offset curves and surfaces. *Computer-Aided Design* 31, 165–173 (1999)
58. Lee, S.H.: Offsetting operations on non-manifold topological models. *Comput. Aided Des.* 41, 830–846 (2009)
59. Nizam, A., Gopal, R., Naing, L., Hakim, A., Samsudin, A.: Dimensional accuracy of the skull models produced by rapid prototyping technology using stereolithography apparatus. *Archives of Orofacial Sciences* 2006, 60–66 (2006)
60. Garimella, R.V., Swartz, B.K.: Curvature estimation for unstructured triangulations of surfaces. Technical Report LA-UR-03-8240, Los Alamos National Laboratory (2003)

Creating Passion for Augmented Reality Applications – A Teaching Concept for a Lab Course

Christian Waechter*, Eva Artinger, Markus Duschl, and Gudrun Klinker

Fachgebiet Augmented Reality
Technische Universität München, Fakultät für Informatik
Boltzmannstraße 3, 85748 Garching b. München, Germany
firstname.surname@cs.tum.edu

Abstract. We introduce a project-based concept for teaching Augmented Reality (AR) applications in a lab course. The key element of the course is that the students develop a stand-alone application based on their own idea. The complexity of Augmented Reality applications requires software engineering skills and the integration of AR specific solutions to occurring problems.

The students work self-responsible in a team with state-of-the-art methods and systems. Moreover they gain presentation and documentation skills. They define and work on their individual goals and challenges, which are assembled in a final application. The identification with the goal of the group creates passion and motivation for creating the AR application.

Beside of the teaching concept we present some of the students' applications in this paper. Furthermore we discuss the supervision effort, our experiences from the supervisors' view and students' feedback.

1 Introduction

Augmented Reality Applications are gaining more and more attention in the last view years. Augmented Reality (AR) is no longer found only in research laboratories. The first commercial applications are already available. AR is used for advertisement purposes or navigation systems. The car manufacturer BMW presents the Mini Cabrio with an AR print advertisement¹. Another AR application is the Virtual mirror provided by the sunglass company Ray-Ban, which enables the user to virtually try on sun glasses². Also navigation systems like Wikitude³ augment the street with virtual navigation information shown on mobile devices. These are only the first steps. AR has a very high potential for the future. Therefore it is very important to teach AR in universities in order to educate the next generation of students, which is able to work in the field after finishing their studies.

We present in this paper the concept of our project-based Augmented Reality lab course first introduced in the winter term 2009. The lab course gained support from the excellence initiative at the university as "Leuchtturm Praktikum" in winter 2009.

* Corresponding author.

¹ <http://www.mini.de/de/de/webcam/index.jsp>

² <http://www.ray-ban.com/usa/science/virtual-mirror>

³ http://www.wikitude.org/category/02_wikitude/wikitude-drive

This initiative awards courses which are very hands-on and which use state-of the art technology. The AR lab course aims to teach the various aspects that must be considered in order to create an Augmented Reality (AR) application. These aspects concern low-level as well as high level implementation and therefore cover the full bandwidth of challenges in building an AR application. The students are taught on the usage of modern software development toolkits and off-the-shelf hardware to program their own stand-alone AR application during the semester.

Students do not necessarily need preliminary knowledge in Augmented Reality but it is an advantage. The fields of study of the students during the last two semester were Computer Science and Computational Science and Engineering. The course is a major part of their semester as students get 10 ECTS points while 30 points are recommended for each semester.

Earlier AR courses which are offered already since ten years at the Technical University of Munich (TUM) were more exercise based [1]. The newer, project-based concept of the AR lab course, introduced in this publication, fulfills also the requirements from the official university module description [4]. The students should gain the following skills during the lab course:

- The competence to apply state-of-the-art methods and systems
- The ability to work goal-oriented and in teams
- Skills to present and document the results

The project-based concept of the course provides a broad range of experience in building Augmented Reality Applications. Those issues range from tracking infrared as well as paper-based markers, to sending data over network connection, to controlling Bluetooth devices. Also non-technical skills are addressed like the development of the application idea and concept as well as the design of the graphics. The students gain knowledge in interpersonal skills like team work and project management. They build the applications in groups from the first idea to the final application, which can be demonstrated at the end of the term. Building a time schedule with project milestones, splitting the work, and putting all individual parts together are also essential parts of the student work.

2 Related Work

Some teaching concepts are briefly reviewed in the following paragraphs due to their importance or inspiration. Maxwell did a survey of different computer vision and image processing courses [2]. He mentions all courses to teach the students well known algorithms by using real-world images and problems. From the survey it is clear that satisfying results are important for the motivation of the students. Analog to this observation we also want to lead the students to build successfully applications that will be demonstrated in public at our chair for one week at the end.

Essa and Brostow [3] aim also for the high motivation of students for creating digital video effects in small projects in order to teach the syllabus of the course. Their

⁴ https://drehscheibe.in.tum.de/myintum/kurs_verwaltung/cm.html?id=IN2106

target group are undergraduate students with only preliminary or no knowledge in the required fields of image processing. They attract always a high number of students (21-30) with their course offer. This course is also a good example for attracting students while teaching the contents in order to reach an appealing goal.

Another interesting group based teaching concept is applied by Pang [4]. He challenges the students by letting them develop own games using game development engines the students had to program in an earlier stage of the course. The students are faced with other students implemented features and their documentation of the code. This teaches the students important aspects of programming applications in a self-awareness process. In order to build AR applications we rather use commercial game development software. There are too many aspects in building an AR application such that this concept could apply here.

3 Teaching Concept

In order to teach the students the various skills of an (AR) software engineer we involve the students from the beginning of the lab course. In order to do so we pay attention to several aspects of the lab course introduced here:

Group Based Working. We separate the students into groups incorporating three to five students. This group finding process takes place in the second session of a semester since there is normally some fluctuation in the first week of the course. The group building phase is important for the further process of the lab course. It is mainly influenced by the availability of the students for the different days during the week, and their native language. Although experience and knowledge in different fields of computer science are good criteria for building groups we attach more importance on having all group members sharing the same day within the week to discuss topics and support each other. These regularly meetings are very important at the beginning of the course such that the students start to share the same vision of their final application. Later on, the students can start arranging their time in a self responsible way.

Own Ideas. We offer the students the possibility to develop applications based on their own ideas. Instead of predefined tasks the students have to make the decision what application they like to implement. The brainstorming of the teams takes place after the group building phase. The students have to design own project drafts and discuss these concepts among their group members. Well-known computer games are the main source of inspiration and we guide these discussions such that the students do not drift away in too detailed discussions. Finally the process of idea creation ends in a presentation of the idea from each group to the whole class.

The process of creating own ideas and the possibility to implement them is a key element in our lab course. We expect a high motivation of the students in order to realize their own ideas.

Self Responsibility. Besides other goals, the students, which will finish our university should be independent of supervision, self-confident, experienced and have the ability to adapt to new fields in computer science quickly. In our lab course we aim for those goals and we want to have the students mainly acting in a self-responsible way. Besides the

shared working day during the week the groups are given no restrictions when to meet and work. The students have to organize their group in a self-responsible way which includes finding their own role within the team. Besides three introduction lectures to the programming environment, the tracking framework and appropriate documentation we give no additional input to the students. We rather let the students get in touch with the provided hard- and software as early as possible, so that they can start gathering experience by themselves. As supervisors we offer office hours for the students. If there are any questions they can contact us and we give them feedback about their actual work. We are also available to discuss possible future steps so that the projects are not diverging but still leave the final decision open to the students.

Documentation. Besides the technical challenges, which the students have to face, they are advised to document the final application as well as their personal work. The application documentation follows the guidelines for the “Game Design Document” given by McGuire and Jenkins [5]. The Game Design Document is a framework for communicating the concept of a game to a publisher. At the same time it should document the development of the described application and therefore it should always incorporate the status of the application. The personal documentation follows no strict framework but should include a more detailed description about the personal work of the group member, the task given inside the group, the problems faced and the solution. This documentation is part of the basis for the student’s grade.

Presentation. Presenting the work is also part of the lab course. Each student has to present his or her own role within the team at the beginning of the lab course. At the end of the semester a short, private presentation to the supervisors should expose the students task within the semester and the problems he or she personally had to face. The short private presentation is also a basis for the student’s grade. In addition to the private presentations the students give one final, public presentation, followed by a public demonstration of their applications. We let the students have the presentations to learn how to communicate own ideas and applications. Especially the final presentation should have the character of a product presentation in order to convince the publisher similar to the “Game Design Document”.

4 Course Contents

Designing an AR application involves several challenges. The students learn in our lab course about various aspects one has to consider and should find own ways of handling certain problems, as far as possible. The most important aspects are listed here.

Tracking. The students are provided with a rigidly mounted infrared tracking system⁵ consisting of four cameras, that cover a space of about nine square meters, the application area. A single, ceiling mounted camera, can be used in addition for marker based tracking using our own tracking library⁶. Although the students are provided with these systems they have to choose the appropriate one for their application. This involves considerations about the advantages and disadvantages of the single computer vision based

⁵ <http://www.ar-tracking.de>

⁶ <http://www.ubitrack.org>

systems for their application. Later on, they have to deal with the complexity of calibrating these systems until the systems are usable for their application. This involves calibration of cameras, room and rigid bodies and absolute orientation or hand-eye calibration. One group came also up with the idea of integrating a different vision-based tracking system. Although they used our tracking at the end it was interesting to see the increasing motivation of the students, driven from demands our provided systems could not met.

Visualization. The main application is developed using 3DVia's Virtools⁷, a rapid prototyping and development environment. Virtools allows to concentrate on designing the higher level application. This approach is in contrast to many lab courses where students program in a lower level language C/C++/C#, JAVA, etc. For their application the students can make use of the visualization techniques that are already incorporated but they can also program own elements, e.g. pixel shaders. This enables the students to deal on a higher level by choosing among appropriate techniques. They can experiment how to create an immersive perception of the virtual elements placed within the real world using drop shadow, reflectivity, alpha blending, etc. of the virtual objects. In addition the occlusion of real world elements, e.g. moving robots, should be considered such that they are correctly overlaid with 3D graphics.

Modeling. By designing their own AR application the students also have to take care of an appropriate look of the virtual objects. Since downloading models from the Internet is in many cases not sufficient the students learn to create own models and animations or editing of already existing models. A challenge the students have to face is the importing/exporting of animations from different file formats such that the models can be used from the rendering engine in the end. If models with a high number of polygons are used the model has to be adjusted such that the rendering engine is capable of showing the model for the AR application in real time.

Interaction. We provide the students with hardware interaction devices like the Wii remote or infrared finger tracking. Using the Wii remote is of high motivation for the students although in all cases there was no time for a complete integration of this device into the final application. Nevertheless the students start to think about meaningful interaction concepts besides normal keyboard controls. In case of the finger tracking some limitations of the hardware led the students to come up with only few simple gestures for controlling their application.

5 Students' Applications

In the following we describe several projects, which were developed during the courses in winter 2009 and summer 2010.

5.1 ARace

ARace is an augmented car racing game for two players. Two robots with markers on top, representing the virtual racing cars, control the virtual game [1\(a\)](#). The robots are

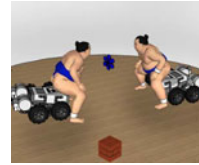
⁷ <http://www.3dvia.com/products/3dvia-virtools/>



(a) ARace



(b) ARace



(c) Sumot



(d) Ragnarök



(e) AR Painting



(f) Treasure Hunt

Fig. 1. Screen shots of the different projects developed during the AR Lab Course. The games shown in figures [I\(a\)](#) to [I\(c\)](#) were developed during the winter term 2009. Figure [I\(d\)](#) to [I\(f\)](#) show results of the lab course in the summer term 2010.

tracked by a camera mounted on the ceiling. The game is played from the robot's perspective using radio cameras mounted at the front of each robot. This view is augmented with the virtual cars, the race course and some special items (speedup, rockets, etc). An augmented view can be seen in [I\(b\)](#). The goal is to drive as fast as possible four laps or to hunt down the other player before he can finish the race⁸.

5.2 Sumot

In the augmented reality game Sumot two Lego Mindstorm robots, equipped similar to the ones in ARace, fight in a Sumo arena, trying to force the opponent out of a ring, see figure [I\(c\)](#). The players are not limited to using only physical forces: They are equipped with several imaginary weapons, which freeze the other ringer or to steal him hit points which affect the physical representation (the robot). The winner is the player who passes the other sumo ringer out of the ring.

5.3 AR Painting

AR Painting is inspired by the 'Tagged in Motion' project from the 'Jung von Matt' group⁹. The graffiti artist Daim used an ARToolKit based drawing application for drawing in different color, brightness and line thickness. The students developed an application with several extended three dimensional drawing tools to choose from. Depending on the chosen tools the user can either draw a ribbon, a tube, use an arbitrary shape or use a particle based spraying tool. The user wears an video-see-through HMD showing the graphics, which is tracked by an infrared tracking system. The application is based

⁸ A video of ARace <http://www.youtube.com/watch?v=BXG115JU35o>

⁹ http://www.jvm.com/de/ideas/#/4_111/jung-von-matt-tagged-in-motion/

on a finger tracking device to have an immersive human-machine interaction so that the user can directly paint using his own fingers by applying certain gestures.

5.4 Treasure Hunt

The goal of this AR game is to find a treasure hidden in a swamp. Each player controls a robot, which is represented as a pirate in the augmented view. This time the robots are tracked using the infrared tracking system and again they are equipped with radio cameras. In order to find the treasure, the players need to collect parts of a treasure map, which are physical blocks standing in the arena. These blocks have to be transported by the robots. One player can only carry one hint at the same time. There are also virtual obstacles like bananas and spiders, which can be placed to hinder the other player.

5.5 Ragnarök

Ragnarök is a game basically featuring the idea of capturing the flag, with some interesting additional ideas. In this game, the robots represent different opposing parties wanting to capture the flag of the opponent. The players perspective is the camera mounted on top of the robots. The AR aspect is mainly realized by making it possible to add so called "towers" to the game arena by dropping physical red square markers. These towers protect the own flag, but they can be shot by the opponent. An interesting aspect of the game is the possibility to change the environment where the "fight" takes place. Possible environments are, medieval, alien terrain or futuristic urban environment.

6 Discussion and Evaluation

The following section will include the experience the supervisors had during the lab course as well as the evaluation of the course.

6.1 Supervision Effort and Supervisors' Experiences

The structure and grading of the lab course changed between the first time (winter term 2009) and the second time the course was conducted. This had to be done due to the enormous amount of time the supervisor had to spend with the students and due the increased number of students in the summer semester 2010.

Winter 09/10. In winter 2009/2010 the supervision effort was enormous, although only 6 students did attend the lab course 3 supervisors did spend at least 5 hours each week supervising the students, not including individual questions to a specific supervisor. The positive aspect of this approach is, that it was easily possible to see the students' effort and grade them accordingly. The official presentation as well as the overall status of their projects was not taken too much into account.

Summer 2010. For the summer semester 2010 it was decided to change the supervising approach simply due to the fact that each supervisor is also involved in industry projects and the number of students increased from 6 to 13. Therefore it was decided that each supervisor is responsible for one group and the grading of the group will be more objectively measurable. This approach first of all resulted in a more accurate planning of

the tasks the students had to perform as well as in more organizational effort that had to be done beforehand. During the semester the division into groups and the introduction of office hours resulted in a more controllable timetable and all together could be considered less time consuming.

6.2 Students' Feedback

This section will cover the evaluation of the lab course given in the summer semester 2010. The following diagrams were created on the base of 13 evaluations, meaning all students did participate in the evaluation process. The evaluation is realized by the student representatives of the Computer Science department each semester. The evaluation uses the same standardized questionnaire for all lab courses in our department.

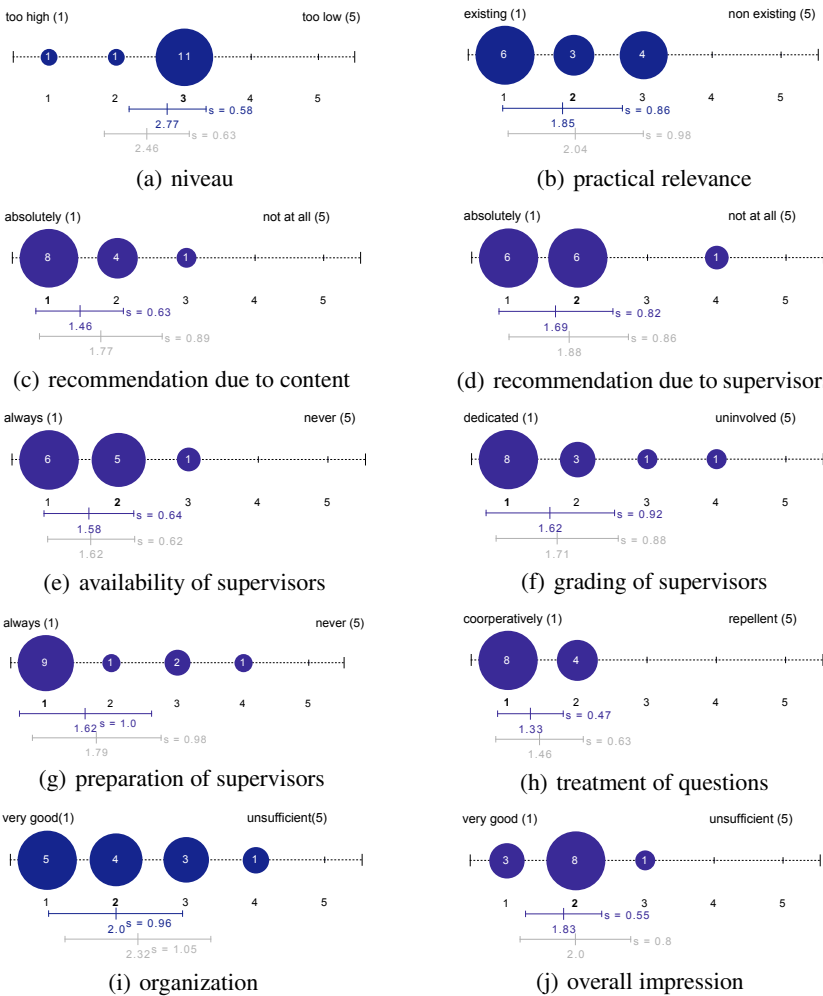


Fig. 2. Evaluation Results

Figure 2(a) shows that the content that was chosen, was experienced as not too difficult as well as not too easy, but just right by 11 out of 13 students. The practical relevance was rated high from 6 students (Figure 2(b)). As already been said, this lab course has a free mind, everything that is suitable in the research field of AR can be integrated, thus those results, especially Figure 2(b) simple reflects the practical relevance of AR in todays society. Figure 2(a) shows how much the students would recommend the lab course to other students.

Concerning the opinion on the supervisors: Figure 2(d) to Figure 2(h) show that the effort and time that was contributed by the supervisors before and during the lab course was greatly appreciated by the students. In all aspects the course received a better rating than the overall average of all lab courses at the computer science department (indicated by the line at the bottom of the image).

Figure 2(i) shows the evaluation of the organization of the lab course. The mixed results again are due to the wide bandwidth of opportunities as well as the rather big changes that were made during the first and second semester. Problems that were occurring are: not available software (Maya), problems with the delivery of additional hardware (Lego NXT and additional workstations). The overall impression of the lab course in Figure 2(j) shows that the idea and the realization of the lab course was received good by the students, although the time they had to invest into the lab course was considerably.

Also in the evaluation there was an part with open questions. Table 1 to table 3 show the questions as well as the answers of the students.

Table 1. Statement on the content

Content			
Content I liked		Quality of the lab course	
different fields	2 statements	good	5 statements
producing a working product	1 statement	appropriate	1 statement
total statements	3 statements	total statements	6 statements

Table 2. Statement on the preparation

Preparation			
Preparation I liked		Could have been done better	
introduction to all the different topics	3 statements	better introduction into Virtools	3 statements
introduction to a specific topic	2 statement	more introduction sessions	1 statement
total statements	5 statements	total statements	4 statements

Table 3. Statement on the organization

Organization			
I liked		Could have been done better	
practical examples of tools	1 statement	more supervision	1 statements
independent working	1 statement	mini deadlines	1 statement
availability of tutors	1 statement	availability of hardware	1 statement
total statements	3 statements	total statements	6 statements

7 Conclusion

Conclusively it can be said, that the evaluation results show exactly what was anticipated by us. Most students accept and like the possibility of an open and free environment, they do not need to be patronized, and being under supervision all the time. It has to be said though that this concept might only be valid in certain study areas, where it is necessary to solve problems with all sort of tools, and no strict solution is given beforehand. Although it shows that the field of Augmented Reality is a good playground for students to (self)learn different techniques while having a clear, touchable and last but not least fun goal.

References

1. Bauer, M., Wagner, M., Toennis, M., Klinker, G., Broy, V.: Lehrkonzept für ein Augmented Reality Praktikum. In: 1. Workshop Virtuelle und Erweiterte Realität, Chemnitz, Germany (2004)
2. Maxwell, B.: A survey of computer vision education and text resources. *International Journal of Pattern Recognition and Artificial Intelligence* 15, 757–774 (2001)
3. Essa, I., Brostow, G.: A course on digital video special effects. In: IEEE Workshop on Undergrad Education and Image Computation, Hilton-Head Island, SC (2000)
4. Pang, A.: Group interactions in a game engine class. In: Proceedings of the 3rd International Conference on Game Development in Computer Science Education, pp. 26–30. ACM, Miami (2008)
5. McGuire, M., Jenkins, O.: *Creating Games: Mechanics, Content, and Technology*. AK Peters Ltd., Wellesley (2009)

Object Material Classification by Surface Reflection Analysis with a Time-of-Flight Range Sensor

Md. Abdul Mannan, Dipankar Das, Yoshinori Kobayashi, and Yoshinori Kuno

Graduate School of Science and Engineering, Saitama University,
255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama 338-8570, Japan
{mannan,dipankar,yosinori,kuno}@cv.ics.saitama-u.ac.jp

Abstract. The main objective of this work is to analyze the reflectance properties of real object surfaces and investigate the degree of roughness. Our non-contact active vision technique utilizes the local surface geometry of objects and the longer wavelength scattering light reflected from their surface. After investigating the properties of microstructure of the material surface, the system classifies various household objects into several material categories according to the characteristic of the micro particles that belong to the surface of each object.

1 Introduction

Some of the dreams of robotic scientists are to develop service robots such as doing tedious housekeeping work, playing with children, and assisting in the therapy of patients in hospitals. To realize these robots, we often need a vision system that can locate certain objects in the environment - the capability, which we denote as object detection or object recognition. Many researchers have attempted to make service robots recognize objects through vision processing in natural environments. However there is no conventional vision system that can recognize target objects in a real-world workspace without fail. Object recognition by using object's shape, size, color or texture has already been well studied in this field. To manipulate an object in more comfortable way, however, a service robot should know some more detail about the object such as object's material, smoothness of the surface or some other physical properties. In this paper, we consider object's surface smoothness, which is closely related to the information about the material of the object.

There has been a great deal of research on obtaining surface properties. How to compute surface orientation or surface geometrical information has attracted much attention of computer vision researchers. Photometric stereo is one of such techniques, which was first proposed by Woodham [1] and later extended by other researchers [2]-[5]. It estimates local surface orientation by using several images of the same surface taken from the same viewpoint but under illumination from different directions. The major limitation of this technique is the need for highly controlled environment. Textural characteristic was also examined to obtain surface orientation [6], [7]. The accuracy of this method highly depends on the visibility of textural characteristic.

Surface properties related to object material have also been investigated. An active vision technique was introduced in [8], where the authors used two sources of laser

light and a pair of CCD cameras to differentiate surface of various materials. They achieved 60% and 78% success rates to identify two unknown materials. In addition to this low accuracy, this technique has some shortcomings that make this method impractical for use in service robots to recognize house hold objects in home environment. The system needs some accurately located devices to illuminate surface area and take image. It requires 2 W YAG laser light that is non-visible and harmful for human eyes. In addition, the paper did not clarify the impact of visible light on result. In this system, there must be an interference effect between visible light and the infrared laser reflection, which will change the actual reflectance pattern. Another active vision technique was proposed in [9]. Here the authors integrated a structured light based vision system to investigate surface roughness, defect and waviness. In this system, however, the complexity of various instruments is also huge and the method requires tedious illumination setting work. An optical-based measurement mechanism was proposed in [10], which enables noncontact assessment of two mechanical properties of objects: Poisson ratio and effective stiffness. In this method, laser generated ultrasound was used as a probe that produces Lamb wave spectrum in the target material and this spectrum was detected by an interferometric optical detector. The major drawback of this method is that the acoustic power generator produces a great deal of local heating on the small area of the object surface. And the surface damage may be unavoidable.

In this paper, we propose a method that overcomes the limitations as mentioned above. We combine the techniques to investigate both surface geometrical and micro structural information. The method can investigate surface of a real object by analyzing its reflectance properties against infrared light and classify the object into several classes according to its surface roughness. This is an active vision technique that uses infrared light to obtain reflection from surface and analyzes the variation of local surface orientation of various objects from a common reference view point. Despite of using longer wavelength light to obtain reflection, the reflection does not become completely color independent. The reflection still contains some color information which hampers the accuracy. Our proposed method can overcomes this type of difficulty and hence the accuracy becomes high.

Another advantage which makes this method more practical for robotic applications is its simplicity. In our proposed scheme we need only a 3D range finder camera using the time-of-flight method to investigate the object surface roughness from distant location. Such range finders have already been used for localization and object shape recognition in robotics. Our method can use this existing range finder to get reflectance from the object surface. Hence in robot applications the method does not need extra equipment.

The main objective of our proposed method is to classify various household objects into their materials according to their surface roughness. It can determine which class of surface roughness that an object belongs to if the object has a single color region on its surface which has constant vertical orientation and varying horizontal orientation or vice versa.

2 Light Reflection Principle

If electromagnetic energy reaches any surface, it must be reflected, absorbed, or transmitted. The proportions accounted for by each process depend upon the nature of

the surface, the wavelength of the energy, and the direction of illumination. Every object has a spectral signature [11]. If we can detect the spectral signature, we can obtain good features to get an insight to the type of objects. In this research, we are interested in the reflectance property of an object that is the ratio of the amount of electromagnetic radiation, usually light, reflected from a surface to the amount originally striking the surface. Depending on the size of the surface irregularities relative to the incident wavelength, there are two reflection types: specular reflection and diffuse reflection [12], [13], [14].

a) *Specular Reflection*

This type of reflection takes place at the interface of two mediums (Fig. 1 (a)). It is a smooth surface reflection relative to the incident wavelength. If surface particles are small relative to the wavelength of incident light, light is reflected in a single direction. The angle of reflection is equal to the angle of incidence. Sometimes it is called 'mirror' reflection.

b) *Diffuse Reflection*

This type of reflection occurs if the surface is rough relative to the incident wavelength (Fig. 1 (b)). It also called isotropic or Lambertian reflectance. Energy is reflected (scattered) equally in all directions (more or less). As the particles are randomly oriented, the reflections will be randomly distributed. Many natural surfaces act as a diffuse reflector to some extent. A perfectly diffuse reflector is termed a Lambertian surface and the reflective brightness is the same when observed from any angle.

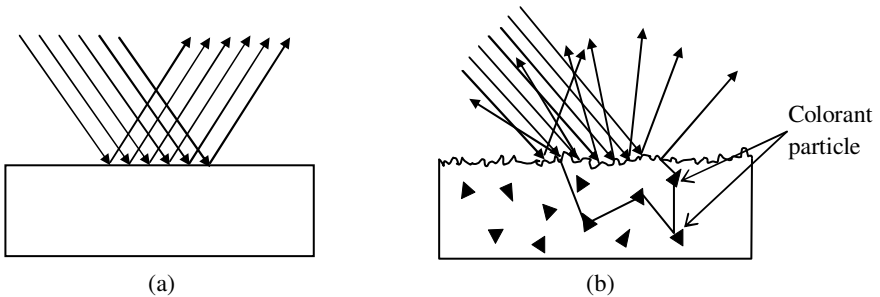


Fig. 1. Light reflection. (a) Specular reflection from the material surface, (b) Diffuse reflection from the material surface and subsurface.

In the diffuse reflection process, the incident light penetrates into the subsurface of the material. After entering the subsurface, some part of the light is absorbed by the colorant particles, some part is transmitted through the material and the rest is reflected and scattered by the colorant particles and get back to the first medium again. The scattering and getting back of the light components entirely depend on the wavelength of the incident light and the type of the colorant particles. This interaction considerably changes the spectral composition of the reflected light. Actually the reflected light components that come out from the subsurface contain the color information of the reflecting surface.

The other important phenomenon of wave is diffraction. Light has that property and that is why if light ray strikes an obstacle in the path of its propagation, the light may be reflected or diffracted depending on the size of the obstacle and the wavelength of light ray.

3 Method

It is a common phenomena that a wave reflects off only an obstacle that has dimension larger than the wave length. If the wavelength is larger than the obstacle size, the wave will bend and pass away the obstacle and no reflection will happen. The same thing happens in the case of light wave. It is difficult to image an object that is smaller than half the wavelength of the light used for imaging. That means the wave that has much longer wavelength than an obstacle cannot give any information about the obstacle. That is why we cannot see virus with a light microscope, but by using an electron microscope we can see it because electron has wavelength 100000 times smaller than the visible light. If we use visible light to illuminate a virus, the light will not reflect back by the virus. This means that the visible light overlooks the virus and it seems we can look through it.

In this way the concept of using longer wavelength to see through smoke is already used. Microwave has comparatively longer wavelength and that is why we can see through smoke fog or rain by using it. As microwave can penetrate haze, light rain, snow cloud and smoke, the wave is good for viewing the earth from space.

Similarly the amount of light reflected by an object, and how it is reflected, is highly dependent upon the degree of smoothness or texture of the surface. When surface imperfections are smaller than the wavelength of the incident light as in the case of a mirror virtually all of the light is reflected equally in the specific direction.

In the real world, however, most objects have convoluted surfaces that also exhibit diffuse reflection, with the incident light being reflected in all directions. If light falls on a surface, some part of the reflection will be specular and some part will be diffuse depending on the size of the micro particles on the surface. By analyzing this reflected light, we can determine the degree of roughness of the surface.

However, the measurement of roughness by using visible light might be difficult in some cases because almost all of the micro particles on the surface cause diffuse reflection especially in the case of matte surface as the wavelength of visible light is much smaller. Thus, if we want to enhance the discrimination features of various surfaces, we have to choose a light which has larger wavelength than the visible light, but it should not be as large as microwave, because light with much larger wavelength will overlook all particles on the surface and we will get only specular reflection for all surfaces.

Thus, we propose to use infrared light to identify surface roughness, which has larger wavelength than the visible light and smaller than microwave. If we use infrared light, some of shorter sized micro particles will be overlooked from all types of surfaces and we will get specular reflection from these surface regions. Other regions that have comparatively large micro particles will give us diffuse reflection. That means depending on the surface micro particle size and the orientation, the percentage of the specular and diffuse reflection will be specified. And hence, by analyzing surface reflectance light, the degree of surface roughness will be measured.

To project the infrared light and to measure its reflectance, we use a range imaging 3D camera, SwissRanger4000 (SR4000) [15]. The SwissRanger4000 is a solid state range imaging device that can determine the distance to objects in a scene simultaneously for every pixels in an image. The device has CCD array to produce the image and it is equipped with a near infrared light source that is ideal for use in dark environment. The device also has optical filter glass in front of it that allows only the light of wavelength near to the illumination LEDs to pass into the camera lens. And it is one of the most important advantages of this device that the indoor lighting has no effect on the SR4000 data. This device produces a digital image data that contains intensity information and also distance or depth information for each pixel of the image in indoor environment. It is capable of producing rang images of 167×144 pixels. The depth information is given by $x y z$ Cartesian coordinate values for each pixel of the image and the intensity value of each pixel ranges from 0 to 65531. It is possible to determine the orientation of the surface part corresponding to each pixel by using its coordinate values.

The sensor receives the reflected infrared light consisting of specular and diffuse components as described in Section 2. The proportion of specular and diffuse components from a surface depends on the degree of roughness of that reflected surface. The more light reflects specularly or diffusely if the surface is comparatively smoother or rougher. In this sensor, both image sensor and light source are placed at the same position. Thus, the sensor receives the maximum reflection from a surface patch if its orientation directs toward the sensor. If the surface orientation is getting away from this setting, the amount of total received reflection decreases. This is because the specular reflection is directional unlike the diffuse reflection. Thus, the larger the specular component, the more the received intensity decreases in such cases. Since the proportion of the specular and diffuse reflection components is a function of surface roughness, we can use the received intensity change rate with the surface orientation change as the measure indicating the surface roughness.

To examine the intensity change rate, we need to find surface segments of different orientations. We divide the viewed surface of an object into small rectangular segments. We check all of the surface segments of the object to find some segments whose orientation with respect to the y -axis (vertical direction) (or x -axis: horizontal direction) is the same over all the segments but whose orientation with respect to x -axis (or y -axis) varies over the segments. These types of segments are considered as *special segments*, which help to determine the surface roughness of the object. In our proposed work the first step is to find such special segments which have approximately the same orientation angle along y -axis (or x -axis) but have varying orientation angle along x -axis (or y -axis). We assume that we can find such special segments on our target objects. It is obvious that most of the household objects roughly resemble to some geometric shapes such as rectangular, pyramidal, circular, conical etc. Thus it is easy to find a region on the surface of household objects that contain some special segments if they are placed in their stable positions. Fig. 2 illustrates that horizontally aligned regions always have special segments for spherical, conical, pyramidal or rectangular objects if they are put in their stable positions.

We adopt this assumption since the main purpose at this stage of research is to confirm if we can obtain surface roughness property through the proposed approach. This assumption, however, can be easily removed. We use a time-of-flight range

sensor to realize our vision system. With this sensor, we can obtain the orientation data of many surface points. From these data, we can find a set of appropriate axes that satisfy the same conditions mentioned above about the x - and y -axes. In other words, we try to find surface orientation vectors on a plane. Then, the new x -axis is taken on the plane and the new y -axis is set perpendicular to the plane. Moreover, if we can precisely compute the surface normal direction at each point from the range data, we may be able to obtain the intensity change rate with the surface orientation direction change directly from it.

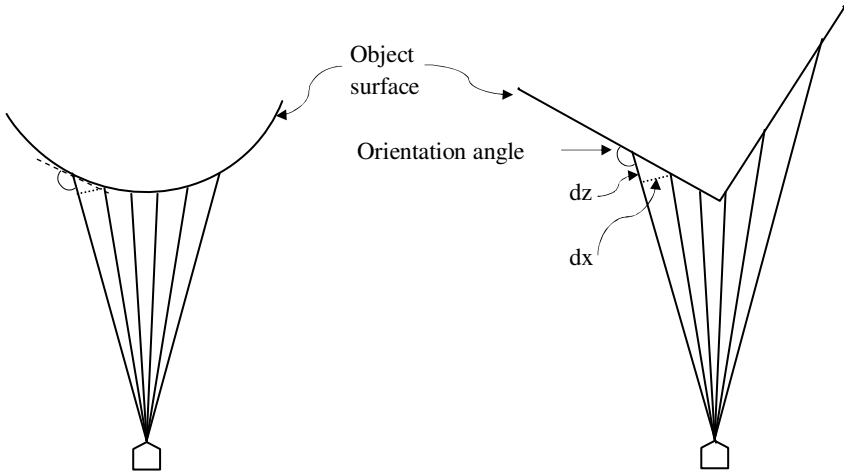


Fig. 2. Surface orientation angles of two common shaped objects

The actual computation method is as follows. We divide the surface into small segments as shown in Fig. 3. Then, we calculate the orientation angle along x and y directions of the surface segments by:

$$\text{Angle}_x = \text{Arctan}(dx/dz), \tag{1}$$

and

$$\text{Angle}_y = \text{Arctan}(dy/dz) . \tag{2}$$

$$\text{Orientation angle along } x\text{-axis} = 180 - \text{Arctan}(dx/dz) . \tag{3}$$

Then we find some special segments. We also determine the value of average intensity I_A of the special segments.

Fig. 4 shows experimental results showing the relationship between the orientation angle and the intensity average. From this figure, it is obvious that the rate of intensity change with orientation angle is different for different surface roughness. This value is higher for those objects whose surface is comparatively smoother than the others. Hence, the system can determine the degree of surface roughness by the value of rate of intensity change with orientation angle.

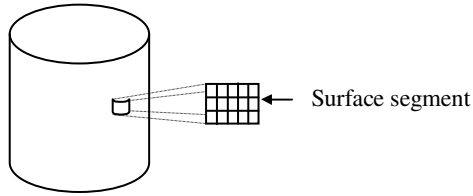


Fig. 3. Surface segment used to measure the orientation angle and reflection intensity

4 Test and Result

We performed experiments using various household objects of six different materials. Fig. 5 displays the intensity images of some objects taken with SR4000 camera. The samples of material are chosen according to the common roughness of regular household object surfaces. The data sets include multiple objects of the same material but with different shapes and sizes such as a roll of toilet paper, a box of tissues, a paper roll, and a paper box. Some objects have also multiple colors even though made of the same material, for example, a dark blue plastic cup and a gray one.

The test was performed by using 12 objects. The intensity value I_A , orientation angle and the rate of intensity change with orientation angle were calculated by applying the technique described in section 3. Fig. 6 shows the result of the 12 test cases.

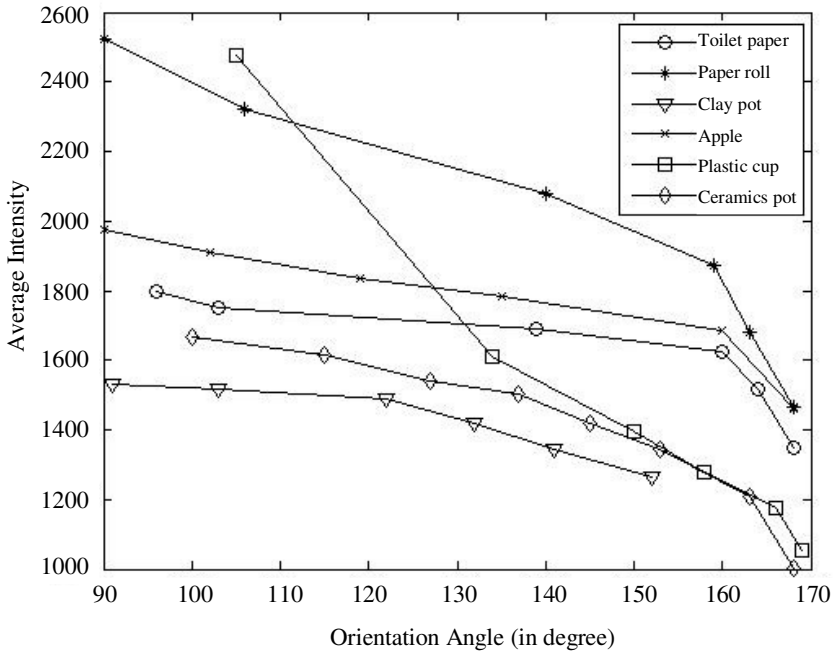


Fig. 4. Surface reflection analysis result. Different surface types are represented by different curves.

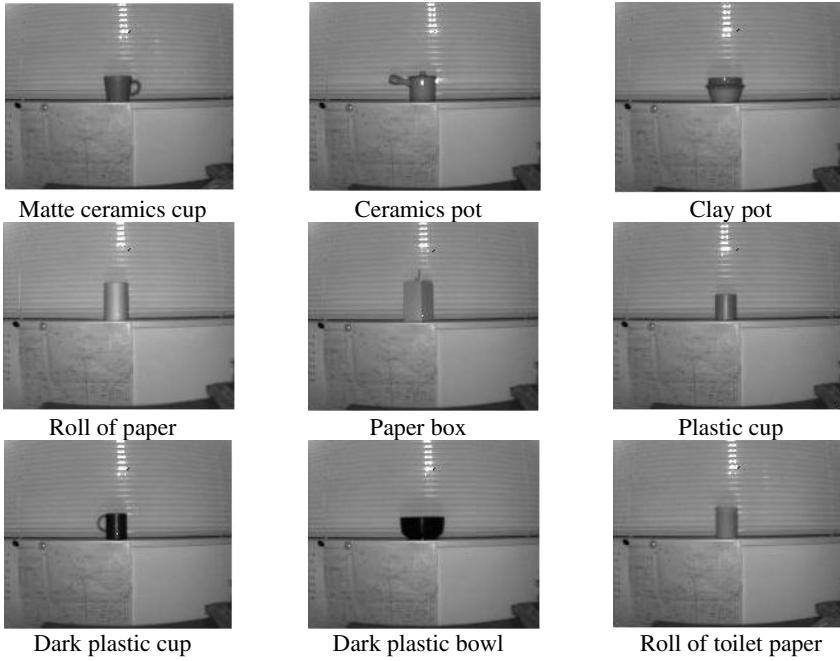


Fig. 5. Intensity images of various household objects taken with SwissRanger4000

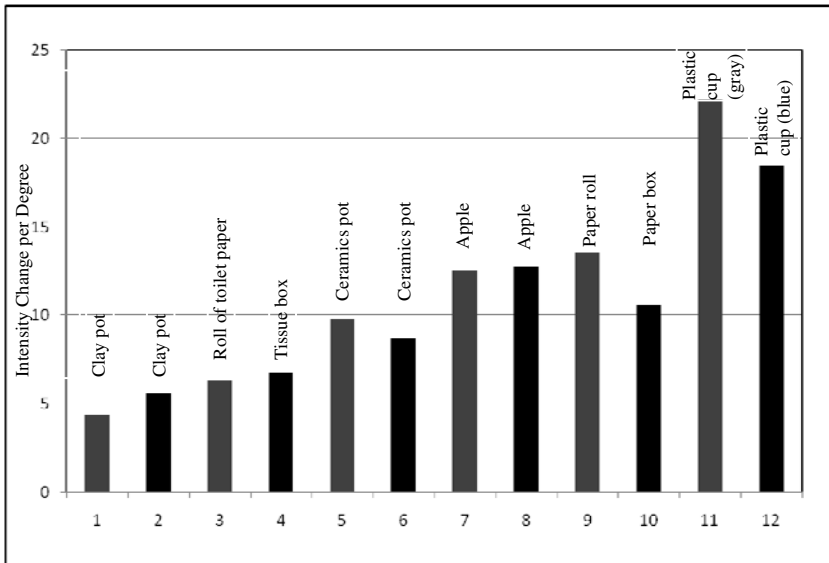


Fig. 6. Result of 12 tests for all materials

Although we need much more experiments, the results are promising that we can classify objects into a certain number (at least three to six) of material classes. The experimental results also show that object color and object shape do not affect the intensity change value.

5 Conclusion

We have proposed a method of computing surface roughness property, which can be considered as object material, with a time-of-flight laser range sensor. We have shown that the intensity of reflected infrared light from a surface patch on an object changes as the relative orientation of the surface patch with respect to the sensor direction changes, and that this changing rate indicates the surface roughness. The range sensor has an infrared light projector and a reflected light receiver. In addition, it can give surface orientation data. Thus, we can obtain the surface roughness property with this sensor alone. Since the original function of the sensor is to obtain 3D shapes of objects, we can develop an object recognition system with this sensor that can consider object material as well as shape. Human users may ask a robot, "Get that metal box," or "Get that plastic box." Our object recognition system can meet such requests. We are now developing such a robot vision system.

Acknowledgments

This work was supported in part by JSPS KAKENHI (19300055).

References

1. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. *Optical Eng.* 19(1), 139–144 (1980)
2. Horn, B.K.P.: *Robot Vision*. McGraw-Hill, New York (1986)
3. Choe, Y., Kashyap, R.L.: 3-D shape from a shaded and textural surface image. *IEEE Trans. Pattern Anal. Machine Intell.* 907–919 (1991)
4. McGunnigle, G., Chantler, M.J.: Rough surface classification using point statistics from photometric stereo. *Pattern Recognition Letters* 21(6-7), 593–604 (2000)
5. Chantler, M.J., Petrou, M., Penirsche, A., Schmidt, M., McGunnigle, G.: Classifying surface texture while simultaneously estimating illumination direction. *International Journal of Computer Vision* 62(1-2), 83–96 (2005)
6. Krumm, J., Shafer, S.A.: Texture segmentation and shape in the same image. In: *Proceedings of the Fifth International Conference on Computer Vision*, pp. 121–127. IEEE Computer Society, Washington (1995)
7. Malik, J.R., Rosenholtz, R.: A differential methods for computing local shape-from- texture for planar and curved surface. In: *Proceeding of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 543–547 (1993)
8. Orun, A.B., Alkis, A.: Material identification by surface reflection analysis in combination with bundle adjustment technique. *Pattern Recognition Letters* 24(9-10), 1589–1598 (2003)

9. Tian, G.Y., Lu, R.S., Gledhill, D.: Surface measurement using active vision and light scattering. *Optics and Lasers in Engineering* 45(1), 131–139 (2007)
10. Culshaw, B., Pierce, G.: Pan Jun: Non-contact measurement of the mechanical properties of materials using an all-optical technique. *IEEE Sensors Journal* 3(1), 62–70 (2003)
11. <http://www.crisp.nus.edu.sg/~research/tutorial/optical.htm>
12. Wyszecki, G., Stiles, W.S.: *Color Science*, 2nd edn. Wiley, New York (1982)
13. Shafer, S.: Using color to separate reflection components. *Color Research Appl.* 10, 210–218 (1985)
14. Tominaga, S., Wandell, B.A.: The standard surface reflectance model and illuminant estimation. *J. Opt. Soc. Amer. A* 6(4), 576–584 (1989)
15. <http://www.swissranger.ch>

Retrieving Images of Similar Geometrical Configuration

Xiaolong Zhang and Baoxin Li

Computer Science & Engineering
Arizona State University

{xiaolong.zhang.1,baoxin.li}@asu.edu

Abstract. Content Based Image Retrieval (CBIR) has been an active research field for a long time. Existing CBIR approaches are mostly based on low- to middle-level visual cues such as color or color histograms and possibly semantic relations of image regions, etc. In many applications, it may be of interest to retrieve images of similar geometrical configurations such as all images of a hallway-like view. In this paper we present our work on addressing such a task that seemingly requires 3D reconstruction from a single image. Our approach avoids explicit 3D reconstruction, which remains to be a challenge, through coding the potential relationship between the 3D structure of an image and its low-level features via a grid-based representation. We experimented with a data set of several thousands of images and obtained promising results.

1 Introduction

Nowadays, online media sharing sites such as Flickr (<http://www.flickr.com>) from Yahoo and Picasa (<http://picasa.google.com>) from Google are flourishing. The amount of image content online has experienced enormous increase. As this trend continues, meaningful ways to retrieve images will be very desirable. Currently, the searching function supported in most media sites are largely based on keywords (tags) or low level features such as color. Being able to retrieve images based on geometrical composition of the scene would be interesting and useful for certain applications such as architecture and tourism. In this paper we propose a method for retrieving images with similar 3D configuration.

The most intuitive solution to this problem is to first perform 3D reconstruction and then make comparison between the query image and pool images base on structural information. Unfortunately this solution is practically very challenging, since 3D reconstruction is still a difficult task especially given only a single image. 3D reconstruction methods such as structure-from-X (motion/stereo/defocus) [3] have been proposed to recover depth information from 2D images based on motion parallax, stereo disparity and local optical feature. Most of such methods require more than a single frame. With multiple frames, such methods may still suffer from issues like having only a sparse set of 3D points reconstructed. Markov Random Field (MRF) and Conditional Random Field (CRF) are often introduced to provide a smooth output but cannot ensure true 3D structure. 3D reconstruction based on a single image is a more challenging task due to the lack of depth cues such as stereopsis and motion parallax.

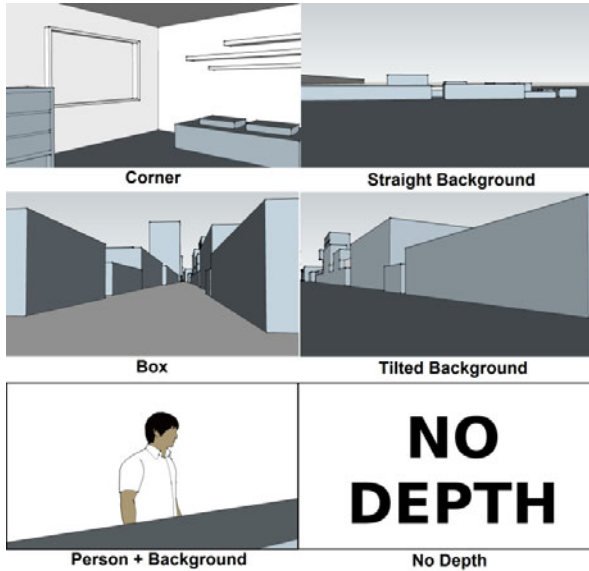


Fig. 1. Six sample scene compositions of an image. Earlier work [7] proposed similar stage types, our work considers six types.

Most single-image-based 3D reconstruction approaches rely on sophisticated image features and learning mechanisms. Hoiem et al. [4,5] proposed a method to classify image regions according to their geometrical orientations with respect to the camera using logistic regression with AdaBoost being used to train the model and with decision trees as weak learners. Saxena et al. [9,10] performed depth estimation from single images by supervised learning with MRF. From a different perspective, Nedović et.al. [7,8] defined several stages based on scene geometry and performed supervised learning on stage types using Support Vector Machine (SVM). Inspired by the concept of scene categorization from [7,8], our approach aims at 3D image retrieval based on six typical scene compositions as shown in Fig. 1. Instead of precise 3D reconstruction from a single image, our objective is to obtain rough description of 3D configuration of the image and then use that for image retrieval.

In Sect. 2 we describe the proposed method, followed by our experiments in Sect. 3, where we describe the image retrieval engine based on the learned results. Finally we provide our discussion and an overview of future work.

2 Proposed Method

The proposed approach is illustrated in Fig. 2. The first component is feature extraction on three levels: segment, macro-grid and entire image. We extract geometrical context features as well as Gabor filter responses and represent the image by concatenating features from the $N \times N$ macro-grids. Retrieval is based on similarity between the query image and pool images using these features.

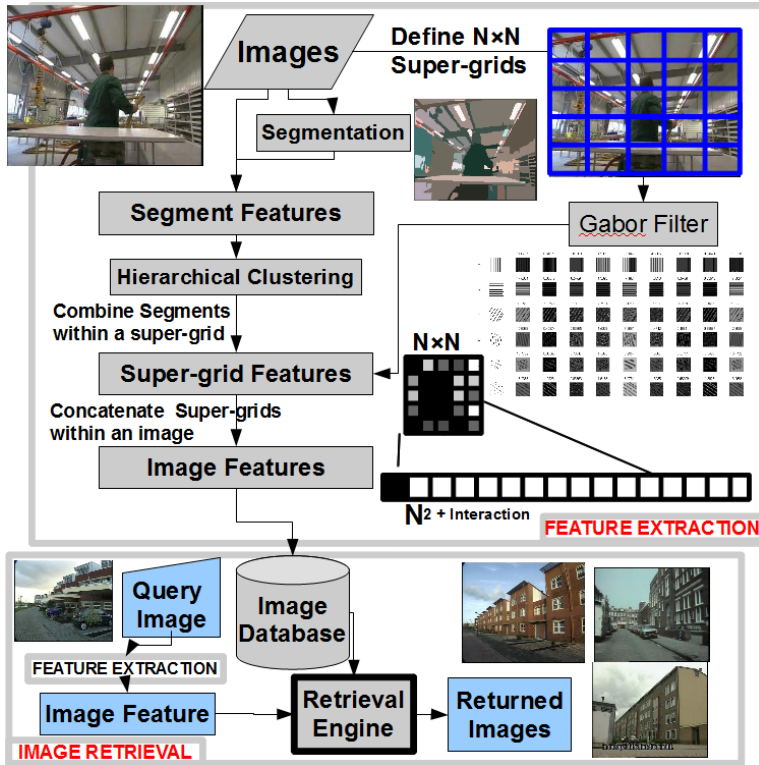


Fig. 2. Illustration of the proposed approach

2.1 Feature Extraction

As mentioned earlier, our approach is based on a three-level image representation: segment, macro-grid and entire image. Segments are generated by applying the graph-cut based algorithm introduced in [2]. Macro-grids represent rectangular regions of the image obtained by dividing an image into $N \times N$ rectangular Mosaics. This representation is a key for the creation of high level label representation of an image, which stands as a mid-ground between the pixel values and the image label that facilitates abstract representation of local image content. Macro-grid features and three types of macro-grid interactions are concatenated to represent an entire image.

2.1.1 Segment Features

To obtain segments, over-segmentation of an input image was performed using a graph-cut based segmentation method originally proposed by Felzenszwalb [2]. Since this fine level of feature represents the most detailed description of an image, in general segments of smaller size are preferred. On the one hand, coarse segmentation result might cause loss of information during feature extraction which involves averaging across all pixels in a segment. On the other hand, having abundant segments does not necessarily harm the performance. The assumption here is that even if a group of adjacent segments belonging to the same region with high similarity are

broken up during the segmentation stage, their locations in the feature space are very likely to remain close to each other when clustering is performed.

After segmentation, feature extraction is performed on each segment. We employ the geometrical context features as used in [3,4]. The first 6 features listed in Table 1 illustrate these features.

2.1.2 Macro-grid Features

Several learning algorithms are based on segment-level features [4,5] while others that further divide the image into artificial blocks [9,10] generally treat the later division more as a means of guaranteeing equal length of image features. We hypothesize that there may be rich semantic and statistical information on this sort of abstract level of representation that can be explored for image retrieval. Interestingly, studies in Cubism [1] suggest that the school of painting pioneered by Pablo Picasso creates an representation of the world that is abstract in form but powerful in expression.

By overlaying an $N \times N$ grid pattern we divide the original image into N^2 parts which is termed as macro-grids in the following sections. For each macro-grid, we further extract three types of features based on the feature, label and Gabor filter responses.

2.1.2.1 Macro-grid Centroid. For each macro-grid, we calculate the weighted average of every centroid that is associated with a segment within this macro-grid. Let S be the number of segments in a macro-grid, the weighted centroid can be expressed as

$$F_{centroid} = \sum_{s=1}^S \alpha_s C_s \quad (1)$$

where

$$\alpha_s = \frac{\# \text{ of Pixels that belong to this segment}}{\text{total \# of Pixels in the macrogrid}} \quad (2)$$

As we can see in (2), the weight coefficient is determined by the area each segment occupies in the macro-grid. C_s stands for the centroid of each segment. In this way the macro-grid centroid is determined jointly by all its members. If all pixels are considered in a block (1) generates the exact centroid of the block but here we eliminate outlier segments by an intermediate closing (dilation followed by erosion) step. The result generates different centroids for each block and forms a quadrangle mesh.

2.1.2.2 Macro-grid Label. The second type of feature we extract for the macro-grid is based on segment labels. To represent an image with a concise $N \times N$ grid pattern, each macro-grid is represented by a single label. Each segment uses its global segment index as its label to represent the homogeneity of local regions, and we take the majority voting result in each super-grid as its final label. The resulting image label matrix conveys a certain amount of information regarding coarse structures of the scene.

2.1.2.3 Gabor Filter Responses. The features presented thus far are mainly from color and texture of local image areas. As we know, long straight lines in an image could form strong perspective geometry cues that indicate camera pose and scene orientation. Instead of performing edge detection and camera calibration, we employ a

set of Gabor filters responses to capture this information. In the proposed work Gabor filters were constructed at six different frequencies in nine different orientations. As a result, 54 filter responses for each macro-grid is collected. The reason we implement this feature in this level is twofold. First of all, it is hard to estimate filters responses given the irregular shapes of segments. Secondly, applying the filter to the entire image does not provide information regarding local regions.

Table 1. Feature components for an image

Feature Descriptions	Dimension
Color-RGB	$3 \times N^2$
Color-HSV	$3 \times N^2$
Location	$2 \times N^2$
Area Ratio	N^2
Location Mode	$2 \times N^2$
Gradient (X & Y)	$2 \times N^2$
Gabor Filter Response	$54 \times N^2$
Block Interactions	$3 \times N$

2.1.3 Image Features

One virtue of the adaptation of macro-grid is a uniform representation of image features. As is shown in Table 1, by concatenating feature from each macro-grid sequentially, we obtain a global vector representing the feature vector of all regions from an entire image. Meanwhile, the main draw back this scheme introduces is the loss of spatial information. Though strictly speaking, the sequence of concatenation does represent the spatial information in an implicit way, the interaction among different regions in the image is certainly not sufficiently represented. We introduce macro-grid interactions to compensate for this problem.

2.2 Learning and Retrieval

In the segment level we perform clustering to simplify the representation of image segments, which directly leads to a uniform representation of scene composition by an $N \times N$ macro-grid pattern. Having obtained this representation of images the retrieval engine performs ranking of the similarity measurement between images in the database and any query image and provide results based on this ranking.

2.2.1 Hierarchical Clustering of Segments

As introduced in Section 3.1, we obtain visual features on the segment level after which clustering is performed. For our dataset, partitional clustering methods such as K-means have shown unstable convergence due to strong influence of initial centroid location. In comparison, hierarchical clustering demonstrated more reliable performance. In our case we adopted agglomerative hierarchical clustering scheme, which takes a “bottom-up” approach by assigning one cluster for each segment and keeping merging the most similar clusters into one larger cluster until all data points are eventually merged into one big cluster. Another advantage of hierarchical clustering is that

this sequential approach provides clustering results of different I to K cluster numbers in just one run.

We choose cosine similarity as the distance metric and average linkage as the linkage criteria.

$$D_{KL} = \frac{1}{n_K n_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j) \quad (3)$$

Here n_K and n_L stand for the number of data points in cluster K and cluster L respectively, and $d(x_i, x_j)$ stands for the distance between two data points. D_{KL} represents the average distance from one cluster to the another. With this measure we can minimize errors introduced by either outliers or spatial adjacency of two clusters.

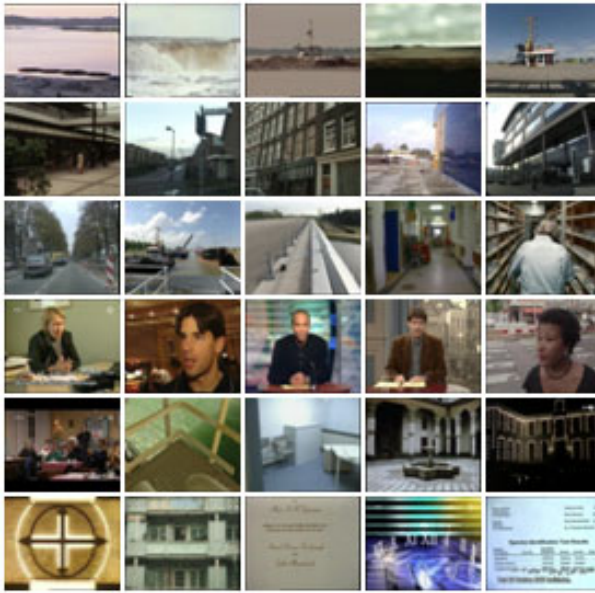


Fig. 3. Sample frames from TV recordings [11]. Columns from left to right stands for each category: Straight background, Sided background, Box, Person+Background, Corner and No Depth.

The advantage of cosine distance is that it automatically provides normalized distance value. The average linkage criteria are chosen to eliminate the concern of outlier within segment clusters. Besides the fact that it is less influenced by extreme values, these criteria also demonstrate the characteristics of favoring clusters with similar variance in general.

2.2.2 Retrieval Scheme

Performing feature extraction on each image in the database, a pool of image represented by their feature is obtained. The retrieval engine works in the following

way: when a query image is received, feature extraction is performed on the query image and pair-wise distances between this image and every image in the database are calculated.

We adopted Levenstein distance for label features which consist of symbolic values. Levenstein distance captures the longest similar sequence in two strings, in our application this improves the robustness of matching similar structure in different part of an images.



Fig. 4. Macro-grid label representation of image content

3 Experiments

To demonstrate the performance of the proposed approach, first we explain the data acquisition and show sample data for each category. Then we present the macro-grid representation of images. Finally we present result for image retrieval based on the representation.

3.1 Data Acquisition

The main source of experimental data is the NIST TRECVID database [11] containing a variety of TV recordings. Out of over 600 hours of video, we obtained 50,000 sample frames and manually labeled 4,000 frames obtaining roughly equal amount of samples per category. Sample frames from each category are illustrated in Figure 3.

3.2 Macro-grid Representation of Images

First we present some sample output of our macro-grid label representation. As is shown in Figure 4, images were divided into $N \times N$ blocks (in this case $N = 5$) and labeled according to clustering result represented by different gray scale. It

Table 2. Mean Reciprocal Rank and Precision

Category	Prior (%)	MRR	P@5	P@10	P@20
Straight Background	16.3	1.0	.80	.80	.65
Tilted Background	16.0	2.5	.60	.70	.45
Box	16.3	1.5	.60	.50	.55
Corner	9.4	3.0	.40	.30	.20
Person + Background	25.7	1.0	1.00	1.00	.95
No Depth	16.2	1.0	.80	.40	.35
Average	16.7	1.67	.70	.62	.53

can be observed that despite the low resolution of this representation, a certain amount of information regarding key component/plane separation is captured by the labels.

3.3 Retrieval Evaluation

For each query image, the retrieval engine returns a list of similar images following decreasing similarity ranking values regarding the original image. When a returned image has the same category type as the query image, this is defined as a hit. The performance of the retrieval system is measured by the average rank of the first hit, and the amount of hit given a certain size of returned images. In our experiment we employ two metrics to evaluate the retrieval performance.

Mean Reciprocal Rank (MRR): Given an input image and a set of retrieved images, *MRR* measures the position of the first relevant image in the returned results averaged across all trials of experiment.

Precision at k ($P@k$): The average precision when k images are returned as retrieved result.

In our experiment we measure $P@5$, $P@10$ and $P@20$. The $P@K$ value is visualized in the following chart.

It can be observed from Table 2 that the *MRR* measurement varies from class to class. The average *MRR* suggests the ranking that on average a “hit” occurs. We can also see that the $P@k$ value decreases as the number of retrieved images (k) increases. Note that among all the categories, the “Corner” and “Person+Background” yielded the lowest and the highest accuracy respectively. Uneven prior distribution might have been an influencing factor.

The process that a human perceives a pictorial stimuli and determines the spatial layout involves complex procedures that engage heavy cognitive processing. The problem our system addresses is technically challenging. By introducing predefined categories of image scenes we added constraints to the problem and simplified the scenario. Although the performance of the system can still be improved, our results have demonstrated that the proposed approach is capable of capturing underlying 3D information for image retrieval with a relatively simple feature set.

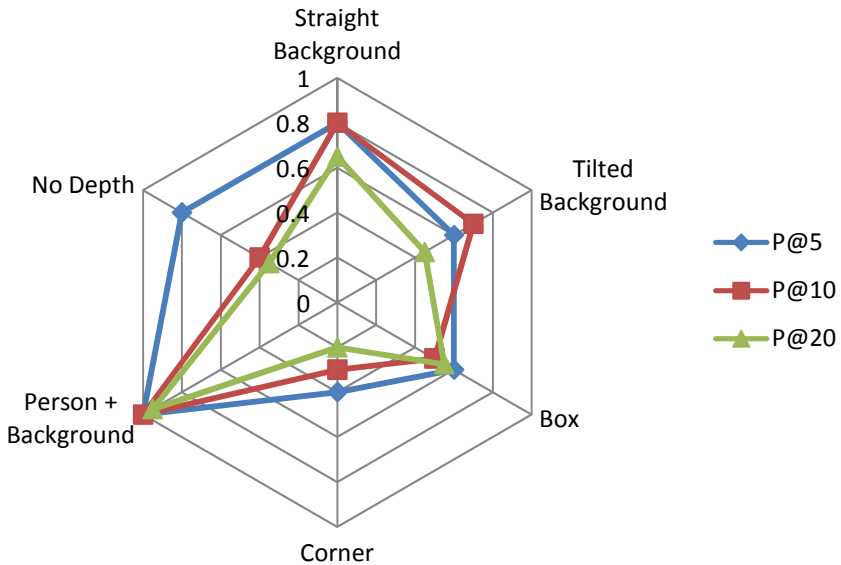


Fig. 5. Precision at K distribution across scene categories

4 Conclusion and Future Work

In this paper we presented a new approach for 3D image retrieval. By dividing image into 5×5 macro-grids and performing tri-layer feature extraction we obtained an image representation that incorporates information on three levels. Given the small amount of information represented in the feature set, the proposed approach demonstrated effective retrieval of images based on inherent 3D information. We believe that the proposed could serve as a pre-processing step for other applications such as image categorization and more precise 3D reconstruction. The results suggest that bridging high-level semantic representation with low-level statistical representation of images is a promising direction for further pursuit.

Another area that would be interesting to study is the robustness of this retrieval framework in terms of spatial invariance. In our experiment, the nature of our data source (TV recordings) already introduced significant amount of camera translation/rotation, and no rectification was performed to our data before processing. At the same time images with salient perspective angles would have a stronger response. Thus it would be interesting to examine whether global or local features dominate the retrieval result if after we apply a random projective transform and to what extent would the results be polluted.

We also identified several potential limitations of the proposed approach such as static macro-grid resolution. The tradeoff between coarse resolution which is associated with greater information loss and fine resolution which leads to increased approximation within each region and dimension increase should be further examined. Another aspect of the problem which is not sufficiently addressed here is the interaction between image regions. Although our feature vector incorporates this feature to a

certain extent, we anticipate more carefully designed distance metric and interaction scheme to bring the performance to a higher level.

Acknowledgement

The authors were partially supported during this work by an NSF grant (Award # 0845469), which is greatly appreciated.

References

- [1] Douglas Cooper, *The Cubist Epoch*, ISBN 0 87587041 4
- [2] Felzenszwalb, P., Huttenlocher, D.P.: Efficient graph-based image segmentation. *IJCV* 59(2) (2004)
- [3] Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003) ISBN 0-521-54051-8
- [4] Hoiem, D., Efros, A.A., Hebert, M.: Automatic Photo Pop-up. In: *ACM SIGGRAPH* (2005)
- [5] Hoiem, D., Efros, A.A., Hebert, M.: Geometric Context from a Single Image. In: *International Conference of Computer Vision (ICCV)* (October 2005)
- [6] McCullagh, P., Nelder, J.A.: *Generalized Linear Models*, 2nd edn. Chapman & Hall/CRC Press (1990)
- [7] Nedović, V., Smeulders, A.W.M., Redert, A., Geusebroek, J.-M.: Depth Information by Stage Classification. In: *Proc. of the 11th IEEE Int'l Conf. on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil, October 14-20 (2007)
- [8] Nedović, V., Smeulders, A.W.M., Redert, A., Geusebroek, J.-M.: Stages as Models of Scene Geometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32(9) (2010)
- [9] Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3-D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI (2008)
- [10] Saxena, A., Sun, M., Ng, A.Y.: Make3D: Depth Perception from a Single Still Image. In: *AAAI* (2008)
- [11] TRECVID2007-09 Sound and Vision Data, <http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html>

An Analysis-by-Synthesis Approach to Rope Condition Monitoring

Esther-Sabrina Wacker and Joachim Denzler

Chair for Computer Vision, Friedrich Schiller University of Jena
{[esther.wacker](mailto:esther.wacker@uni-jena.de), [joachim.denzler](mailto:joachim.denzler@uni-jena.de)}@uni-jena.de
<http://www.inf-cv.uni-jena.de>

Abstract. A regular rope quality inspection is compulsory for wire ropes in security-relevant applications. Principal procedures of such quality checks are the visual inspection for surface defect detection, the magnetic inspection for a localization of broken wires and the measurement of the rope diameter. However, until today it is hardly possible for the human inspector to measure other important rope characteristics as the lay length of wires and strands over time. To close this gap, we present a model-based approach for rope parameter estimation. The usage of a theoretically correct and regular 3d rope, embedded in an analysis-by-synthesis framework, allows a purely image-based monitoring of important rope parameters. Beyond that, also a quantification of the degree of abnormality becomes possible. Our evaluation on real-world and synthetic reference data demonstrates that the approach allows a measurement of the individual lay lengths of wires and strands up to an accuracy more precise than 1 mm.

1 Introduction

Automatic visual inspection is an arising field of interest. Especially in scenarios, which imply a high risk for the human life or which demand utterly concentration over a long time period, an automatic support is highly appreciated. The inspection of wire ropes used for ropeways, elevators or bridges is one example of such a task. As they are used in security-relevant application areas, a regular quality check is obligatory. Until today, such a quality check is composed of two different parts: the magnetic inspection of wire ropes allows a detection of interrupts or deformations within the wire structure up to a certain extent. However, this method is limited to defects which change the magnetic signal of the wire course. For this reason, the second part is normally a visual inspection of the rope surface by human experts, which is meant to identify all visible anomalies like missing and broken wires or corrosion in the surface structure. Anyhow, none of these inspection methods allows a monitoring of main rope parameters as for example the lay length of strands and wires. And even for a human expert it is an almost impossible challenge to observe creeping changes in the main rope structure, since these variations are not visible to the naked eye. The only way to achieve this would be a manual and repeated measurement which is imprecise

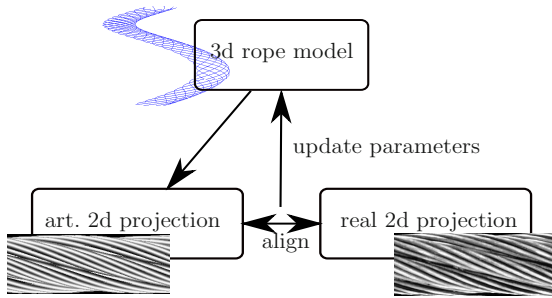


Fig. 1. The general analysis-by-synthesis framework for rope condition monitoring

and time-consuming. To the best of our knowledge, until today there exists no automatic approach, which provides measurements for the main rope parameters. Nevertheless, a monitoring of these variables could give important and so far unused information about the rope condition.

On this account, we introduce a new, model-based approach for rope condition monitoring embedded in an analysis-by-synthesis framework. The general process chain of our approach is sketched in Fig. 1. Given just the 2d projections of the real-world rope, they are aligned with a perfectly regular and theoretically founded, parametric 3d model. This 3d rope model will be introduced in Sect. 3.1. The 2d/3d alignment is performed by registering an artificially generated 2d projection of the model with the real rope projection. The synthesis procedure for generating artificial 2d projections is introduced in Sect. 3.2. The rope parameters are then updated in an iterative fashion, based on a similarity measure which allows a comparison of the real and synthetic projections.

The periodic rope structure makes great demands on this registration procedure, as it turns the alignment into a complex and highly ambiguous problem. To solve for this problem and to provide an automatic, accurate and combined estimation of individual rope parameters is the main contribution of this work. This optimization strategy is explained in detail in Sect. 3.3. The experiments in Sect. 4 account for the applicability of the presented approach as it provides a very accurate estimation of the actual rope characteristics and enables a surveillance of individual fabrication components. Last but not least, method and payoff are summarized and discussed in Sect. 5.

2 Related Work

Rope Inspection. Most of the work in the field of automatic rope inspection normally makes use of magnetic measurement techniques as described for example in [1] to identify abnormal changes in the specific signal of the wire course. There exists few work coping with automatic approaches for visual rope inspection. As the most important problem in this context is a lack of missing

defective examples for supervised learning strategies, Platzer et al. present a one-class classification approach to surface defect detection in wire ropes based on different features [2]. In [3] the same authors make use of Hidden Markov models to solve the problem of defect localization in wire ropes. However, none of these approaches focuses on an automatic estimation of meaningful rope parameters, which would close the gap between all the different inspection techniques.

Analysis-by-Synthesis. Our approach deals with the problem of image-based (2d-based) estimation of time-variant (dynamic) scene parameters. A well-known approach to this problem is analysis-by-synthesis. The parametrized 3d model is adapted to the 2d input image by minimizing the measurable difference between this input image and a synthetically generated projection of the 3d model. The best alignment of model and scene finally results in the parameter estimates.

Typical application examples for analysis-by-synthesis include camera calibration [4] and camera pose estimation [5,6], human motion analysis [7] and (object) tracking applications [8]. A similar concept is used in medical applications such as 2d-3d image registration, where digitally reconstructed radiographs (DRRs) are used to estimate the transformation parameters between 2d images and the corresponding 3d dataset [9].

For rope condition monitoring, analysis-by-synthesis is of particular interest for two reasons: at first, the task of rope condition monitoring can be interpreted as a parameter estimation problem and secondly the alignment with a perfectly regular 3d rope model facilitates the detection and quantification of anomalies with respect to the important rope characteristics.

3 Rope Parameter Estimation

3.1 Parametric 3d Rope Model

The basic prerequisite for the estimation of 3d scene parameters is a parametric 3d rope model. A description of the general rope geometry of wire ropes can be found in many specific literature [10,11]. As for the image-based analysis just the 2d volumetric appearance of the wires is of relevance, a simplified 3d wire centerline model is used for our scope.

In general, a stranded rope consists of wires, which are organized in strands. These strands, in turn, form the entire rope. In the top left of Fig. 2 the fundamental construction of a wire rope is shown: A certain number of strands (big circles) are grouped around the rope core (gray shaded) and each strand consists of an also fixed number of wires (small circles). Both, strands and wires can be described by helix-shaped space curves around the rope or rather strand axis. Taking this into account, we can build a 3d model describing the wire centerlines of the rope. A wire centerline $\mathbf{W}_{i,j}$ of wire i in strand j can be described by two intertwined helices, whereas the first helix \mathbf{S}_j describes the space curve of strand j and \mathbf{W}_i represents the space curve of the wire i . As a line camera is used for the acquisition of the real rope, the rope model is rotated around the

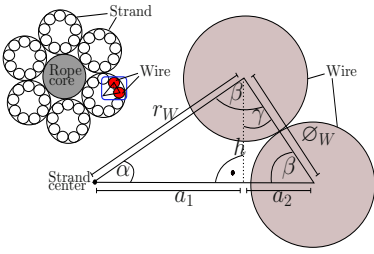


Fig. 2. General rope geometry (top left) and wire geometry (right)

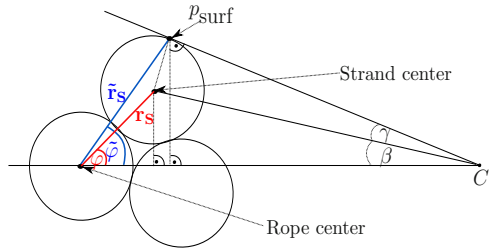


Fig. 3. Geometry for the computation of the model contour of a volumetric rope projection

y-axis to align the rope axis (time axis) with the x-axis of the camera coordinate system. So, the wire i in strand j can be described as:

$$\begin{aligned}
 \mathbf{W}_{i,j}(\mathbf{p}_S, \mathbf{p}_{W_j}, t) = & \quad (1) \\
 & \underbrace{\begin{pmatrix} t \\ r_S \sin(t \frac{2\pi}{L_S} + jk_S + o_S) \\ -r_S \cos(t \frac{2\pi}{L_S} + jk_S + o_S) \end{pmatrix}}_{\mathbf{S}_j} + \underbrace{\begin{pmatrix} 0 \\ r_W \sin(t \frac{2\pi}{L_{W_j}} + ik_W + o_{W_j}) \\ -r_W \cos(t \frac{2\pi}{L_{W_j}} + ik_W + o_{W_j}) \end{pmatrix}}_{\mathbf{W}_i}.
 \end{aligned}$$

Thereby, r_S and r_W are the fix radii of the strand and wire space curves. $jk_S = j \frac{2\pi}{\#S}$ denotes the phase displacement for the j -th strand in the rope and $ik_W = i \frac{2\pi}{\#W}$ is the phase displacement of the i -th wire in each individual strand. $\#S$ is the number of strands in the rope and $\#W$ the number of wires in a strand respectively. Given these fixed values, the rope can be parametrized by the remaining free and dynamic parameters which are organized in the parameter vectors $\mathbf{p}_S = (L_S, o_S)$ and $\mathbf{p}_{W_j} = (L_{W_j}, o_{W_j})$. They contain the lay lengths of strands L_S and the lay lengths of wires in the j -th strand L_{W_j} as well as their position o_S and o_{W_j} with respect to the corresponding periods.

The fix radii of the strand and wire space curves r_S, r_W can be computed based on the rope specification. Such a specification normally contains information about the rope and wire diameters $\varnothing_R, \varnothing_W$ as well as the number of strands per rope $\#S$ and wires per strand $\#W$. In Fig. 2 the basic rope geometry is sketched. Applying standard trigonometric operations on this geometry allows the formulation of the radius of the wire space curve as $r_W = \frac{\sin(\beta)\varnothing_W}{\sin(\alpha)}$. The computations for r_S are analog.

3.2 2d Image Synthesis

Given the 3d rope model from Sect. 3.1 an artificial 2d projection can be computed. Since the rope acquisition is done with line cameras [12], also known as pushbroom cameras, the specific projection geometry of this 1d sensor type must be taken into account. The projection matrix for linear pushbroom cameras, which leads to a perspective projection along the sensor array and to an

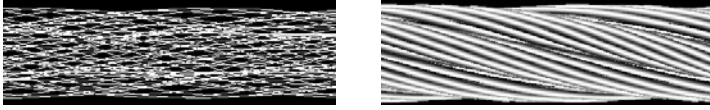


Fig. 4. Synthetic wire centerline projection (left) and volumetric wire projection (right)

orthographic projection along the time axis, is derived by Gupta and Hartley in [13] under the prerequisite of constant and linear camera motion. In the rope acquisition procedure this is fulfilled, as the camera moves with a constant velocity along the rope axis. According to Gupta and Hartley the pushbroom projection of a 3d rope point $\mathbf{W}_{i,j}(\mathbf{p}_S, \mathbf{p}_{W_j}, t)$ to a 2d point (u, v) can be written as:

$$\begin{pmatrix} u \\ v \end{pmatrix} \leftarrow \begin{pmatrix} u \\ wv \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & f & p_v \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} \frac{1}{V_x} & 0 & 0 \\ \frac{-V_y}{V_x} & 1 & 0 \\ \frac{-V_z}{V_x} & 0 & 1 \end{pmatrix}}_{\mathbf{V}} \left(\mathbf{W}_{i,j}(\mathbf{p}_S, \mathbf{p}_{W_j}, t) + \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \right). \quad (2)$$

Here, d is the unknown but fix camera-to-scene distance. Therefore, the parameter vector \mathbf{p}_S is extended by the further free parameter d . We assume that the camera moves one camera line per time step along the rope axis implying $V_x = 1$ and $V_y = V_z = 0$. Also, we have no specific calibration data for the camera, but the 3d rope diameter \varnothing_R as well as the rope diameter in pixels in the real 2d projection \varnothing_R^{2d} are known. Accordingly, a 3d rope point is projected with a focal-length $f = 1$ and a projection center $p_v = 0$. Afterwards, the resulting projection is rescaled to the given 2d rope diameter in pixels. This allows an optimization of the ratio between f and d . The projection of a 3d rope point $\mathbf{W}_{i,j}(\mathbf{p}_S, \mathbf{p}_{W_j}, t)$ in Cartesian coordinates can be given combining (1) and (2):

$$(u, v)^T = \left(t, \frac{r_S \sin(t \frac{2\pi}{L_S} + jk_S + o_S) + r_W \sin(t \frac{2\pi}{L_{W_j}} + ik_W + o_{W_j})}{-(r_S \cos(t \frac{2\pi}{L_S} + jk_S + o_S) + r_W \cos(t \frac{2\pi}{L_{W_j}} + ik_W + o_{W_j})) + d} \right)^T. \quad (3)$$

An exemplary 2d wire centerline projection is depicted in the left of Fig. 4. The volumetric wire appearance, which is needed for the alignment process, can be approximated by centering a 1d Gaussian around each projected wire centerline pixel for every time step. The width of the Gaussian mask was based on the projected wire diameter \varnothing_W^{2d} in pixel. A Gaussian seems to be a good choice, as the uncertainty for wire presence slabs with the distance of the pixel to its centerline. The volumetric projection can be seen in the right image of Fig. 4.

3.3 Estimation of Rope Parameters

The parameter optimization is performed in a two-step manner: at first, the strands of the rope model and the real projection are aligned by optimizing the strand parameters included in \mathbf{p}_S . Afterwards, the wires of each individual strand are aligned, leading to the estimates for the wire parameters \mathbf{p}_{W_j} of strand j .

A good strategy for a combined alignment of all rope strands is based on the rope contour, as it contains all necessary information. The upper and lower rope contours $c_u^r(t)$ and $c_l^r(t)$ of the real rope are automatically extracted from the real input image. For the upper and lower model contours in the synthetic 2d projection an analytical description can be derived. Figure 3 shows the underlying geometry. The strand center, defined through the angle given by $\varphi_S = t \frac{2\pi}{L_S} + jk_S + o_S$ and the radius of the strand space curve r_S (both marked in red), will lead to a contour point in the 2d *centerline* projection. However, to describe the 3d point p_{surf} , which will lead to a contour point of the *volumetric* 2d rope projection, a new angle $\tilde{\varphi}_S$ and a new distance \tilde{r}_S (marked in blue) can be computed with help of trigonometric operations.

The computation of the 2d projection of a strand centerline \mathcal{S}_j (see (II)) with regard to the projection geometry introduced in Sect. 3.2 is straight forward. Hence, we just denote the modified formula using $\tilde{\varphi}_S$, \tilde{r}_S instead of φ_S , r_S :

$$\tilde{S}_j^{2d}(\mathbf{p}_S, t) = \left(\begin{array}{c} t \\ \frac{\tilde{r}_S \sin(\tilde{\varphi}_S)}{-(\tilde{r}_S \cos(\tilde{\varphi}_S)) + d} \end{array} \right) \quad (4)$$

A minimum/maximum operation on the y-coordinates \tilde{S}_j^{2d} of (4) for all strands in a *time frame* $T = [t_1, t_2]$ leads to the rope contour of the volumetric model:

$$c_u^m(\mathbf{p}_S, t) = \min_j \tilde{S}_j^{2d}(\mathbf{p}_S, t), \quad \forall t \in T \quad (5)$$

$$c_l^m(\mathbf{p}_S, t) = \max_j \tilde{S}_j^{2d}(\mathbf{p}_S, t), \quad \forall t \in T. \quad (6)$$

The optimization of the parameter vector \mathbf{p}_S is performed by evaluating the normalized, 1d cross correlation coefficient $NCC_T^{1d}[\cdot, \cdot]$ of both given contours:

$$\hat{\mathbf{p}}_S = \arg \max_{\mathbf{p}_S} NCC_T^{1d}[c_u^r(t), c_u^m(\mathbf{p}_S, t)] + NCC_T^{1d}[c_l^r(t), c_l^m(\mathbf{p}_S, t)]. \quad (7)$$

After the determination of the strand parameters, the wire parameters \mathbf{p}_{W_j} are estimated in a similar fashion. The optimization procedure is carried out separately for the wires of each individual strand, as their parameters need not necessarily be equal. Instead of the 1d normalized correlation coefficient its 2d counterpart $NCC_T^{2d}[\cdot, \cdot]$ is used to align the image data of real and synthetic rope projections $\mathcal{I}_{real}, \mathcal{I}_{syn}$ for the time frame T :

$$\hat{\mathbf{p}}_{W_j} = \arg \max_{\mathbf{p}_{W_j}} NCC_T^{2d}[\mathcal{I}_{real}, \mathcal{I}_{syn}(\hat{\mathbf{p}}_S, \mathbf{p}_{W_j}, t)]. \quad (8)$$

Both optimization steps make use of the Downhill Simplex optimization scheme [14], whereas a global grid search is put in front of the wire alignment step. This is necessary, as the wire alignment is a highly periodic and therefore ambiguous problem which demands for a good initialization. Although only the strand lay length and the individual wire lay lengths are of interest for the monitoring task, the remaining parameters have to be estimated to achieve a time-variant alignment of strands and wires.

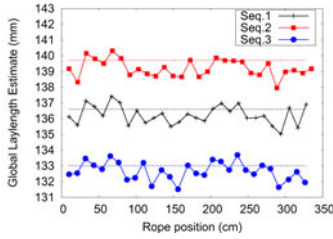


Fig. 5. Strand lay length estimation on the three different real data sequences

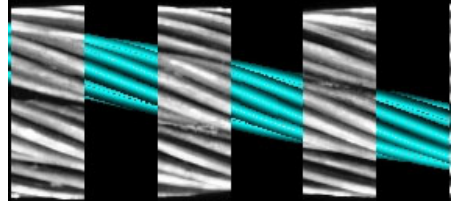


Fig. 6. Registration result: reference (all strands) and model projection are displayed in an alternating fashion

4 Experimental Evaluation

In this section the presented method is validated on real rope data sets. To the best of our knowledge this is the first approach, which allows an automatic estimation of the stated rope parameters. Therefore, we are not able to compare with other approaches. Moreover, a manual measurement of a real rope would be too time-consuming and imprecise, so that it is not possible to compute an estimation error on real-world data. Hence, the quantitative evaluation of the accuracy is assured additionally with help of a simulated ground truth data set. Certainly, this synthetic reference data is computed based on the same ropemodel. Nevertheless, the segmentation and extraction of the rope contour from the reference projection is the critical part of the analysis, which forms the foundation for the parameter estimation. Thus, at first we prove the functional efficiency on real-world data. In the synthetic experiments we then focus on the tracking ability of our approach with respect to creeping parameter variations, as they are a matter of particular interest for (visual) rope inspection.

Real Rope Datasets. The acquired 6×19 seal rope consists of six strands and nine visible, outer wires. It has a length of approximately 3.4 meters, a diameter of 20.46 mm and an expected strand lay length of 137.5 mm. One camera line corresponds to 0.1 mm of rope. Note, that the expected lay length is just an initial guess. It varies around a few millimeters even for intact ropes.

Three different 2d input sequences of the same rope are used. The first one, denoted by **Seq.1**, is the reference sequence. For **Seq.3** the rope was manually untwisted and for **Seq.2** it was re-twisted. These manipulations should primarily result in an altered strand lay length. The parameter tracking results for the strand lay length of these three sequences are displayed in Fig. 5. The manually measured strand reference lay length for some exemplary frames of each sequence is given by the dashed line. This value is 136.6 mm for **Seq.1**, 139.5 mm for **Seq.2** and 133.0 mm. for **Seq.3**. Fig. 5 reveals the following outcomes: at first the approach is capable of clearly identifying the difference in the strand lay length of the three different sequences. Secondly, the variation in the estimation results over time varies around ± 1 mm which is less than one percent of the



Fig. 7. Backprojected Strand (white stripes) into the original rope projection

measured value. Last but not least, it becomes clear, that the lay length is a time-variant, dynamic parameter for which it is hard to define a reference value by manual measurement. Furthermore, the reader’s attention should be drawn to the visible correlation in the lay length course of the three different sequences. This is a further indicator for the quality of our estimation results. The variation coefficient (standard deviation/mean) for 10 different estimation runs is 0.0028% for the strand lay length and 3.3% for the wire lay lengths.

Furthermore, the quality of the parameter estimates can be evaluated by backprojecting the model strands and wires to the original 2d projection of the real rope. This allows a visual judgment of the results. Fig. 7 displays the backprojected strand whereas Fig. 6 displays the registration result for the wires of one central strand (blue). In both cases a checkerboard representation, which displays reference image and the backprojected rope model in an alternating fashion (gray and white/colored stripes) is used. As one can see, the strand alignment as well as the wire alignment is very precise. Most of the crossings between real and synthetically generated wires fit almost perfectly together. In the few cases, where the crossings between the wires are not seamless, this can be justified by the fact, that the rope model is a perfectly regular structure whereas this does not hold for the real rope due to manufacturing tolerances.

The computation times achieved on an Intel Core2 (2 MHz) vary around 230 s / m for strand and wire alignment. Note, that currently this is a non-GPU implementation, so that there is a lot of space for performance improvement.

Simulated Ground Truth Data. The simulated rope is composed of six strands and nine wires. For each individual testrun we simulate 30 m rope. The temporal resolution is 0.1 mm per camera line. The ground truth values of all free parameters for every individual testrun are randomly chosen with strand lay lengths from 123 mm to 152 mm and wire lay lengths from 59 mm to 99 mm.

To evaluate the accuracy of the presented approach Gaussian noise with different noiselevels is added to the rope contours and the grayvalues of the ground truth rope projection. The resulting error distributions are visualized by means of boxplots [15]. In Fig. 8 the boxes depict the 0.25 and 0.75 quantiles and the middle bar marks the median error in millimeters obtained for all time steps of 20 randomly initialized test runs per noise level. The left plot shows the results for the strand parameter L_S and the right image illustrates the error distribution for the wire lay length of an exemplary chosen strand. The maximum position errors are around 1.3 mm and the camera-to-scene distance d can be measured with a mean accuracy of ~ 3.7 mm. Although this high accuracy is obtained with respect to a simulated test data set, these results prove the functional capability of our approach.

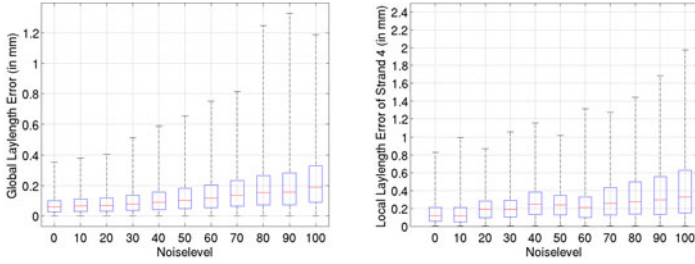


Fig. 8. Boxplots showing the robustness of the lay length estimates of strands (left) and wires (right) to noise

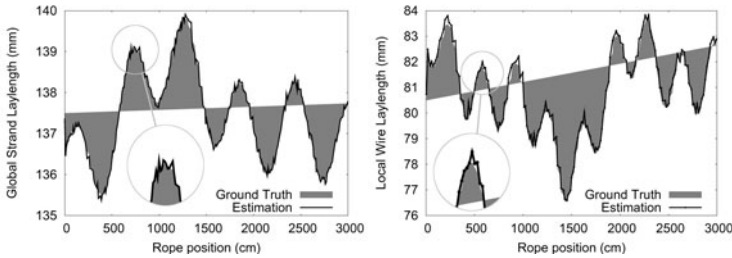


Fig. 9. Parameter tracking of lay lengths for strands (left) and wires (right). The gray curve represents the ground truth parameter progression, the black one the estimates.

In order to prove the capability of the approach to track parameter variations, a randomly generated parameter progression is precomputed for every individual lay length parameter. This progression is marked by the gray curve in the plot of Fig. 9. The estimates for every frame resulting from the analysis-by-synthesis loop are represented by the solid, black curve. The high accuracy is also reflected in the low mean errors obtained in 100 randomly initialized test runs. These are 0.06 mm for the strand lay length and 0.23 mm for the wire lay lengths.

5 Conclusions and Outlook

An new approach to image-based rope condition monitoring was presented. With help of an analysis-by-synthesis framework, important rope characteristics as the lay lengths of wires and strands can be measured and a quantification of anomalies becomes possible. Experiments on simulated ground truth data reveal a high accuracy with worst-case estimation errors around 2 mm in the presence of noise. Beyond that, the applicability to real data is attested by the almost perfect alignment of wires of synthetic and real projections. To the best of our knowledge this is the first approach allowing a combined and automatic estimation of strand and wire parameters and thus provides an univocal mapping of a real 2d rope projection and a theoretically perfect 3d rope model.

Besides the parameter monitoring task, an image-based comparison of real and synthetic projections also can be a great benefit with respect to surface defect detection in wire ropes due to the direct comparability of the appearance of each arbitrary wire. This will be the focus of future work in order to provide an exhaustive methodology for an all-embracing visual rope inspection.

References

1. Zhang, D.L., Cao, Y.N., Wang, C., Xu, D.G.: A New Method of Defects Identification for Wire Rope Based on Three-Dimensional Magnetic Flux Leakage. *Journal of Physics: Conference Series* 48, 334–338 (2006)
2. Platzer, E.-S., Süße, H., Nägele, J., Wehking, K.-H., Denzler, J.: On the Suitability of Different Features for Anomaly Detection in Wire Ropes. In: *Computer Vision, Imaging and Computer Graphics: Theory and Applications* (Springer CCIS), pp. 296–308. Springer, Heidelberg (2010)
3. Platzer, E.-S., Nägele, J., Wehking, K.-H., Denzler, J.: HMM-Based Defect Localization in Wire Ropes - A New Approach to Unusual Subsequence Recognition. In: Denzler, J., Notni, G., Süße, H. (eds.) *Pattern Recognition. LNCS*, vol. 5748, pp. 442–451. Springer, Heidelberg (2009)
4. Eisert, P.: Model-based Camera Calibration Using Analysis by Synthesis Techniques. In: *Proceedings of the Vision, Modeling, and Visualization Conference (VMV)*, Aka GmbH, pp. 307–314 (2002)
5. Koeser, K., Bartczak, B., Koch, R.: An Analysis-by-Synthesis Camera Tracking Approach Based on Free-Form Surfaces. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) *DAGM 2007. LNCS*, vol. 4713, pp. 122–131. Springer, Heidelberg (2007)
6. Wuest, H., Wientapper, F., Stricker, D.: Adaptable Model-Based Tracking Using Analysis-by-Synthesis Techniques. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) *CAIP 2007. LNCS*, vol. 4673, pp. 20–27. Springer, Heidelberg (2007)
7. Moeslund, T.B., Hilton, A., Krüger, V.: A Survey of Advances in Vision-based Human Motion Capture and Analysis. *Computer Vision and Image Understanding* 104, 90–126 (2006)
8. Hasler, N., Rosenhahn, B., Asbach, M., Ohm, J.-R., Seidel, H.-P.: An Analysis-by-Synthesis Approach to Tracking of Textiles. In: *Proceedings of the IEEE Workshop on Motion and Video Computing*, p. 25. IEEE Computer Society, Los Alamitos (2007)
9. Penney, G.P., Weese, J., Little, J.A., Desmedt, P., Hill, D., Hawkes, D.J.: A comparison of similarity measures for use in 2-D-3-D medical image registration. *IEEE Transactions on Medical Imaging* 17, 586–595 (1998)
10. Feyrer, K.: *Wire Ropes: Tension, Endurance, Reliability*. Springer, Berlin (2007)
11. Shitkow, D.G., Pospechow, I.T.: *Drahtseile*. VEB Verlag Technik Berlin (1957)
12. Moll, D.: Innovative procedure for visual rope inspection. *Lift Report* 29, 10–14 (2003)
13. Gupta, R., Hartley, R.: Linear Pushbroom Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 963–975 (1997)
14. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *Numerical Recipes in C*, 2nd edn. Cambridge University Press, Cambridge (1992)
15. McGill, R., Tukey, J., Larsen, W.A.: Variations of Boxplots. *The American Statistician* 32, 12–16 (1978)

Fast Parallel Model Estimation on the Cell Broadband Engine

Ali Khalili¹, Amir Fijany¹, Fouzhan Hosseini¹, Saeed Safari^{1,2},
and Jean-Guy Fontaine¹

¹ Italian Institute of Technology, Genova, Italy

² School of Electrical and Computer Engineering, University of Tehran, Iran
{ali.khalili, amir.fijany, fouzhan.hosseini,
saeed.safari, jean-guy.fontaine}@iit.it

Abstract. In this paper, we present fast parallel implementations of the RANSAC algorithm on the Cell processor, a multicore SIMD architecture. We present our developed strategies for efficient parallel implementation of the RANSAC algorithm by exploiting the specific features of the Cell processor. We also discuss our new method for model generation to increase the efficiency of calculation of the Homography transformation by RANSAC. In fact, by using this new method and change of algorithm, we have been able to increase the overall performance by a factor of almost 3. We also discuss in details our approaches for further increasing the efficiency by a careful vectorization of the computation as well as by reducing the communication overhead by overlapping computation and communication. The results of our practical implementations clearly demonstrate that a very high sustained computational performance (in terms of sustained GFLOPS) can be achieved with a minimum of communication overhead, resulting in a capability of real-time generation and evaluation of a very large number of models. With a data set of size 2048 data and a number of 256 models, we have achieved the performance of over 80 sustained GFLOPS. Since the peak computing power of our target architecture is 179 GFLOPS, this represents a sustained performance of about 44% of the peak power, indicating the efficiency of our algorithms and implementations. Our results clearly demonstrate the advantages of parallel implementation of RANSAC on MIMD-SIMD architectures such as Cell processor. They also prove that, by using such a parallel implementation over the sequential one, a problem with a fixed number of iterations (hypothetical models) can be solved much faster leading to a potentially better accuracy of the model.

1 Introduction

Mobile robots and humanoids represent an interesting and challenging example of embedded computing applications. On one hand, in order to achieve a large degree of autonomy and intelligent behavior, these systems require a very significant computational capability to perform various tasks. On the other hand, they are severely limited in terms of size, weight, and particularly power consumption of their embedded computing system since they should carry their

own power supply. The limitation of conventional computing architectures for these types of applications is twofold: first, their low computing power, and the second, their high power consumption. Emerging low-power parallel SIMD and MIMD architectures provide a unique opportunity to overcome these limitations of conventional computing architectures.

Computer vision and image processing techniques are very common in robotic applications. A large number of computer vision applications require a robust estimation algorithm by which a model parameters are obtained from a set of data containing a significant portion of outliers. The RANSAC (RANDOM Sample Consensus) algorithm, originally developed by Fishler and Bolles [1], has become one of the most widely used robust estimators in the field of computer vision [2]. For example, it has been used in applications such as stereo matching [3]. RANSAC is an iterative method to estimate parameters of a certain mathematical model from a set of data which may contain a large number of outliers. It represents a hypothesize-and-verify framework [6]. Each iteration of RANSAC consists of two steps: first, generation of a hypothetical model based on a sample subset of data, and then evaluation of the hypothetical model by using the whole set of data. This iterative procedure is repeated till the probability of finding a better model drops below a certain threshold and the iterations terminate. For many applications, a real-time implementation of RANSAC is indeed desirable. However, its computational complexity represents a major obstacle for achieving such a real-time performance. The computational complexity of RANSAC is a function of the number of required iterations, i.e., the number of generated hypothetical models, and the size of data set. In fact, RANSAC can often find the correct model even for high levels of outliers [6]. However, the number of hypothetical models required to achieve such an exact model increases exponentially, leading to a substantial computational cost [6]. Consequently, there has been significant effort to improve the performance of RANSAC by either reducing the number of models, e.g. [7, 8, 9], or by reducing the size of data set for model evaluation [2, 10].

An efficient alternative to improve the performance of RANSAC is to speed up the computation by exploiting parallelism. Interestingly, however, and to our knowledge, such a parallel implementation has not been extensively and rigorously considered in the literature. In fact, it seems that the only reported work on parallel implementation of RANSAC is [11] wherein a very limited parallelism has been exploited. It considered the implementation of pRANSAM algorithm, a limited parallelization of the RANSAM algorithm [12] which is an enhancement of the original RANSAC, on an Intel multi-core processor. The pRANSAM is implemented on a system equipped with an Intel Core2 Quad processor, which indeed represent a very limited parallel implementation. The reported results show that the achievable speedup depends on both the number of processing nodes and the operating system. One can consider a rather straightforward parallel implementation of RANSAC by exploiting parallelism at each iteration. Note that at each iteration, a same model is evaluated for all the elements of the

data set. This represents a data parallel computation since the evaluation for all the elements of the data set can be performed in parallel. A much more promising approach, presented in this paper, is based on a full parallelization of the whole computation of RANSAC. This approach is motivated by the simple observation that the iterations of RANSAC are, to a large degree, independent and can be performed in parallel. In fact, as will be discussed in Section 2, in a typical RANSAC implementation, at the end of each iteration a check is performed to determine whether more iteration is needed. However, for any practical problem, a large number of iterations is needed before terminating the process. In fact, for many practical real-time problems with a required fixed computation time, a fixed number of iterations is chosen a priori [10].

Our approach to parallel implementation of RANSAC can be considered as a multi-stage process wherein, at each stage, a large number of models are generated and evaluated in parallel. The checking is then performed at the end of each stage to determine whether more stages are needed. Note that, if the number of hypothetical models is fixed a priori or for real-time applications, wherein a fixed computation time is given, then our parallel implementation can be performed in one single stage wherein all the hypothetical models are generated and evaluated in parallel. This approach leads to a massive parallelism in the computation which might be limited only by the resources of the target parallel computing architecture. We consider and discuss the implementation of our parallel RANSAC algorithm on the Cell processor, a multi-core (MIMD-SIMD) architecture. Considering the specific features of the Cell processor, we develop appropriate techniques for efficient implementation to achieve an optimal computational performance with a minimum of overhead. The results of our practical implementations clearly demonstrate that a very high sustained computational performance, in terms of sustained Giga floating operations per second (GFLOPS) can be achieved, resulting in a very fast generation and evaluation of a very large number of models. For Homography estimation, we have achieved a performance of 80 sustained GFLOPS and the capability to generate and evaluate 989 models per millisecond. For a real-time image processing application with 30 frames per second, this represent a capability of generating and evaluating 32966 models per frame. Our results clearly demonstrate the advantages of parallel implementation of RANSAC on SIMD vector processing architectures. They also prove that, by using such a parallel implementation over the sequential one, a problem with a fixed number of iterations (hypothetical models) can be solved much faster and/or for real-time applications, with a fixed given computation time, much more models can be generated and evaluated, leading to a potentially better accuracy of the model.

The rest of this paper is organized as follows. In section 2, the RANSAC algorithm is briefly reviewed. Section 3 describes the overall architecture of Cell Broadband Engine. The parallel implementation of the RANSAC algorithm on the Cell processor is discussed in section 4. And finally, some concluding remarks are given in section 5.

2 RANSAC

As mentioned before, RANSAC has become a fundamental tool in computer vision and image processing applications, and variations to the original algorithm have been presented to improve its speed and accuracy. Despite these modifications, the core of RANSAC algorithm consists of the following two main steps. First, a minimal sample set (MSS) is randomly selected from the dataset. Cardinality of MSS is smallest sufficient number of data to determine model parameters. Then, parameters of the model are computed, using only MSS elements. Then, RANSAC determines the set of data in entire dataset which are consistent with the model and parameters estimated from MSS in the first step. This set of data is called consensus set (CS). These steps are performed iteratively until the probability of finding a better CS drops below a certain threshold and RANSAC terminates.

To describe RANSAC more formally, assume that the dataset, consisting of N elements, is indicated by $D = \{d_1, d_2, \dots, d_N\}$ and that θ denote the parameter vector. Let S denotes a selected MSS, and $err(\theta, d_i)$ be an appropriate function which indicates the error of fitting datum d_i in the model with parameter vector θ . RANSAC checks which elements in D fit in the model. Each datum d_i is considered to fit the model if its fitting error, $err(\theta, d_i)$, is less than a threshold δ . If this is the case, then the datum is added to the consensus set, CS. After that, the CS is compared with the best consensus set CS^* obtained so far. If CS is ranked better than CS^* , best consensus set and best model parameters are updated. The sets CS can be ranked by using various measures. In the original RANSAC [1], consensus set are ranked according to their cardinality. Other measures have also been considered [13].

Finally, the algorithm checks if more iteration is needed. Assume p is the probability of selecting an inlier from dataset D . Thus, the probability of selecting an MSS, named S , that produces an accurate estimation of model parameters will be $p^{\bar{s}}$, where \bar{s} is the cardinality of S . So, the probability of selecting an MSS which contains at least one outlier is $(1 - p^{\bar{s}})$. If the algorithm iterates h times, the probability that all selected MSSs contain outliers is $(1 - p^{\bar{s}})^h$. Consequently, h should be chosen large enough so that $(1 - p^{\bar{s}})^h$ becomes equal or smaller than an acceptable failure threshold ϵ . The required number of iterations, T_{itr} , is then obtained as:

$$T_{itr} = \frac{\log \epsilon}{\log (1 - p^{\bar{s}})} \quad (1)$$

Note that, p is not known a priori. However, a lower bound on the p can be estimated as $\frac{N_I}{N}$, where N_I is the size of CS^* . Estimation of p is then updated as the algorithm progresses [6]. In the following, we discuss the application of RANSAC for Homography estimation.

2.1 Homography Estimation

Homography is a linear transformation in projective space which relates two images of a planner scene, taken from different views by a pin-hole camera. In

the field of computer vision, homography transformation has many applications such as image rectification, image registration, and structure from motion. The homography transform H which maps point d in one image to point d' in the other image ($d(x, y), d'(x', y') \in \mathbb{R}^2$) is given by:

$$d' = H_\theta(d) = \begin{bmatrix} \theta_1 x + \theta_4 y + \theta_7 \\ \theta_3 x + \theta_6 y + \theta_9 \\ \theta_2 x + \theta_5 y + \theta_8 \\ \theta_3 x + \theta_6 y + \theta_9 \end{bmatrix} \quad (2)$$

The goal is to estimate parameters vector θ such that Eq. 2 holds for all inliers. To estimate the parameters vector at least four corresponding points are required. This represents eight degree of freedom while the total number of parameters is nine. A common approach to find the homographic transformation is called Direct Linear Transform (DLT) method and is implemented as follows [15]. Eq. 2 can be rewritten as $A(d_i, d'_i)\theta = 0$ where $A(d_i, d'_i)$ is given by:

$$A(d_i, d'_i) = \begin{bmatrix} x & 0 & -xx' & y & 0 & -yx' & 1 & 0 & -x' \\ 0 & x & -xy' & 0 & y & -yy' & 0 & 1 & -y' \end{bmatrix} \quad (3)$$

To calculate the parameters vector, we can stack one upon the other 2×9 matrices $A(d_i, d'_i)$. Stacking four matrices, we then obtain an 8×9 matrix A_S , given by:

$$A_S = \begin{bmatrix} A(d_{i_1}, d'_{i_1}) \\ A(d_{i_2}, d'_{i_2}) \\ A(d_{i_3}, d'_{i_3}) \\ A(d_{i_4}, d'_{i_4}) \end{bmatrix} \quad (4)$$

The parameters vector, $\Theta = \{\theta_1, \theta_2, \dots, \theta_9\}^T$, is then the eigenvector corresponding to the smallest eigenvalue of $A_S^T A_S$ [15], which is usually obtained by SVD decomposition [14]. The error function then associated with the two corresponding points d_i and d'_i is determined as:

$$err(\theta, d_i, d'_i) = |d'_i - H(d_i)|^2 + |d_i - H^{-1}(d'_i)|^2 \quad (5)$$

For computing the eigenvalues of the symmetric matrix M , we used the Jacobi eigenvalue method which is a simple yet efficient approach for finding the eigenvalues and eigenvectors of matrices with small and moderate order. Jacobi uses a plane rotation $M' = P_{pq}^T M P_{pq}$. Typical matrices require six to ten sweeps to achieve the convergence or $3n^2$ to $5n^2$ Jacobi rotation which means $24n^3$ to $40n^3$ floating point operations (8 operations per each iteration) [14]. At the end of the algorithm, the diagonal elements of the M include eigenvalues and the column i of the vector V is the eigenvector related to the eigenvalue $M_{i,i}$, where $V = P_1.P_2.P_3\dots$, and P_i is the i^{th} Jacobi rotation matrix.

An alternative solution: An alternative approach could be employed as follows. Consider Eq. 2 and assume that $\theta_9 \neq 0$. The goal is to compute parameters vector Θ of the homography. We can assume that in the RANSAC applications in computer vision, the coordinates of d' (x' and y') are always non-zero (Here,

zero values of x' and y' means the pixel at left and top border of the image. However, we always consider the internal points of the image not the points in the borders). Scaling Eq. 2 by $\theta_9 \neq 0$, we obtain the following equation:

$$\begin{aligned} \left(\frac{\theta_1}{\theta_9} \frac{x}{x'} + \frac{\theta_4}{\theta_9} \frac{y}{y'} + \frac{\theta_7}{\theta_9} \frac{1}{x'}\right) - \left(\frac{\theta_3}{\theta_9} x + \frac{\theta_6}{\theta_9} y\right) &= 1 \\ \left(\frac{\theta_2}{\theta_9} \frac{x}{y'} + \frac{\theta_5}{\theta_9} \frac{x}{x'} + \frac{\theta_8}{\theta_9} \frac{1}{x'}\right) - \left(\frac{\theta_3}{\theta_9} x + \frac{\theta_6}{\theta_9} y\right) &= 1 \end{aligned} \tag{6}$$

By stacking up 4 of such equations, we would then have a system of 8 equations for 8 unknown variables. We can directly solve this system of linear equations. The transformation matrix would be then given as:

$$T_1 = \begin{bmatrix} \frac{\theta_1}{\theta_9} & \frac{\theta_4}{\theta_9} & \frac{\theta_7}{\theta_9} \\ \frac{\theta_2}{\theta_9} & \frac{\theta_5}{\theta_9} & \frac{\theta_8}{\theta_9} \\ \frac{\theta_3}{\theta_9} & \frac{\theta_6}{\theta_9} & 1 \end{bmatrix} = \begin{bmatrix} \theta_1^1 & \theta_4^1 & \theta_7^1 \\ \theta_2^1 & \theta_5^1 & \theta_8^1 \\ \theta_3^1 & \theta_6^1 & \theta_9^1 \end{bmatrix} \tag{7}$$

If the assumption of $\theta_9 \neq 0$ is true, we can then consider T_1 as Θ . Although this approach is simple and computationally much more efficient than the previous one based on SVD decomposition, its practical use has not recommended, due to the possible unstable numerical computation [15]. In fact, the problem is that theoretically θ_9 might be zero or very close to zero. In this case, the true solution cannot be reached, and thus, this method can be expected to lead to unstable results [15].

Our key enabling observation is that this method can be improved as follows. In fact, instead of computing one we can compute three transformations. Indeed, we can use the same approach assuming that $\theta_6 \neq 0$ or $\theta_3 \neq 0$ to compute two other models. In this case, we have three models, denoted as T_1 , T_2 , and T_3 . As θ_3 , θ_6 and θ_9 cannot be zero or near to zero at the same time (see Eq. 2), at least one of these models is a correct model. In fact, the correct Θ can be obtained by considering the last rows of T_1 , T_2 and T_3 as follows:

- If there is no value near to zero, that is, θ_3 , θ_6 and θ_9 are all far from zero, then in this case, T_1 , T_2 and T_3 would differ only by a constant factor multiplication. Therefore, any of them can be chosen as the correct transformation.
- If there is one value near to zero at the same index of last row of two transformations (for example θ_3^1 in T_1 and θ_3^2 in T_2), there would then be a zero at that index in Θ (for example $\theta_3 = 0$). In this case, one of the matrices containing these zeros (for example T_1 and T_2) could be selected as the correct transformation. This indicates that our assumption of $\theta_3 \neq 0$ has been wrong, resulting in the other matrix T_3 to have unstable values.
- If there are two values near to zero in only one of the computed matrices (for example θ_3^1 and θ_6^1), there would be two zeros at those indexes in Θ (for example $\theta_3 = \theta_6 = 0$). In this case, the matrix containing these zeros, T_1 , should be selected as the correct transformation. This indicates that two of our assumptions of $\theta_3 \neq 0$ and $\theta_6 \neq 0$ have been wrong, resulting in the other matrices T_2 and T_3 to have unstable values.

3 Cell Broadband Engine

Cell Broadband Engine Architecture (CBEA) is a processor architecture made by the partnership Sony, Toshiba and IBM (STI), originally designed for PlayStation3. But its capabilities have made it suitable for image and signal processing, scientific computation and visualization. In this section, we briefly review the Cell architecture with emphasis on some of its salient features that have been exploited in our implementation (see [16] for more detailed discussion).

The cell processor, the first implementation of CBEA, is a heterogeneous multi-core architecture consists of one 64-bit PowerPC processing element (PPE) and eight synergetic processing elements (SPEs). The PPE runs the operating system and coordinates all the resources including SPEs. Each SPE is a RISC-like in-order processor with 256KB instruction and data memory called local storage (LS) and 128 128-bit registers. The large number of registers facilitates highly efficient software level instruction scheduling which enables compilers and software developers to utilize optimization techniques like loop unrolling. There are two instruction pipelines in each SPU called even (including fixed and floating point units) and odd (including other functional units) pipelines. In each clock cycle, at most (up to) two instructions can be issued, one per pipeline. Each SPE consists of a synergetic processor unit (SPU) and a memory flow controller (MFC) which includes a DMA controller, a memory management unit, a bus interface unit, and an atomic unit for synchronization with other SPUs and the PPE. While the main memory of the board can only be accessed directly by the PPE, the MFC of each SPE manages various communications including mailboxes (a short queue of 32-bit messages), signal notifications, memory mapped IO, atomic operations and the most important DMA access. The overall architecture of Cell processor is illustrated in Fig. 1.

SPEs are SIMD vector processors and they have SIMD instruction set which provides a rich set of operations (such as logical, arithmetical, casting, load and store, and so on) that can be performed on 128-bit vectors of either fixed-point or floating-point values. The vectors can contain various sizes of variables, such as 8, 16, 32 or 64 bits. Since SPEs are pure SIMD processors and are not optimized to run scalar code and handling unaligned data, high application performance is strongly tied to heavy use of vectorized operations. When using the fused multiply and add single precision floating point operations, each SPU can achieve the peak performance of 25.6 GFLOPS at 3.2GHz (i.e. 204.8 GFLOPS for eight SPEs in the board). The communication between processing elements, main system memory and external I/O is provided by the high bandwidth element interconnect bus (EIB).

While single precision floating point instructions can fully pipelined, double precision are performed in four way SIMD fashion and are only partially pipelined and stall dual issues of other instructions. So, programs with heavily use of double precision floating point operations have not good performance on SPEs (the peak performance of double precision is 14.6 GFLOPS instead of 204.8 GFLOPS). Another important architectural constraint of SPU is the lack of branch prediction hardware on the SPEs. Hence, SPE applications should

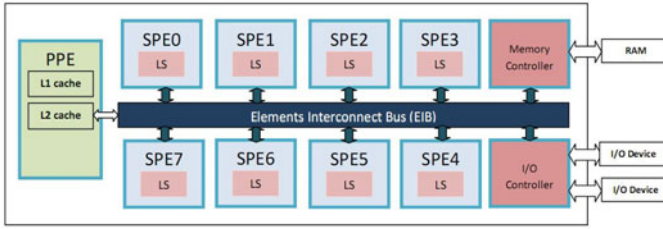


Fig. 1. The overall architecture of the Cell Processor

avoid using conditional instructions as much as possible to keep the pipeline utilization high.

4 Implementation of RANSAC on CBE

In this section, we describe the implementation of RANSAC algorithm on the cell processor and four levels of parallelism exploited in the implementation. In the implementation, different kinds of parallelism are exploited. So, at the first step, we will describe the optimization of RANSAC algorithm on one SPE. After that, the parallel implementation on multiple SPEs will be given.

4.1 Optimization for One SPE

In this section, we present and discuss our optimization techniques for efficient implementation of the RANSAC algorithm on one SPE.

In the implemented RANSAC algorithm on Cell processor, PPE selects the required set MSS from data for each SPE and puts the both data and MSS in the global memory (RAM). PPE then sends a message to each SPE to start its operation. When a given SPE receives the message, it starts to bring the required data from global memory to its local memory and it then starts the model generation to obtain models required in verification phase. When the process ends, SPE put the results in global memory and sends a message to inform PPE.

As stated before, SPEs are pure vector processor and vectorization has the biggest impact in terms of relative gains that can be achieved in the computation. Thus, the first level of parallelization is vectorizing the scalar code. To exploit vector computation, the developer must explicitly program the parallel execution in the code by exploiting SIMD instructions, called intrinsics, which are inline assembly-language instructions in the form of function calls available in high level languages (i.e. C and FORTRAN). In this approach, instead of scalar computation, we exploit vector instructions. For example, instead of performing computation on one point, determined by scalar values (x_i, y_i) , the operations are performed on (X_i, Y_i) including four points, i.e. $X_i = (x_{4*i}, x_{4*i+1}, x_{4*i+2}, x_{4*i+3})$ and $Y_i = (y_{4*i}, y_{4*i+1}, y_{4*i+2}, y_{4*i+3})$.

Software instruction reordering, which in assembly language should explicitly be done by programmer and in high level languages can be done with the aid of compiler, is another important optimization technique. The instructions can be reordered to exploit both odd and even pipeline of the SPE. For example, execution load and store instruction (which should be done in even pipeline) and fixed and floating point instruction (that is related to odd pipeline) can be executed in parallel. This parallelization is the second level of parallelization. The large size of register file can help instruction reordering technique to be useful in avoiding pipeline stall with aggressive loop unrolling. If the number of loop iterations is a small constant, we consider using a straight code instead of a loop. The loop unrolling technique is also beneficial due to decreasing the cost of branch mis-prediction.

To reduce data-access latencies (bringing data from main memory to the local storage), we have utilized double buffering technique. As the third level of parallelization, it helps us to perform communication (data movement using non-blocking DMA request) with computation in parallel, that is, to achieve overlapping of communication with computation.

4.2 Parallel Implementation

Consider the parallel execution of RANSAC on S SPEs (as the forth level of parallelization). Let M denotes the set MSS in which each element is randomly selected by the PPE from the dataset. M could be divided to S distinct subsets M_i which satisfies:

$$M = \bigcup_{i=1}^{i \leq S} M_i, \forall i, j (1 \leq i, j \leq S) : |M_i| = \frac{|M|}{S}, M_i \cap M_j = \emptyset \quad (8)$$

The PPE sends a message to all SPEs to start the computation. Upon receiving this message, each SPE_i starts the computation with reading the M_i via a DMA request and afterward generates the model set M_i . When the model generation phase is finished, the model verification can be started in which the SPE verifies its set of models using all data in the dataset. The pseudo code is given in Algorithm 1. Like in one SPE case, after finishing the process, each SPE puts the results in global memory and sends a message to inform the PPE. The overall computation finishes when the computation ends by all SPEs.

Cell processor does not support data broadcasting. Instead, each SPE has to bring all data from the main memory individually. Fortunately, as mentioned before, Cell has a 4-ring high bandwidth interconnect bus and also the amount of required data to be transferred from the main memory to local memory of each SPE is small in comparison with the amount of computations.

4.3 Experimental Results

We have applied RANSAC for homography model estimation. The experiments run on a Mercury Cell acceleration board 2 (CAB2) [17], a PCI Express accelerator card based on the Cell Broadband Engine processor with an internal

processor clock of 2.8 GHz. So, the total theoretical peak performance of eight SPEs in the board is 179.2 GFLOPS. The performance of implemented RANSAC in terms of run time and sustained floating point operations per second (FLOPS) are shown in Table 1 which illustrates the performance metrics in hypothetical model generation step, model verification step, and total for one stage for the given number of models and data elements.

Algorithm 1. Parallel RANSAC Algorithm on Multiple SPEs

Input: S : Number of SPEs; D : Data set; δ : fitting threshold
Output: θ^* : model parameters which best fit data
 Select M (the set MSS s) from D and then, put D and M in the global memory
 Send a message to all SPEs to start their computation
In each SPE_i , $1 \leq i \leq S$, after receiving the start message:
 $CS_i^* = \emptyset$
 $itr_idx = 0$
while $itr_index < T_itr$
 $CS_i = \emptyset$
 $S = \text{Read } \bar{s} \text{ values from } M_i$
 $\theta = \text{Estimate model parameters based on } S$
foreach point d_j in the dataset ($d_j \in D$)
if point d_j fits the model ($err(\theta, d_j) \leq \delta$)
 $CS_i = CS_i \cup d_j$
if CS_i is better than CS_i^*
 $CS_i^* = CS_i$
 $\theta_i^* = \theta$
 $N_i = |CS_i|$
end if
 $itr_idx = itr_idx + 1$
 $p = N_i/N$
 $T_{itr} = \lceil \log \epsilon / (\log(1 - p^{\bar{s}})) \rceil$
end while
 Put (θ_i^*, N_i) in global memory as the best local model
 Wait for all SPEs to be finished
 Select $(\theta^*, CS^*) = (\theta_k^*, N_k)$ where $N_k = \max_{1 \leq i \leq S} (N_i)$
return θ^*

The performance of implemented algorithm in terms of run time and sustained floating point operations per second (FLOPS) for estimating homography are shown in Table 1 which illustrates the performance metrics in hypothetical model generation step, model verification step, and total for one stage for 256 models and 2,048 data elements in addition to communication overhead for both one SPE and eight SPEs.

For model generation phase, we have examined both approaches (using Jacobi eigenvalue computation and the alternative approach). The two important advantages of the alternative approach is that finding the solution needs less times, compared with computing SVD or Eigenvalue computation, and it can

Table 1. Performance evaluation of homoraphy estimation

#	Impelemtation	Sustained GFLOPS			I/O Overhead	Time(us)
		Model Generation	Model Verification	Overall		
1	on 1 SPE	3.73	12.34	6.11	<1%	5971.87
	on 8 SPEs	29.90	95.73	48.52	2%	766.68
2	on 1 SPE	1.39	12.34	10.89	<1%	1901.95
	on 8 SPEs	11.13	95.73	80.09	5%	258.59

Table 2. Number of generated and verified model/second (data set size:2048)

	Homography (1)	Homography (2)
modesl/s (1 SPE)	42,800	134,000
modesl/s (8 SPE)	333,000	989,000

also be correctly done by using single-precision computation, whereas in the first approach the precision of floating point operations in CBE is not completely precise for all cases. In the first approach by using Eigenvalue computation, model generation phase consumes nearly 80% of total time and as the performance of model generation is low (compared with the performance of model verification), the overall performance is near to the performance of model generation (rather than the model verification). But if we use the alternative approach, although the sustained GFLOPS is less than the previous ones (due to the large amount of memory movements needed for Gaussian elimination algorithm with pivoting), it takes less time and the overall algorithm will be nearly three times faster than by using the first approach.

As mentioned, the EIB is a high speed and high bandwidth bus and in fact, the main reason for increasing communication overhead over multiple SPEs is that the computation time is less than for one SPE case. Thus, overlapping of computation and data communication technique is less effective since there is not enough computation to hide the communication. Another considered performance evaluation measurement is the number of generated and verified models which is presented in Table 2.

5 Conclusion and Future Works

In this paper, we presented an efficient strategy for implementation of the RANSAC algorithm on Cell processor. We presented the results of our practical implementations which clearly demonstrate that a very high sustained computational performance (in terms of sustained GFLOPS) can be achieved, resulting in a fast generation and evaluation of a large number of models. For example, we have achieved the performance of 80 sustained GFLOPS for estimating homography, and the capability to evaluate 989 models per millisecond. Our results clearly demonstrate the advantages of parallel implementation of RANSAC on MIMD-SIMD architectures. They also prove that, by using such a parallel implementation over the sequential one, a problem with a fixed number of iterations

(hypothetical models) can be solved much faster and/or for real-time applications, with a fixed given computation time, much more models can be generated and evaluated, leading to a potentially better accuracy of the model. We are currently developing several image processing applications which require a fast and real-time implementation of RANSAC.

References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Commun.* 24(6), 381–395 (1981)
2. Chum, O., Matas, J.: Randomized RANSAC with $T_{d,d}$ test. In: *Proc. British Machine Vision Conference*, pp. 448–457 (2002)
3. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: *Proc. Int. Conf. on Computer Vision*, pp. 754–760 (1998)
4. Torr, P.H.S.: Outlier detection and motion segmentation. Ph.D. dissertation, Dept. of Engineering Science, University of Oxford (1995)
5. McLauchlan, P., Jaenicke, A.: Image mosaicing using sequential bundle adjustment. In: *Proc. British Machine Vision Conference*, pp. 751–759 (2000)
6. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II. LNCS*, vol. 5303, pp. 500–513. Springer, Heidelberg (2008)
7. Chum, O., Matas, J.: Matching with PROSAC - progressive sample consensus. In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 220–226 (2005)
8. Tordoff, B.J., Murray, D.W.: Guided-MLESAC: Faster image transform estimation by using matching priors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(10), 1523–1535 (2005)
9. Myatt, D.R., Torr, P.H.S., Nasuto, S.J., Bishop, J.M., Craddock, R.: NAPSAC: High noise, high dimensional robust estimation. In: *Proc. British Machine Vision Conference*, pp. 458–467 (2002)
10. Nistèr, D.: Preemptive RANSAC for live structure and motion estimation. *Mach. Vision Appl.* 16(5), 321–329 (2005)
11. Iser, R., Kubus, D., Wahl, F.M.: An efficient parallel approach to random sample matching (pRANSAM). In: *Proc. Int. Conf. of Robotics and Automation*, pp. 1199–1206 (2009)
12. Winkelbach, S., Molkenstruck, S., Wahl, F.M.: Low-cost laser range scanner and fast surface registration approach. In: *28th Annual Symp. of the German Association for Pattern Recognition*, pp. 718–728 (2006)
13. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78(1), 138–156 (2000)
14. Press, W.H., et al.: *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. Cambridge University Press, Cambridge (2007)
15. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
16. Arevalo, A., et al.: *Programming the Cell Broadband Engine Architecture: Examples and Best Practices*. IBM Redbook (2008)
17. Mercury CAB2, http://www.mc.com/products/boards/accelerator_board2.aspx

Organizing and Browsing Image Search Results Based on Conceptual and Visual Similarities

Grant Strong, Enamul Hoque, Minglun Gong, and Orland Hoerber

Dept. of Computer Sci., Memorial Univ. of Newfoundland, St. John's, NL, Canada

Abstract. This paper presents a novel approach for searching images online using textual queries and presenting the resulting images based on both conceptual and visual similarities. Given a user-specified query, the algorithm first finds the related concepts through conceptual query expansion. Each concept, together with the original query, is then used to search for images using existing image search engines. All the images found under different concepts are presented on a 2D virtual canvas using a self-organizing map. Both conceptual and visual similarities among the images are used to determine the image locations so that images from the same or related concepts are grouped together and visually similar images are placed close to each other. When the user browses the search results, a subset of representative images is selected to compose an image collage. Once having identified images of interest within the collage, the user can find more images that are conceptually or visually similar through pan and zoom operations. Experiments on different image query examples demonstrate the effectiveness of the presented approach.

1 Introduction

The primary method of image retrieval used on the Web is based on keyword search [12]. Search engines have merely adapted their document retrieval algorithms to the metadata (keywords, tags, and/or associated descriptions) of images and present the results in a scrollable list that is ranked based on relevance to the query. While list interfaces are easy to use, there is limited ability to manipulate and explore search results. In addition, keyword search relies on the assumptions that the contents of images are accurately described by the metadata, and that the searcher is able to provide a concise description of what they are seeking; these assumptions are not always valid.

On the other front, content-based image retrieval (CBIR) techniques conduct search using visual features [4, 17]. However, they often lead to a *semantic gap*: the gap between the way a person finds similarities between images at the conceptual level and the way the system generates similarity based on pixel statistics [4]. Furthermore, CBIR techniques often require users to draw sketches as visual queries or to rank suggested images, making them somewhat cumbersome to use.

A nice middle ground seems to be searching images using keywords and then organizing the search results using similarity-based image browsing (SBIB) techniques [8]. This allows searchers to use easy-to-construct textual queries, as well as facilitates their locating of desired images through the structured layout of retrieval results. Google Swirl is such an approach, which uses a visual similarity graph to present the images found through a given textual query [7].

Previous studies have shown that, many image search queries are associated with conceptual domains that include nouns, people's names, and locations [2, 10]. It is advantageous for the search results of such queries to be diverse in nature, both from different conceptual perspectives as well as the visual features. A diversified set of search result images provides a broad scope from which the searcher can seek the images that match their needs. In situations such as this, it may be more beneficial to consider not only visual similarity but also conceptual relatedness between images in organizing the results, and then allow the searcher to focus on a specific area to explore conceptually related images.

Motivated by the above hypothesis, here we propose a novel approach for organizing and browsing textual search results using a combination of conceptual and visual features. Given a user-specified query, our system first performs conceptual query expansion to ensure that a set of conceptually diverse images are retrieved. This is done by automatically extracting a list of concepts from Wikipedia that are relevant to the query, and then perform textual retrieval using both the original query and the related concepts. The relations between the concepts used for query expansion are used to derive a conceptual feature vector for each image, which is used in conjunction with the visual feature vector extracted from the image to form a hybrid feature vector. A self-organizing map based approach is then used to map images onto a 2D canvas, so that the ones with similar concepts and/or visual features are placed close to each other. The searcher can visually explore the search results on the 2D canvas, which initially contains representative images only. Zooming into an area of interests will unveil more conceptually and visually similar images; panning allows the searcher to move within the image space.

1.1 Related Work in Conceptual Query Expansion

A promising direction for improving the quality of search results in general is the introduction of query expansion based on the most related concepts to the query [5]. Such an approach is particularly useful for diversifying the search results covering different concepts and enabling searchers to assist with the query refinement process. However there are a number of challenges associated with conceptual query expansion. The first problem is finding a suitable knowledge base that has sufficient coverage of a realistic conceptual domain. While WordNet has been used to improve image retrieval [11], it does not contain information pertaining to the proper nouns that are common in image search queries. As such, using Wikipedia for reformulating queries has shown promise [14], and is the approach we use in our work.

The second challenge is in ranking the extracted concepts for the purposes of selecting the most relevant of these. A useful approach to this problem is to measure the semantic relatedness between the original query and each of the concepts derived from that query. A number of different methods have been devised to use Wikipedia for this purpose, including WikiRelate! [21], Explicit Semantic Analysis (ESA) [6], and Wikipedia Link-based Measure (WLM) [13]. We use WLM in our work because of its computationally efficiency and accuracy.

1.2 Related Work in Similarity-Based Image Browsing

Unlike CBIR, which aims to provide users with the desired images based on a set of input images, SBIB studies how to organize images, either from personal collections or online search results, based on their visual similarities. The challenge of SBIB is to arrange images in such a way as to support the browsing and exploration experience, as well as to facilitate users in locating the image(s) they are looking for.

Several SBIB approaches have been proposed, all of which use similar color, structure, and texture features as the basis of similarity measures for their organizations, but differ in how those similarity measures are used to relate images [8]. Heesch and Ruger et al. model relations between images using nearest neighbor networks adapted from document browsing techniques [9]. Search, in their case, is the process of following a path through the series of connections by clicking relevant images. Nguyen and Worring propose a system to meet the three requirements of overview, structure preservation (as it relates to image similarity), and visibility in [15]. They organize using non-linear probabilistic methods and k-means clustering to determine overview images while browsing. In [16], Pecenovic et al. use Sammon’s projection to map the images onto 2D space for visualization and use a heuristic balanced k-means algorithm for determining representative centroids. Strong and Gong’s approach also maps images onto 2D space, but using a multi-resolution self-organizing map algorithm, which can evenly spread images across the available screen space and provide priority information for interactive browsing [18]. User studies have shown that this browsing interface helps to reduce the time users needed to locate the desired images [20].

Strong and Gong’s approach is chosen as the basis for organizing image search results in our work. However, a key difference between our approach and all existing SBIB techniques is that we not only organize images based on visual information, but also extract and utilize concept information. Our experiments show that incorporating conceptual information can make the organization results much more meaningful than using visual information alone.

2 The Proposed Approach

2.1 Concept Extraction Using Wikipedia

Images retrieved using a user specified query does not carry conceptual information. To address this problem, as well as to obtain a set of conceptually diverse images, we first apply conceptual query expansion to discover different concepts related to the input query. The concepts are then used to diversify the image search results, as well as for generating conceptual feature vectors for the images found.

In this work, we use Wikipedia as the core knowledge base for the query expansion process. Wikipedia is an excellent source of information for the purposes of image search since it includes a large number of articles describing people, places, landmarks, animals, and plants. It is densely structured, with hundreds of millions of links between articles within the knowledge base. Most of the articles also contain various representative images and associated textual captions.

A dump of the Wikipedia collection was obtained in June 2010, and was preprocessed to support the type of knowledge extraction required for our purposes.

Matching a user-supplied query Q to this knowledge base is simply a matter of selecting the best matching article (referred as the home article) using Wikipedia's search feature. In the case where query Q is ambiguous and Wikipedia suggests multiple links, the ones with higher commonness values are used as home articles. Here the commonness value of an article is calculated based on how often it is linked by other articles.

In analyzing Wikipedia, we observed that the in-link articles (ones having links to a home article) and out-link articles (ones to which a home article links) often provide meaningful information that is closely related to one of the home articles, and hence the user-specified query. Therefore, these linked articles are located and their titles are extracted as candidates for related concepts.

For some queries, the total number of linked articles found might be very high and some of them may not be well-related to the query. Thus, a filtering step is necessary to ensure the quality of the concepts that are extracted. The filtering is performed based on the semantic distance between each linked article and its corresponding home article. WLM [13] is used for this purpose, which applies Normalized Google Distance (NGD) on the domain of Wikipedia articles. The NGD between any two articles a and b is calculated using the hyperlink structure of the associated articles to determine how much they share in common. That is:

$$NGD(a, b) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))} \quad (1)$$

where A and B are the sets of all articles that link to the article of a and b , respectively, W is the set of all articles on Wikipedia, and operator $|\cdot|$ computes the number of articles in the set. The distance obtained is between 0 and 1, with a smaller value indicating a higher degree of relatedness.

Once the semantic distance measures for all linked articles are calculated, they are sorted in increasing order. The titles of the top N articles are then selected as related concepts.

2.2 Image Search Using Conceptual Query Expansion

Given a user-specified query Q and the corresponding related concepts, $\{R_k | 1 \leq k \leq N\}$, we generate a set of N sub-queries by combining the query with each related concept. Each sub-query $\langle Q, R_k \rangle$ is then used to retrieve a set of images from the Web using the Ajax Goggle API. To avoid duplicate images returned for different sub-queries, a union operation is performed when combining the result sets. All images retrieved are tagged with the corresponding concept R_k . The total number of images retrieved is limited to T , a user tunable parameter.

For example, if a user enters the query "Washington", the system will use the query to perform a search in Wikipedia, which will return multiple articles about "Washington". Articles having higher commonness scores, such as "Washington (state)", "Washington, D.C.", "University of Washington", and "George Washington" are then selected as home articles. All articles link to or linked by one of the home articles are considered as candidates for related concepts. The top N candidates with smallest semantic distances are used for generating sub-queries (e.g., "Washington

Monument” and “White House” for the home article “Washington D.C.”; “Martha Washington” and “Benjamin Franklin” for the home article “George Washington”).

2.3 Hybrid Feature Vector Generation

The set of images retrieved through conceptual query expansion are highly diverse, both conceptually and visually. While this helps to ensure that the desired images are returned, finding them by going through the whole set can be time consuming. To facilitate searchers in locating the images they seek, a SOM-based image organization technique is applied to the image retrieval results.

SOM is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional representation of the input space of the training samples. To use it for image organization, we first need to represent all images using feature vectors, the distances among which indicate the similarities between the corresponding images. Different ways for generating feature vectors from visual information and their performances have been studied previously [19]. While these types of feature vectors can be used to organize images based on color and/or shape similarities, they cannot group conceptually related, but visually different, images together. To address this problem, here we propose a hybrid feature vector.

The hybrid feature vector for an image contains two portions: a conceptual portion determined using the concept tag carried by the image; and a visual portion extracted from pixel intensities and distributions. For the visual portion, we choose the color-gradient correlation approach since it considers both color and general shape information, is efficient to calculate, and offers good organizational performance [19].

To compute the color-gradient correlation of an image I , we first compute the gradient magnitude l_p and gradient orientation θ_p for each pixel p . We then divide the color and gradient orientation spaces into N_c and N_θ bins, respectively. With functions $\mathcal{C}(p)$ and $\theta(p)$ providing the color and gradient orientation bin indices for pixel p , the sum of gradient magnitudes for all pixels belonging to the k^{th} color and gradient orientation bin can be computed using:

$$m_k = \sum_{\mathcal{C}(p) \times N_\theta + \theta(p) = k} l_p \quad (2)$$

The visual feature vector $\mathbf{V}(I)$ is then formed using the normalized values of all bins to make the vectors generated from images of different sizes comparable.

Extracting conceptual feature vectors, on the other hand, is not as straightforward. To simplify the problem, we assume that different images retrieved using the same sub-query (i.e., the same related concept) are conceptually the same and, hence have the same conceptual feature vector. Consequently, what we need to do is to derive a feature vector \mathbf{C}_k for each concept R_k used for retrieving images. Even though it is difficult to convert concepts into vectors directly, we can first compute an $N \times N$ semantic distance matrix for the N related concepts using Equation (1):

$$\mathbf{D} = \begin{bmatrix} 0 & NGD(R_1, R_2) & \cdots & NGD(R_1, R_N) \\ NGD(R_2, R_1) & 0 & \cdots & NGD(R_2, R_N) \\ \vdots & \vdots & \ddots & \vdots \\ NGD(R_N, R_1) & NGD(R_N, R_2) & \cdots & 0 \end{bmatrix} \quad (3)$$

where, by definition, we have $NGD(R_k, R_k) = 0$ and $NGD(R_j, R_k) = NGD(R_k, R_j)$. Hence, the above matrix is symmetric.

The above matrix encodes the relatedness information among different concepts, which is then used to generate a set of m -dimensional vectors $\{\mathbf{C}_1, \dots, \mathbf{C}_N\}$. We need the vectors to model the relatedness information as closely as possible (i.e., the distance between any two vectors is approximate to, if not the same as, the semantic distance between the corresponding concepts). This is the same as minimizing the following least-squares function:

$$\{\mathbf{C}_1, \dots, \mathbf{C}_N\} = \underset{\mathbf{C}_1, \dots, \mathbf{C}_N}{\operatorname{argmin}} \sum_{1 \leq j, k \leq N} (\|\mathbf{C}_j - \mathbf{C}_k\| - \mathbf{D}_{j,k})^2 \quad (4)$$

where $\|\mathbf{C}_j - \mathbf{C}_k\|$ is the Euclidean distance between the two vectors.

As a result, our task of finding a set of vectors \mathbf{C} based on a given distance matrix \mathbf{D} becomes the classical multi-dimensional scaling problem [3], which can be solved by existing techniques [1].

In the end, the hybrid feature vector $\mathbf{H}(I)$ for a given image I is formed as $\langle \mathbf{C}_{R(I)}, \mathbf{V}(I) \rangle$, where $R(I)$ is the concept used to retrieve image I . Since the conceptual portion has m dimensions and the visual part has $N_c \times N_\theta$ dimensions, the total dimensions of a hybrid feature vector is $m + N_c \times N_\theta$. In this paper, we set $m = 4$ and $N_c = N_\theta = 8$, resulting 68 dimensional hybrid feature vectors.

2.4 Image Organization Using Self-Organizing Map

With hybrid feature vectors generated for all images, the next step is to map the vectors onto a 2D virtual canvas. This is achieved through training a SOM, a process similar to the one discussed in [18, 19], with changes applied for handling hybrid vectors. The SOM is capable of organizing non-linear vectors in a topologically meaningful way. Unlike many other dimension reduction and vector embedding techniques, such as Sammon's projection [16], it has an inherent balancing property that seeks to spread the sample vectors over the whole map regardless of the span of the input vectors. Thus, when visualizing the organization of the image search results, images are evenly distributed in the display, making full use of the available screen space.

An SOM consists of 2D network of $M \times M$ interconnected units, where each unit x has, initially, a randomly generated weight vector $W(x)$ associated with it. During each iteration of the training process, all feature vectors affect an area of the map in a random order. The area is chosen by finding the unit with the weight vector that most closely matches the feature vector in terms of minimum distance. Then the best match unit and the neighboring units' weight vectors are interpolated toward the feature vector, where the amount of interpolation varies based on a Gaussian decay. The overall learning effect dwindles exponentially over time.

Since hybrid feature vectors are used to encode images in our approach, the SOM can be trained based on either visual or conceptual information, resulting images being grouped by either visual or conceptual similarities. More generally, we can organize images by both visual and conceptual information through using a weighted average of both visual and conceptual distances. That is, given two images I and J , whose feature vectors are $\mathbf{H}(I)$ and $\mathbf{H}(J)$, their distance is calculated as:

$$\text{Dist}(\mathbf{H}(I), \mathbf{H}(J)) = \alpha \|\mathbf{C}_{R(I)} - \mathbf{C}_{R(J)}\| + (1 - \alpha) \|\mathbf{V}(I) - \mathbf{V}(J)\| \quad (5)$$

where the parameter α controls the relative importance of the conceptual distance and visual distance.

When the training is complete, the coordinates of the best match unit for each feature vector give us the best position for that feature vector's image in 2D. The SOM's topology preserving property ensures that visually and conceptually similar images are mapped to locations that are closer to each other, and vice versa.

2.5 Image Browsing Interface

Taking the images and their mapped locations as input, the browsing interface dynamically generates an image collage based on which portion of the 2D virtual canvas is currently in view. The users can adjust the viewing area through intuitive panning and zooming operations. Once the viewing area is set, all of the images inside the area are candidates for generating the image collage, but only a number of representative images are actually used. An image is chosen to represent nearby images if its feature vector is close to the average of feature vectors in the neighborhood.

The number of images actually used depends on the screen size of the browsing window, as well as the user specified image display size. The zooming operation does not affect the display size of each individual image, but reduces the portion of the canvas that is viewable. As a result, when users zoom in, they can observe more images in the region rather than see the same set of images at higher resolution.

Under a typical browsing scenario, the user initially sees only a couple representative images. These may come from specific expanded queries that produce groups of images, or from other visually similar images that are obtained from multiple expanded queries. Once having identified images of interest, the searcher can find more images with similar concepts or visual features by zooming into the area.

3 Experiment Results

Thus far in this paper, we describe how to infuse concept information into the image search process and how to utilize the concept information in organizing the resulting images. Next we examine the merit of doing this for the purposes of browsing and exploring image search results.

The proposed algorithm is tested using a variety of datasets, all are retrieved using Google image search with the above described conceptual query expansion procedure. The results for two queries are shown here to demonstrate the performance of the algorithm under different levels of query ambiguity. The first one, "Washington", is highly ambiguous and the related images range from landmarks (Washington Monument and Space Needle), to persons (George Washington and Denzel Washington), and maps (Washington, D.C. and Washington state). It is also one of the examples used by Google Swirl, the result of which can be found at [7]. The second query simply appends "D.C." to the original query, which increases the specificity of the results significantly. Yet, the query expansion process still finds multiple related concepts, such as "Washington Monument", "White House", "United States Capitol", etc.



Fig. 1. Comparison between image organization results generated using the visual feature vector alone (top) and the hybrid feature vector (middle), as well as the zoomed-in views of the areas marked by the red squares (bottom). For illustration purposes, images of several persons related to George Washington are highlighted in the left dataset, and the ones related to buildings in Washington, D.C. are highlighted in the right dataset. One can readily see that the grouping of the highlighted images is much better with the hybrid feature vector (middle) than the visual feature vector (top).

The organization of the results (shown in Figure 1) suggest that, since conceptually related images may be visually different, organizing images using visual features only does not place them together. Using concept information, on the other hand, not only can images retrieved using the same concept be grouped together, but also, images from related concepts can be placed at nearby locations. Hence, once users identify the images of interest, they can easily zoom into the area to find more conceptually related images.

4 Conclusions

In this paper, we present a novel approach of organizing and browsing image search results based on visual and conceptual similarities. The main contributions of this work are the infusing of related concepts into image search results through query expansion, the encoding of both visual and conceptual information in feature vector generation, the adopting of hybrid feature vectors in the training of the SOM, and finally the organizing of the image search results based on both conceptual and visual features. The benefit of this organization is that it groups images from the same or related concepts together, while simultaneously grouping visually similar objects, allowing users to explore their own areas of interests based on both conceptual and visual features of the images. Moreover, our experimental result shows that, this visualization technique can be particularly helpful for dealing with an ambiguous query, by separating images of ambiguous concepts from each other into their own area and placing visually and conceptually similar concepts near one another.

The future work includes plans to allow the user to provide further input on the process of query refinement and image search results organization. We are also interested in enhancing the quality of feature vectors through incorporating other types of visual features. Finally, we want evaluate the benefit of this approach through user evaluations.

References

1. Algorithmics Group: MDSJ: Java Library for Multidimensional Scaling, Version 0.2 (2009), <http://www.inf.uni-konstanz.de/algo/software/mdsj/>
2. André, P., Cutrell, E., Tan, D.S., Smith, G.: Designing novel image search interfaces by understanding unique characteristics and usage. In: Proc. IFIP Conference on Human-Computer Interaction, pp. 340–353 (2009)
3. Borg, I., Groenen, P.: Modern Multidimensional Scaling: Theory and Applications, 2nd edn. Springer, Heidelberg (2005)
4. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys* 40(2), 1–60 (2008)
5. Fonseca, B.M., Golgher, P., Póssas, B., Ribeiro-Neto, B., Ziviani, N.: Concept-based interactive query expansion. In: Proc. ACM International Conference on Information and Knowledge Management, pp. 696–703 (2005)
6. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proc. International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)

7. Google: Google Image Swirl (2009), <http://image-swirl.googlelabs.com/>
8. Heesch, D.: A survey of browsing models for content based image retrieval. *Multimedia Tools and Applications* 40(2), 261–284 (2008)
9. Heesch, D., Rüger, S.: Image Browsing: A semantic analysis of NNk networks. In: *Proc. International Conference Image and Video Retrieval*, pp. 609–618 (2005)
10. Jansen, B.J., Spink, A., Pedersen, J.: An analysis of multimedia searching on AltaVista. In: *Proc. ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 186–192 (2003)
11. Joshi, D., Datta, R., Zhuang, Z., Weiss, W.P., Friedenber, M., Li, J., Wang, J.Z.: PARAGRAB: A comprehensive architecture for web image management and multimodal querying. In: *Proc. International Conference on Very Large Databases*, pp. 1163–1166 (2006)
12. Kherfi, M.L., Ziou, D., Bernardi, A.: Image Retrieval from the World Wide Web: Issues, Techniques, and Systems. *ACM Computer Survey* 36(1), 35–67 (2004)
13. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: *Proc. AAAI Workshop on Wikipedia and Artificial Intelligence*, pp. 25–30 (2008)
14. Myoupo, D., Popescu, A., Borgne, H.L., Moëllic, P.A.: Multimodal image retrieval over a large database. In: Peters, C., Caputo, B., Gonzalo, J., Jones, G.J.F., Kalpathy-Cramer, J., Müller, H., Tsirikla, T. (eds.) *CLEF 2009 Workshop, Part II. LNCS*, vol. 6242, pp. 1–8. Springer, Heidelberg (2010)
15. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. *J. Vis. Lang. Comput.* 19(2), 203–224 (2008)
16. Pečenović, Z., Do, M., Vetterli, M., Pu, P.: Integrated Browsing and Searching of Large Image Collections. In: Laurini, R. (ed.) *VISUAL 2000. LNCS*, vol. 1929, pp. 279–289. Springer, Heidelberg (2000)
17. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(12), 1349–1380 (2000)
18. Strong, G., Gong, M.: Browsing a large collection of community photos based on similarity on GPU. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) *ISVC 2008, Part II. LNCS*, vol. 5359, pp. 390–399. Springer, Heidelberg (2008)
19. Strong, G., Gong, M.: Organizing and Browsing Photos Using Different Feature Vectors and Their Evaluations. In: *Proc. International Conference on Image and Video Retrieval*, pp. 1–8 (2009)
20. Strong, G., Hoeber, O., Gong, M.: Visual image browsing and exploration (vibe): user evaluations of image search tasks. In: *Proc. International Conference on Active Media Technology*, pp. 424–435 (2010)
21. Strube, M., Ponzetto, S.P.: WikiRelate! computing semantic relatedness using wikipedia. In: *Proc. AAAI Conference on Artificial Intelligence*, pp. 1419–1424 (2006)

Evaluation of a Difference of Gaussians Based Image Difference Metric in Relation to Perceived Compression Artifacts

Gabriele Simone¹, Valentina Caracciolo^{1,2}, Marius Pedersen¹,
and Faouzi Alaya Cheikh¹

¹ Gjøvik University College, Gjøvik, Norway

² University of Roma Tre, Rome, Italy

Abstract. In this paper we investigate if the Difference of Gaussians model is able to predict observers perceived difference in relation to compression artifacts. A new image difference metric for specifically designed for compression artifacts is proposed. In order to evaluate this new metric a psychophysical experiment is carried out, where a dataset of 80 compressed JPEG and JPEG2000 images were generated from 10 different scenes. The results of the psychophysical experiment with 18 observers and the quality scores obtained from a large number of image difference metrics are presented.

Furthermore, a quantitative study based on a number of image difference metrics and five additional databases is performed in order to reveal the potential of the proposed metric. The analyses show that the proposed metric and most of the tested ones do not correlate well with the subjective test results, and thus the increased complexity of the recent metrics is not justified.

1 Introduction

We are witnessing a rapid growth in the number of media-rich documents that are created, transmitted, and consumed in our everyday activities. Furthermore, multimedia applications are also developing fast and becoming able to manage rather large images (e.g. image archiving, network image transmission, document imaging, digital photography, medical imaging, remote sensing, ...). Therefore, the demand for efficient and versatile image compression is more pressing than it has ever been before. Additionally, the improved media reproduction hardware has put a strain on the acceptable level of the quality of the processed media. During acquisition, communication, and consumption digital images are subjected to a wide variety of distortions, such as errors, noise and in particular compression artifacts. To ensure a certain quality level of multimedia applications/services one needs automatic means to evaluate the quality of digital images. Typically to identify the best reproduction among a number of variants of one reproduction algorithm (e.g. JPEG or JPEG2000 for compression), a psychophysical experiment has to be carried out. This results in a scale with

the perceived visual difference of the reproductions from the original image. Psychophysical experiments are very good in estimating the perceived distortion/quality of the media, they are, however, both time and resource demanding. Furthermore, they are not practical to use when automatic quality evaluation is needed, for example in computer vision based systems. This is why objective image difference metrics have been introduced. These are in general very simple mathematical formulae that give an objective measure of the quality of a reproduction, e.g PSNR, RMS, and MSE. They are however not sensitive to what the Human Visual System (HVS) is sensitive to.

Image difference metrics are based on a number of different ideas, even so, these metrics usually follow a general framework. In the most common framework the image and its reproduction are transformed into a suitable color space, preferably a perceptually uniform one. Then a simulation of the HVS is carried out, from simplistic methods as smoothing of the image by a local neighbor to more complex methods, such as using Contrast Sensitivity Function(CSF). Finally the difference between the two images is calculated, in general, using a color difference formula. This accounts for the outstanding number of image difference metrics that have been developed so far [1].

The rest of this paper will be organized as follows: Section 2 provides an insight into the state-of-the-art of image difference metrics. Section 3 describes the proposed metric, while Section 4 describes the psychophysical experiment. Section 5 presents the experimental results and discusses how the results from the image difference metrics reflects their perceptual quality estimation. Finally, in section 6 conclusions are drawn.

2 State of the Art of Image Difference Metrics

In 1976, CIE published the CIELAB color space as a uniform color space, in which the difference between two colors ΔE_{ab}^* is represented by their Euclidean distance. The CIELAB metric has been used as a tool for measuring perceptual differences between uniform patches of colors. Although non-appropriate, the CIELAB ΔE_{ab}^* has been used for measuring the color difference between images by computing the color difference of all the pixels and averaging. The unsatisfactory uniformity of CIELAB space induced researchers to produce other color-difference data and search for better color-difference formulae. The last CIE formula for small-medium color differences is the ΔE_{00}^* one, termed CIEDE2000 and based on a wider set of empirical data [2].

In 1997, Zhang and Wandell proposed a spatial extension to the CIELAB color-difference formula, termed $S - CIELAB$ [3], that provides both a spatial filtering to simulate the blurring of the HVS and a consistency with the *CIELAB* for large uniform areas [4]. In 2001 Johnson and Fairchild followed a similar approach [5], where the spatial filter is implemented in the frequency domain, obtaining a more precise control of the CSFs.

The Hue Angle Algorithm proposed by Hong and Luo in 2002 [6] is based on the known fact that systematic errors over the entire image is quite noticeable

and unacceptable. A histogram based of the hue angle is computed, and sorted in ascending order so that weights can be applied to four different quartiles of the histogram. The overall color difference is calculated by multiplying the weighted hue angle for every pixel. The Spatial Hue Angle Metric (*SHAME*), proposed by Pedersen and Hardeberg [1], can be considered as the combination of the original *S - CIELAB* and the hue angle algorithm, thus taking into account the spatial properties of the HVS. *SHAME - II* is a variation of *SHAME - I* that applies the filtering proposed by Johnson and Fairchild before applying the hue angle metric.

Very recently, in 2009, a Euclidean color-difference formula for small-medium color differences in log-compressed OSA-UCS space, termed ΔE_E , has been published by Oleari et al. [7]. This formula is statistically equivalent to CIEDE2000 in the prediction of many available empirical datasets, but with greater simplicity and clear relationship with visual processing [8]. In 2009 Simone et al. [9] proposed and tested a new metric, named *S- ΔE_E* , which works as the *S-CIELAB* from Johnson and Fairchild, but the ΔE_{ab} is substituted with the ΔE_E . Following, Ajagamelle [10] applied a novel technique, developing *S_{DOG} - CIELAB* which uses the Difference of Gaussians (DOG) model as a basis for the spatial filtering with the ΔE_{ab}^* as a color difference formula, and *S_{DOG} - DEE* which uses the DOG model and ΔE_E as color difference formula. The ΔE_E and the derivate image difference metrics have been extensively tested by Simone et al. [9] and Ajagamelle et al. [11].

Universal Image Quality (UIQ) proposed by Wang and Bovik [12] models image distortions as the combination of three elements: loss of correlation, luminance distortion and loss of contrast. Structural Similarity (SSIM) from Wang et al. [13] introduces the possibility of choosing the importance exponent for each of the three factors.

Many other image difference metrics using different approaches are available in the literature. For an extensive and detailed description of these metrics we refer the reader to the survey from Pedersen and Hardeberg [1].

3 The New Metric

Recent studies have shown that contrast is an important image attribute that falls under the umbrella of image quality [14]. In 2000 Tadmor and Tolhurst [15] developed a local contrast measure based on the DOG receptive-field model, modified and adapted to natural images. The conventional model describes the spatial sensitivity in the center of receptive fields (central component) by a bi-dimensional Gaussian with a peak amplitude at 1.0:

$$Center(x, y) = \exp \left[- (x/r_c)^2 - (y/r_c)^2 \right],$$

where x and y indicate the row and the column of the pixel (x, y) , and r_c is the radius of the Gaussian. The surround component is represented by a Gaussian curve as well, with a larger radius r_s :

$$Surround(x, y) = 0.85 (r_c/r_s)^2 \exp \left[- (x/r_s)^2 - (y/r_s)^2 \right].$$

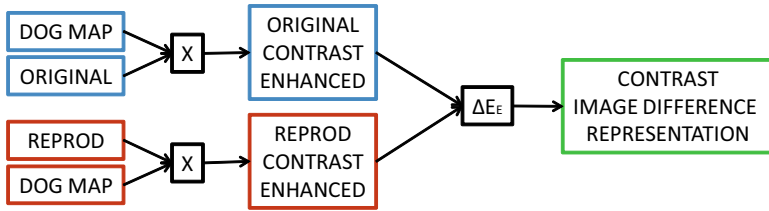


Fig. 1. Workflow of $M_{DOG} - DEE$

For a central point of the receptive-field positioned at (x, y) , the output of the center component for an image pixel at position (i, j) is given by:

$$R_c = \sum_i \sum_j Centre(i - x, j - y) Picture(i, j),$$

while the output of the surround component is:

$$R_s = \sum_i \sum_j Surround(i - x, j - y) Picture(i, j).$$

The following criterion for the measure of contrast was proposed, where the response gain is set by the local mean luminance:

$$C(x, y) = (R_c(x, y) - R_s(x, y)) / (R_c(x, y) + R_s(x, y)).$$

The DOG model has revealed to be beneficial in the identification of edges and blocks, and it has been extensively tested as contrast measure by Simone et al. [16] and as image difference metric by Ajagamelle et al. [11].

In our novel approach the DOG model is used as weighting map for each pixel before applying a color difference formula, in order to give more importance to those regions where edges and blocks can appear due to compression artifacts. As color difference formula we have selected the ΔE_E from Oleari et al. [7]. The workflow of this metric, that we call $M_{DOG} - DEE$ is shown in Figure 1.

We selected several combinations of r_c and r_s in order to see whether a particular configuration would result in better adequacy with the subjective evaluation. We have chosen the following combinations as suggested in [16]: $r_c = 1$ and $r_s = 2$ (C1); $r_c = 1$ and $r_s = 3$ (C2); $r_c = 2$ and $r_s = 3$ (C3); $r_c = 3$ and $r_s = 4$ (C4).

4 Psychophysical Experiment

4.1 Creating the Dataset

A total of 10 images have been chosen, covering a wide range of distortions and a wide range of scenes, for the evaluation of several image difference metrics. The test database has been selected according to the quality attributes that are

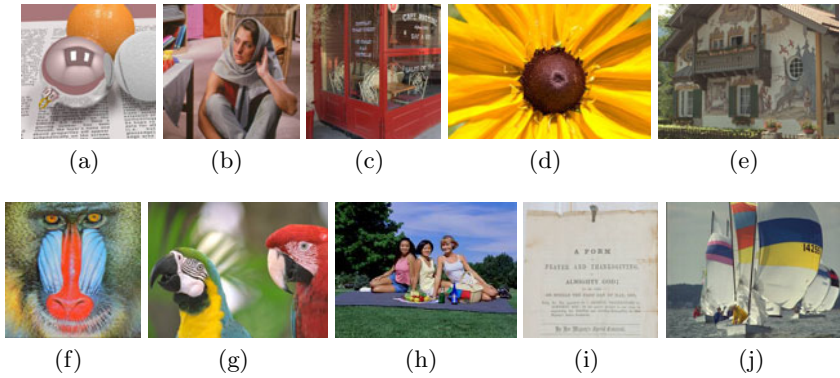


Fig. 2. Scenes used for creating the dataset

being evaluated following the recommendations of Field [17] and the CIE [18]. The chosen images are shown in Figure 2. Three images (Figure 2(c), Figure 2(d) and Figure 2(i)) were captured and provided by one of the authors; Figure 2(e), Figure 2(g) and Figure 2(j) were selected from TID2008 database [19]; Figure 2(h) was selected from a standard natural image set provided by the CIE [20]; Figure 2(a) was selected from High Dynamic Range Imaging - Acquisition, Display and Image-Based Lighting [21]; Figure 2(b) was selected from Alan Gersho's lab at U.C. Santa Barbara; Figure 2(f) was selected from Le Callet database [22].

Four different levels of compression were used for JPEG and JPEG200, resulting in eight compressed images for each scene and a total number of 80 test images. The compression rate in bit per pixel (bpp) was calculated as:

$$bpp = (file\ size\ in\ bytes \times 8 / image\ size\ in\ pixels).$$

For the experiment the compression rates were chosen such that JPEG and JPEG2000 have similar values of the bpp . Table 1 reports the bpp used to compress each scene. All the compressed images were generated with the quality

Table 1. Selected $bpps$ for each image

Figure	bpp JPEG and JPEG2000			
2(a)	0.8783	0.9331	1.0074	1.1237
2(b)	0.9614	1.0482	1.1003	1.2569
2(c)	0.8293	0.9080	0.9869	1.0977
2(d)	0.6611	0.7226	0.7844	0.8726
2(e)	0.9817	1.0500	1.1445	1.2916
2(f)	1.4482	1.5910	1.7057	1.9366
2(g)	0.6230	0.6891	0.7372	0.8169
2(h)	1.0353	1.1338	1.2351	1.3765
2(i)	0.4148	0.4474	0.4875	0.5420
2(j)	0.7518	0.8198	0.8774	0.9927

factor in the range 0-100 for JPEG and JPEG2000 with the associated *bpp* and a pre-test was carried out to choose the range of the just noticeable distortion (JND).

4.2 Experimental Setup

The experiment was conducted using the category judgement method [23]. The observer was instructed to judge an image according to quality, where the quality of the image is assigned to one of seven categories. All pairs of images were presented to the 18 recruited observers with the original one on the left and one compressed with random *bpp* and random type of compression (JPEG or JPEG2000) on the right side of the display. Each pair of images were displayed on an Eizo ColorEdge CG241W digital LCD display. The monitor was calibrated and profiled using GretagMacbeth, Eye-One Match 3. The settings on the monitor were sRGB with 40% brightness and a resolution of 1600 × 1200 pixels. The experiment took place in a windowless room with neutral grey walls, ceiling and floor. The ceiling lights in the room was set to provide a level of ambient illumination around 32 lux, which is below the upper threshold recommended by the CIE (64 lux) [24]. The white point was set to the D65 white point and the gamma is set to a value of 2.2. The display was placed at a viewing distance of 70 cm.

From the pair comparison experiment z-scores were calculated. In Figure 3 the z-scores with confidence intervals for all images are displayed. The results that we obtained are in agreement with what we expected, we have obtained an ascending order, from the image with the lowest *bpp* to the image with the highest *bpp* for JPEG and the same for JPEG2000. As the JPEG2000 bottom value is higher than JPEG we can have higher compression with small differences.

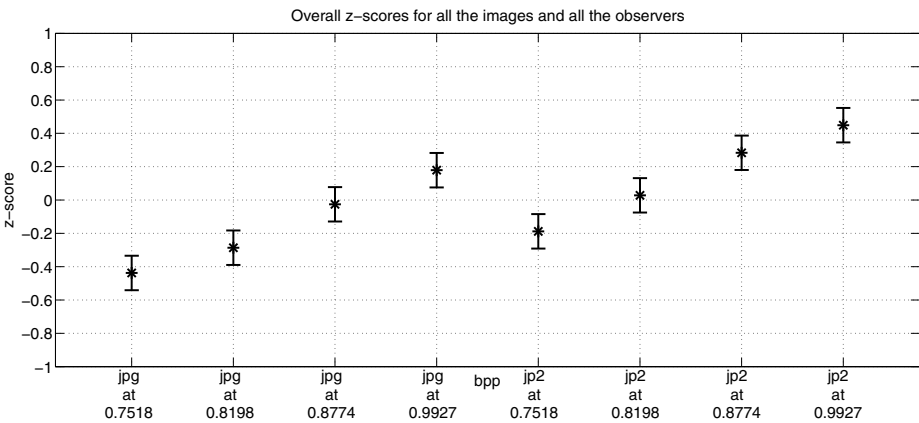


Fig. 3. Z-scores of all images

For this we can claim that JPEG2000 seems to perform better than JPEG (in agreement with the theory) [25]. The range between the highest and lowest values for JPEG2000 is larger than in the range in the JPEG and because of this it seems we have more perceived difference among the compression levels. The range between the highest and the lowest values in Figure 3 is quite narrow, in accordance with the fact that we were interested in evaluating the JND.

5 Experimental Results

In order to reveal potential differences between the methods and how $M_{DOG} - DEE$ performs on this particular dataset, two types of correlation were computed: the Pearson product-moment Correlation Coefficient (CC), which assumes that the variables are ordinal, and finds the linear relationship between them; the Spearman rank CC, which is a non-parametric measure of correlation that uses the ranks as basis instead of the actual values, thus the relationship between the variables is described without making any assumptions about the frequency distribution.

Table 2. Pearson CC for the tested metrics. In cyan the most performant metrics, while in red the least performant ones.

Metric—Database	Proposed	Pedersen	IVC	Ajagamelle	Fabienne	TID2008
$M_{DOG} - DEE$ (C1)	0.078	0.178	-0.010	0.709	0.034	-0.201
$M_{DOG} - DEE$ (C2)	0.086	0.188	0.011	0.698	0.048	-0.242
$M_{DOG} - DEE$ (C3)	0.081	0.188	0.025	0.699	0.029	-0.305
$M_{DOG} - DEE$ (C4)	0.087	0.183	0.035	0.692	0.026	-0.315
Maximum Difference	0.451	0.108	0.653	-0.534	0.081	0.189
UIQ	0.050	0.446	0.819	0.634	0.313	0.622
SSIM	0.294	0.217	0.705	0.635	0.159	0.550
ΔE_{ab}^*	0.156	0.764	0.539	0.751	-0.025	0.232
S-CIELAB	0.242	0.798	0.705	0.675	-0.067	0.433
$S - CIELAB_{JOHNSON}$	0.199	0.778	0.485	0.584	0.016	0.318
$S_{DOG} - CIELAB$	0.232	0.201	0.288	0.407	0.047	0.385
Hue Angle	0.065	0.805	0.345	0.625	0.006	0.262
SHAME	0.171	0.802	0.662	0.499	0.042	0.300
SHAMEII	0.126	0.827	0.458	0.622	-0.100	0.408
ΔE_E	-0.041	0.183	0.023	0.597	-0.003	0.273
$S - DEE$	0.080	0.179	0.295	0.402	-0.002	0.294
$S_{DOG} - DEE$	0.146	0.201	0.655	0.474	0.020	0.328
ΔE_{00}^*	0.134	0.626	0.383	0.739	0.034	0.086
RMS	0.267	0.750	0.605	0.746	0.063	0.536
MSE	0.264	0.666	0.510	0.614	0.086	0.535
Structural Content	-0.255	0.058	0.234	-0.702	-0.017	0.025
Average Difference	-0.207	-0.015	-0.008	-0.733	-0.025	0.153
PSNR	0.312	0.656	0.671	0.723	0.045	0.508

Table 2 shows the Pearson CC of the proposed metric and other state of the art metrics in addition to color difference formulae and several numerical objective quality measure as MSE, RMS, PSNR, structural content, average difference, and maximum difference [26]. Due to page limitations we will present only a selection of the calculated results.

$M_{DOG} - DEE$ shows a really poor correlation, also using different combinations of r_c and r_s indicating that the metric is not able to predict observers perceived difference in relation to compression artifacts. Maximum Difference and PSNR are the most performant metrics while Structural Content, Average Difference and ΔE_E are the least performant, showing a negative correlation. It is interesting to see that all the metrics using ΔE_{ab}^* perform better than the ones using ΔE_E , and RMS and MSE, perform better than most of the metrics. The overall conclusion is that none of the tested metrics show a good correlation, indicating an inefficiency in predicting perceived compression on this dataset. Spearman CC gives very similar results.

In order to investigate the performance of $M_{DOG} - DEE$ extensively and to see if it is suitable for other image quality purposes, we have tested it on five other datasets:

- Luminance changed images: this database from Pedersen [27] includes four original images reproduced with different changes in lightness. Each scene has been altered in four ways globally and four ways locally.
- IVC database: the IVC database from Le Callet et al. [22] contains blurred images and images distorted by three types of lossy compression techniques (JPEG, JPEG2000, and Locally Adaptive Resolution).
- Images altered in contrast, lightness, and saturation: this database from Ajagamelle contains a total of 10 original images covering a wide range of characteristics and scenes [10]. The images were modified on a global scale with separate and simultaneous variations of contrast, lightness, and saturation.
- Gamut mapped images: this database from Dugay et al. [28] is composed of 20 original images, which were gamut mapped with five different algorithms.
- TID2008: this database from Ponomarenko et al. [19] contains a total of 1700 images, with 25 reference images and 17 types of distortions over 4 distortion levels.

$M_{DOG} - DEE$ shows good correlation only on the database of images altered in contrast, lightness, and saturation from Ajagamelle [10], but it does not outperform other state of the art metrics and it turns out to be the least performant on the TID2008 database. Furthermore it is interesting to notice that none of the tested metrics show a good correlation in the database of gamut mapped images from Dugay et al. [28]. For all the databases Spearman CC gives very similar results. In conclusion the results from this analysis supports the findings from the first dataset, where using the DOG model as weighting map does not improve the performance of state of the art metrics in predicting perceived compression distortions and image quality.

6 Conclusions and Perspectives

In this paper the Difference of Gaussians model has been investigated with the purpose to see if it is able of predicting observers perceived difference in relation to compression artifacts. For that purpose a new image difference metric has been developed and a psychophysical experiment was conducted. A dataset of 80 compressed images were generated from 10 different scenes, based on different levels of compression via JPEG and JPEG2000. From psychophysical experiment it is easy to see that as expected JPEG2000 performs better than JPEG, having higher compression with small differences in accordance to observers perceived distortion. The metric developed and most of the tested state of the art image difference metrics show a poor correlation with the viewers perceptual quality ratings, indicating an inefficiency in predicting perceived compression distortions on this dataset. Furthermore, from extensive tests on five other databases, it can clearly be seen that the increased computational complexity of the recent image difference metrics are not proportional with the obtained performance improvement over simpler metrics.

References

1. Pedersen, M., Hardeberg, J.Y.: Survey of full-reference image quality metrics. Høgskolen i Gjøviks rapportserie, vol. 5. Gjøvik University College, The Norwegian Color Research Laboratory, Gjøvik, Norway (2009)
2. Luo, M., Cui, G., Rigg, B.: The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research and Application* 26, 340–350 (2001)
3. Zhang, X.M., Farrell, J., Wandell, B.: Application of a spatial extension to cielab. In: *IS&T/SPIE Electronic Imaging 1997*, vol. 3025, pp. 154–157 (1997)
4. Johnson, G.M.: Measuring images: differences, quality and appearance. PhD thesis, Rochester Institute of Technology (2003)
5. Johnson, G.M., Fairchild, M.D.: Darwinism of color image difference models. In: *IS&T/SID 9th Color Imaging Conference*, Scottsdale, AZ, USA, pp. 108–112 (2001)
6. Hong, G., Luo, M.R.: New algorithm for calculating perceived colour difference of images. *Imaging Science Journal* 54, 86–91 (2006)
7. Oleari, C., Melgosa, M., Huertas, R.: Euclidean color-difference formula for small-medium color differences in log-compressed osa-ucs space. *Journal of the Optical Society of America* 26, 121–134 (2009)
8. Huertas, R., Melgosa, M., Oleari, C.: Performance of a color-difference formula based on OSA-UCS space using small-medium color differences. *Journal of the Optical Society of America* 23, 2077–2084 (2006)
9. Simone, G., Oleari, C., Farup, I.: Performance of the euclidean color-difference formula in log-compressed OSA-UCS space applied to modified image-difference metrics. In: *11th Congress of the International Colour Association (AIC)*, Sydney, Australia, p. 81 (2009)
10. Ajagamelle, S.: Analysis of the difference of gaussians model in perceptual image difference metrics. Master's thesis, Gjøvik University College and Grenoble Institute of Technology (2009)
11. Ajagamelle, S.A., Pedersen, M., Simone, G.: Analysis of the difference of gaussians model in image difference metrics. In: *5th European Conference on Colour in Graphics, Imaging, and Vision (CGIV)*, Joensuu, Finland, pp. 489–496 (2010)

12. Wang, Z., Bovik, A.: A universal image quality index. *IEEE Signal Processing Letters* 9, 81–84 (2002)
13. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE TIP* 13, 600–612 (2004)
14. Pedersen, M., Bonnier, N., Hardeberg, J.Y., Albrechtsen, F.: Attributes of image quality for color prints. *Journal of Electronic Imaging* 19, 011016-1– 011016-13 (2010)
15. Tadmor, Y., Tolhurst, D.: Calculating the contrasts that retinal ganglion cells and LGN neurones encounter in natural scenes. *Vision Research* 40, 3145–3157 (2000)
16. Simone, G., Pedersen, M., Hardeberg, J.Y.: Measuring perceptual contrast in digital images. *Journal of Visual Communication and Image Representation* (2010) (under review)
17. Field, G.G.: Test image design guidelines for color quality evaluation. In: *IS&T/SID 7th Color Imaging Conference*, Scottsdale, AZ, USA, pp. 194–196 (1999)
18. CIE: Guidelines for the evaluation of gamut mapping algorithms. Technical Report, CIE TC8-08 (156:2004) ISBN: 3-901-906-26-6
19. Ponomarenko, N., Lukin, V., Egiazarian, K., Astola, J., Carli, M., Battisti, F.: Color image database for evaluation of image quality metrics, pp. 403–408 (2008), <http://www.ponomarenko.info/tid2008.htm>
20. ISO: Graphic technology - prepress digital exchange. Technical report, ISO 12640-2, 1 edn. (2004)
21. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P.: *High Dynamic Range Imaging - Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publisher, San Francisco (2005)
22. Le Callet, P.A.: Subjective quality assessment irccyn/ivc database 2005. In: *IRC-CyN* (2005)
23. Engeldrum, P.G.: *Psychometric Scaling*. Imcotek Press, Winchester (2000)
24. Djik, J.: In search of an objective measure for the perceptual quality of printed images. PhD thesis, Technische Unisersitet Delft (2004)
25. Caracciolo, V.: Just noticeable distortion evaluation in color images. Master's thesis, Gjøvik University College and Roma Tre University (2009)
26. Eskicioglu, A., Fisher, P., Chen, S.: Image quality measures and their performance. *IEEE Transactions on Communications* 43, 2959–2965 (1995)
27. Pedersen, M.: Importance of region-of-interest on image difference metrics. Master thesis, Gjøvik University College (2007)
28. Dugay, F., Farup, I., Hardeberg, J.Y.: Perceptual evaluation of color gamut mapping algorithms. *Color Research & Application* 33, 470–476 (2008)

Distance Field Illumination: A Rendering Method to Aid in Navigation of Virtual Environments

Matt Boggus and Roger Crawfis

The Ohio State University

boggus@cse.ohio-state.edu, crawfis@cse.ohio-state.edu

Abstract. In this paper we introduce a new use for distance fields: lighting of three dimensional environments to assist in navigation. Studies on visual search have shown that it is easy to locate an object if it has contrast in color or luminance with other elements in the scene. Since luminance and color guide visual attention, their application to aid in navigation is a natural extension. A user study comparing distance field illumination with existing navigational aids verifies the effectiveness of the new technique.

1 Introduction

Exploring a virtual environment can be difficult due to the lack of internal (idiothetic) sensory information. Using full human motion as input brings the sense of motion and idiothetic cues into virtual worlds, but this technology is still far from wide adoption. Instead, since adding a small amount of geometry to a virtual world is cheap, a common approach to this problem is providing more external (allothetic) sensory information in the form of visual or aural cues. When considering this methodology, there are two key questions to answer. First, is the method effective? The user must be able to interpret it. Second, does it fit with the goal of the application? For example, in a training simulation we want to reinforce learning, so we may desire providing subtle hints rather than a direct solution.

In this paper we introduce the use of distance fields for lighting and shading of polygonal models to assist in navigation within virtual environments. In computer graphics research literature, distance fields are defined as a scalar field where each value stores the distance to the closest piece of geometry in the environment. In this paper we allow construction of distance fields based on distance to objects within the scene. It can also be thought of as distance to a subset of the scene geometry. We propose that distance illumination is a novel, viable, and useful method to assist in navigation within a virtual environment. We evaluate this claim by comparing its use with existing navigational aids such as pointing devices, maps, and paths.

2 Related Work

We assume that there are some objects or points of interest, which we call goals or destinations, in the virtual environment that the user is looking for. There are two

main subtasks of navigation: exploration and search [16]. The user explores the environment to discover the goal locations, then searches for a path through the environment to reach them. Darken and Sibert demonstrated the effectiveness of navigational aids such as landmarks and mini-maps [5], so we presume if distance based illumination is as good as these existing techniques its use is validated. Ruddle and Lessels categorized evaluation methods for navigation aids [13]. Our user study provides a qualitative performance measure based on time spent to locate goals within a complex virtual environment and a questionnaire for cognitive rationale.

There are two ways of incorporating navigational aids in virtual environments. The first is by adding another object to the scene. Some examples are paths [14] [19], pointing devices [3], and multiple-resolution maps [9]. The other way is to alter the environment itself. This is done by placement of landmarks within the world [18]. Usually this occurs during the design of the environment, but the process can also happen dynamically [4].

Many navigational aids assume an outdoor environment where occlusion is limited. However, the challenges of navigation indoors are different [12]. New techniques like time rewinding after a missed turn [15] could prove to be more useful in a more closed virtual environment. When adding objects into a scene like a floating arrow, the feeling of presence within the environment may suffer. It is difficult to quantify presence and immersion [6][11], but it can be an important requirement in game applications [2].

With this in mind we introduce a new method to aid in navigation by subtle adjustment of luminance within the virtual environment. Early studies on visual search tasks showed that luminance can direct visual attention [17]. More recent results suggest that color is an effective visual cue when objects are on opposite ends of color space [20]. Results in [1] show that when altering an image, luminance change is more effective to guide visual search than warm-cool color modulation, though the authors also suggest the best method may be image-dependent. Luminance change is still effective at guiding visual search with distracters or false positives [10]. Most importantly, when only changing luminance, presence and immersion are not reduced [1]. Unlike many navigational aids, distance based illumination is independent of user location and viewing angle. Therefore, no additional work is required to support multiple users or changes in user position.

Two closely related works to this paper are [7] and [8]. The first is a study of dynamic placement of lights in computer games in order to stimulate visual attention. The problem addressed is getting the user to look at the correct place on screen rather than where to move within the environment. The second uses signed distance fields generated from all geometry in a scene in order to approximate occlusion of a hemispherical sky light surrounding the entire scene. The implementation in this paper differs in two ways: distance fields are generated using objects or a subset of geometry within the scene and distance values are directly mapped to illumination. Additionally, the topic of this paper is navigation whereas [8] attempts to approximate global illumination.

3 Navigational Aids

One way to classify navigational aids is by how much instruction they provide the user. Some provide step-by-step directions. The user can reach their destination by

following the directions without doing any route planning of their own. However, at any given point in time the user may not know the exact location of their destination. In contrast, other aids provide information about the destination and environment instead of giving directions. As long as the user is given enough information, they can plan their own route to determine which direction they should move. Support for multiple destinations is another way that aids can vary. One pointing device can only refer to one destination, so multiple pointers are required to scale up. In contrast, a single map can simultaneously provide information about many destinations. In this section, we provide several examples of navigational aids and introduce our navigation assistance method: distance field illumination.

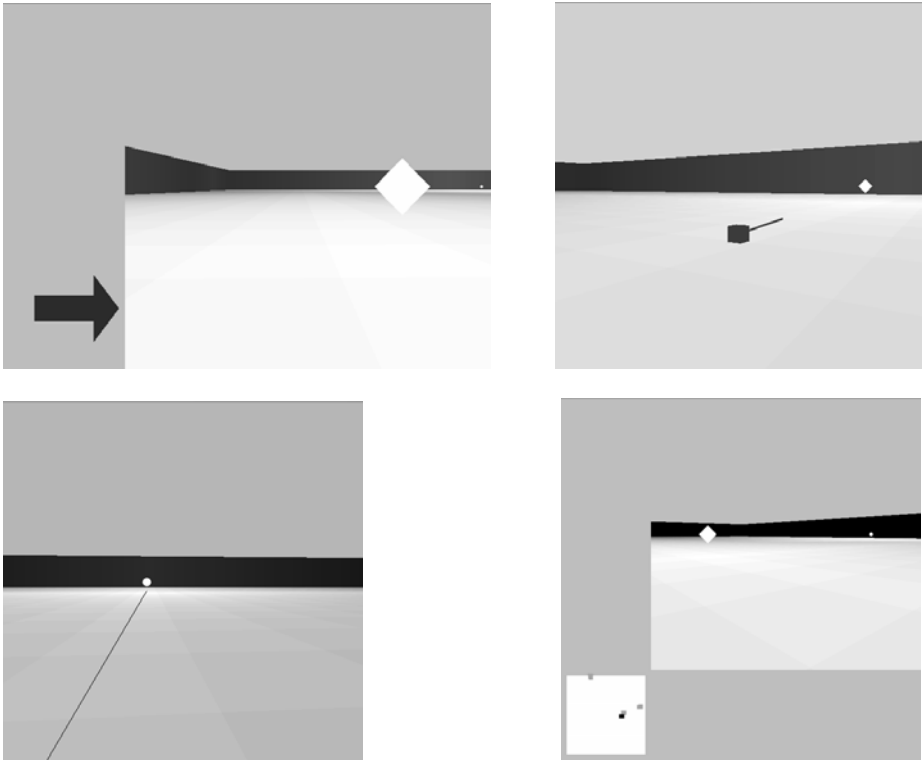


Fig. 1. Four navigational aids are shown. An arrow indicates that the user should turn right (top left). A line extends in the direction of a goal (top right). A path from the user to the goal is drawn within the environment (bottom left). A map depicts the locations of the user and goals within the environment (bottom right).

3.1 Screen Space Pointing Arrow

A screen space pointing arrow is shown in Figure 1 (top left). The arrow is placed to the left of the viewing window that displays the world. Its orientation is based on the shortest path to the goal and user's current heading. The arrow provides the action the

user should take to reach the goal: up and down mean move forward or backward while left and right indicate a direction to turn.

3.2 World Space Pointer

Alternatively, we can place a pointing device in world space. A line coming out of a cube indicates the vector from the user's current position to a goal. An example is shown in Figure 1 (top right). The pointer only provides information on direction to the goal's location. In cases where the goal is occluded, the aid will point towards a wall, not the path required to get to the goal as this would be misleading.

3.3 World Space Path

Rather than give directions to follow the shortest path, instead we can render the path within the world. This provides a guideline for the user to follow and the location of the goal when it is in view. This aid is shown in Figure 1 (bottom left). Paths can be rendered in screen space, but that requires combined use with the next navigational aid, a map.

3.4 Screen Space Map

Rendering a map in screen space allows the user to see their location as well as multiple goals. Minimaps can also give information about the environment, in this case we show walls that restrict movement. An example of the minimap, in a case without walls, is shown in Figure 1 (bottom right).

3.5 Distance Based Illumination

The concept behind distance field illumination is to brighten areas closer to goals, essentially highlighting regions. First we construct a distance field based on the

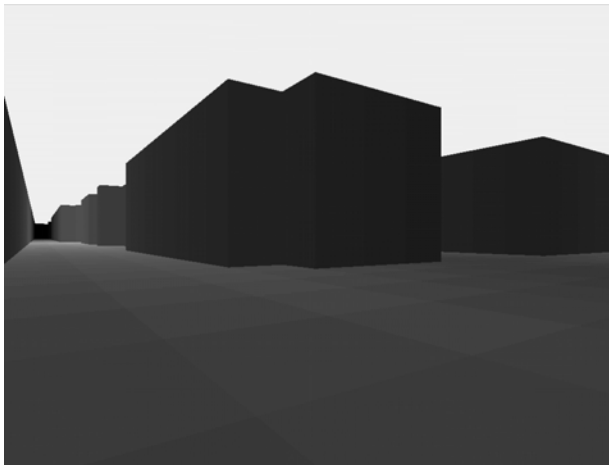


Fig. 2. Brighter illumination to the left of the user indicates a goal is closer in this area than the darker regions on the right

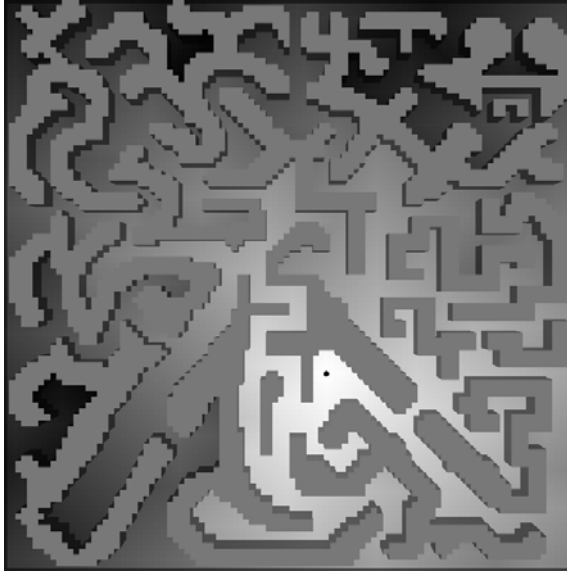


Fig. 3. The environment used in the user study, as seen from above, is illuminated according to path-based distance to the goal, depicted as a black dot, within it

positions of source objects (in this case the goals) in the environment. Every point in the distance field stores the distance to the closest source. In this paper, we use path-based distance, so points occluded from sources end up with higher distance values. These distance values can be used to color or illuminate the scene. We can precompute the distance field and use its values as a lookup table accessible in constant time when setting color or material properties of polygons in the scene. Figure 2 shows an example of this technique. From the position of the viewer, light can be seen in the left hand side of the scene, encouraging the movement towards that area. Distance fields support several sources, so guidance to multiple goals is possible, allowing the user to choose which area they would like to explore first. Another view of distance illumination is shown in Figure 3, which shows an environment as seen from above. Areas close to the goal are much lighter than areas that are far away.

4 User Study

To test the viability of distance based illumination assisting in navigation, we developed an application that included the aids described in the previous section as well as the new technique. Virtual environments in the test program were defined by a 100 by 100 image, dark pixels indicating walls and light pixels indicating open space. Movement and viewing of the environment was done from a first person perspective. Goals were placed in the environment for the user to find. The pointing devices and path lead to one goal at a time and switched to another after each was found. The program was viewed on a seventeen inch LCD monitor in a room lit with fluorescent lamps.

The study consisted of a brief training period to familiarize users with the navigational aids and controls for the test program. The test program was structured as a series of five trials. Each trial consisted of finding three goals in the environment, aided by one of the navigation aids. Multiple goals were used to determine if the limitation of guiding to one goal at a time was a significant detriment to user assistance. The maze-like environment used in each trial is shown in Figure 3. The same maze was used in every trial to eliminate the variable of difficulty caused by the organization of the environment itself. The maze was designed to be complex in order to keep users from learning the layout which would skew results of later trials. Since the same maze is used for each trial, we did not want to use the same goal positions every time as users may recognize this pattern and memorize a route to find all of the goals quickly without any navigation aid. Since the position of goals was randomized, we present results in measures of time to complete the trials and the ratio of time to distance from each goal to the starting position. Due to the difficulty of the task, users were allowed to give up on trials. In this case, an indication of quitting a trial was recorded along with the time passed before doing so. After the test program was completed, users filled out a questionnaire reacting to their experiences with each of the aids.

We recruited seventeen volunteers of varying levels of familiarity with navigation in virtual environments. Most had little experience. Of the seventeen, seven reported no previous experience with three dimensional first person perspective computer games. Trials where users quit are omitted from the results. Six users gave up on the screen space arrow, thirteen on the world space pointer, one on distance illumination, and none on paths and minimaps. The average time to complete each trial is shown in Table 1 and the average time/distance ratio is shown in Table 2. The average time for distance illumination trials is the lowest, but on average the distance to reach the goals for this technique was slightly lower than others. Still, when factoring in distance, distance illumination was the second most efficient navigation aid. While the screen space arrow gave a solution to the problem, the information was discrete and limited to neighboring cells in the distance field so planning ahead was difficult. The limited information and easy misinterpretation of the global pointer causes delays due to having to bypass obstacles. The ease of use for paths was countered by the time spent backtracking in cases where the first goal found was actually the furthest away. The minimap provided consistent results, but was slower than paths and distance illumination since users had to mentally construct their own routes.

Since the majority of our user study consisted of novice users, we performed an additional study on an experienced user. After fifteen practice runs, the user was allowed to take the test program ten times. The results are shown in Tables 3 and 4. The average time to distance ratios are proportionally similar to that of the novices, though

Table 1. The average time (in seconds) to complete each trial is shown

	screen arrow	world pointer	world path	minimap	distance illumination
average	239.09	206.82	145.59	174.84	143.94
std.dev.	82.92	72.84	47.30	65.46	64.13

Table 2. The average time/distance ratio to complete each trial is shown

	screen arrow	world pointer	world path	minimap	distance illumination
average	0.77	0.75	0.45	0.58	0.53
std.dev.	0.30	0.19	0.15	0.24	0.25

Table 3. The average time (in seconds) for the expert to complete each trial is shown

	screen arrow	world pointer	world path	minimap	distance illumination
average	72.77	59.34	34.94	37.47	33.64
std.dev	38.46	21.53	14.07	9.49	6.28

Table 4. The average time/distance ratio for the expert to complete each trial is shown

	screen arrow	world pointer	world path	minimap	distance illumination
average	0.25	0.20	0.11	0.13	0.12
std.dev.	0.11	0.07	0.03	0.06	0.03

the ratio for the pointing cube is slightly lower in this case. Again, distance illumination is proven to be useful as a navigation aid in the challenge of exploring and searching a virtual environment.

On the questionnaire, the seventeen users selected how much they liked each navigation aid on a scale from 1 to 5, representing strong dislike to strong enjoyment, with 3 as a neutral rating. They were also prompted to explain their answers. Table 5 shows the average rating of each of the navigation aids. Unsurprisingly, the techniques with more efficient performance are rated higher. The higher the number of quit trials for a technique, the lower the average rating was. Considering that distance illumination is a new technique and does not map to a real world counterpart like paths and maps and that it is rated just slightly below them and well above the neutral rating of three is promising.

The user comments indicated frequent confusion in how to interpret pointing devices and that the map, path, and illumination methods were easier to follow. One user's comments were particularly insightful, "I liked the arrow because it was helpful to navigate with but it was not exactly precise. I hated the pointing cube, it was frustrating to use. The path, map, and lighting/coloring were the easiest. I like having a vision of where I was headed and I could move there more quickly". The comment of the user who quit the trial for distance illumination also provided insight, "[after I found the second goal], it went pitch black and I couldn't find my way out. Eventually I became really frustrated with this [aid]." The mapping from distance field value to illumination is important. There is a delicate balance between illuminating the entire environment and providing enough contrast between nearby areas to be noticeable.

Table 5. The average rating, on a range of 1-5, of each navigation aid is presented

	screen arrow	world pointer	world path	minimap	distance illumination
averages	3.24	2.24	4.47	4.59	4.12
std.dev.	1.09	1.35	0.72	0.62	0.93

The study shows that distance illumination is a viable navigation aid. Note that we are not arguing for the superiority of distance illumination over existing navigational aids, only that it is a useful method for navigation assistance. When it comes to selecting a navigation aid, ease of use and interpretation are extremely important, but the purpose of the application should be considered as well. Each navigation aid has its own set of pros and cons. One of distance illumination’s unique advantages is that it can provide navigational cues within world space by altering the scene itself rather than adding more objects to it. This is useful when immersion in the virtual world is desired. The technique provides information to make navigation easier without providing a solution as bluntly as a pointing device. The downside to distance illumination is that since it does not correspond to a navigational aid in the real world, it is dependent on giving sufficient contrast to capture the user’s visual attention strongly enough to encourage exploration of the brighter area closer to the point of interest.



Fig. 4. A recent example of a navigational aid, used in a commercial video game, is shown. In LEGO Harry Potter (released July 2010), the player can find the next area where the story progresses by following a path of “ghost studs” (small, semi-transparent discs) and proceeding through the passage marked with a similarly colored “ghost arrow”. While appropriate for the visual style of the environment, the same effects in a game with “realistic” lighting would look out of place and negatively impact the immersion of the player.

5 Conclusion

In this paper we introduced a new method to assist in navigation of virtual environments. Studies in perception have shown luminance to be an effective cue for aiding in visual search. We have shown that its use also extends to navigation. Our method scales well for additional users and goals. Distance illumination can be applied subtly to preserve immersion. In some cases, such as in Figure 4, a path of bright, translucent coins can fit seamlessly in the scene, but this is not the case for all games and simulations.

In practice, there is a pair of issues to consider when using distance illumination. First, the effect of distance illumination varies based on scene geometry and grid resolution. This parameter space is too large to exhaustively search, but further testing may result in heuristics for selecting resolution for a given scene. Second, scaling values must be chosen so that the effect of the distance illumination is visible, but not overwhelming the contribution of artist placed lights. The number of possible lighting conditions is innumerable, so again we hope to determine heuristics or guidelines for scaling distance illumination within an already lit scene. Manual selection of these parameters takes little time, but automated methods would be useful in extending the technique to include animation effects. Even as a static effect, there is a bright future for the use of distance field illumination within computer graphics.

References

1. Bailey, R., McNamara, A., Sudarsanam, N., Grimm, C.: Subtle gaze direction. *ACM Trans. Graph.* 28(4), 1–14 (2009)
2. Bostan, B.: Requirements analysis of presence: Insights from a RPG game. *Comput. Entertain.* 7(1), 1–17 (2009)
3. Burigat, S., Chittaro, L.: Navigation in 3D virtual environments: Effects of user experience and location-pointing navigation aids. *Int. J. Hum.-Comput. Stud.* 65(11), 945–958 (2007)
4. Cliburn, D., Winlock, T., Rilea, S., Van Donsel, M.: Dynamic landmark placement as a navigation aid in virtual worlds. In: *Proc. Symposium on Virtual Reality Software and Technology*, pp. 211–214 (2007)
5. Darken, R.P., Sibert, J.L.: Navigating large virtual spaces. *Int. J. Hum.-Comput. Interact.* 8(1), 49–71 (1996)
6. Darken, R.P., Bernatovich, D., Lawson, J.P., Peterson, B.: Quantitative measures of presence in virtual environments: the roles of attention and spatial comprehension. *Cyberpsychol. Behav.* 2(4), 337–347 (1999)
7. El-Nasr, M.S., Vasilakos, T., Rao, C., Zupko, J.: Dynamic Intelligent Lighting for Directing Visual Attention in Interactive 3D Scenes. *IEEE Transactions on Computational Intelligence and AI in Games* 1(2), 145–153 (2009)
8. Evans, A.: Fast approximations for global illumination on dynamic scenes. In: *ACM SIGGRAPH 2006 Courses*, pp. 153–171 (2006)
9. Lam, H.: Overview Use in Multiple Visual Information Resolution Interfaces. *IEEE Transactions on Visualization and Computer Graphics* 13(6), 1278–1285 (2007)
10. McNamara, A., Bailey, R., Grimm, C.: Search task performance using subtle gaze direction with the presence of distractions. *ACM Trans. Appl. Percept.* 6(3), 1–19 (2009)
11. Pausch, R., Proffitt, D., Williams, G.: Quantifying immersion in virtual reality. In: *Proc. SIGGRAPH*, pp. 13–18 (1997)

12. Ramirez, L., Deneff, S., Dyrks, T.: Towards human-centered support for indoor navigation. In: Proc. CHI, pp. 1279–1282 (2009)
13. Ruddle, R.A., Lessels, S.: Three levels of metric for evaluating wayfinding. Presence: Teleoper. Virtual Environ. 15(6), 637–654 (2006)
14. Salomon, B., Garber, M., Lin, M.C., Manocha, D.: Interactive navigation in complex environments using path planning. In: Proc. Symposium on interactive 3D Graphics, pp. 41–50 (2003)
15. Simon, A., Stern, C.: Active guideline: spatiotemporal history as a motion technique and navigation aid for virtual environments. In: Proc. Symposium on Virtual Reality Software and Technology, pp. 199–202 (2007)
16. Tan, D.S., Robertson, G.G., Czerwinski, M.: Exploring 3D navigation: combining speed-coupled flying with orbiting. In: Proc. CHI, pp. 418–425 (2001)
17. Theeuwes, J.: Abrupt luminance change pops out; abrupt color change does not. Perception & Psychophysics 57(5), 637–644 (1995)
18. Vinson, N.G.: Design guidelines for landmarks to support navigation in virtual environments. In: Proc. CHI, pp. 278–285 (1999)
19. Wan, M., Dachille, F., Kaufman, A.: Distance-field based skeletons for virtual navigation. In: Proc. Visualization, pp. 239–246 (2001)
20. Wolfe, J.M., Horowitz, T.S.: What attributes guide the deployment of visual attention and how do they do it? Nature Reviews Neuroscience 5(6), 1–7 (2004)

Indirect Shader Domain Rendering

Daqing Xue¹ and Roger Crawfis²

¹ Intel Corporation

daqing.xue@intel.com

² Ohio State University

crawfis@cse.ohio-state.edu

Abstract. This paper presents an indirect shader domain rendering technique to combine different shader rendering effects to visualize the investigating data. Multiple shaders are associated with the geometries or voxels in volumetric data. The shader is resolved at run time to be selected for rendering. Our indirect shader synthesizer provides a novel method to control the appearance of the rendering over multi-shaders. We demonstrate an interactive shader painting technique using our indirect shader synthesizer to generate highly informative images from both geometric and volumetric datasets.

1 Introduction

A shader in computer graphics normally indicates the rendering program to produce a 3D image from volumetric data and/or surface geometries. The shader can be implemented to run on both CPU and GPU. The modern graphics hardware provides the high programmability on GPU [1, 22]. The GPU-based shader has been becoming more and more efficient, flexible, and popular in real 3D applications [32].

Generally speaking, in a rendering pipeline, there are two kinds of shaders: vertex shaders and pixel shaders (or fragment shaders in OpenGL) [18, 31]. The vertex shader is responsible for transforming input vertices into clip space; pixel shader is responsible for shading the pixels from rasterization into the frame buffer. Most recently, more shaders such as geometry shader, hull shader, domain shader, tessellation shader, and compute shader [19] are introduced into the latest graphics hardware pipeline for new geometry primitive generation, geometry patch control, or parallel computation. We are more interested in using a shader program to generate pixel color in the final image. Hence, in this paper, we limit our discussion to pixel shaders and the term “shader” refers to a pixel shader if not specified.

In a single shader application, all pixels in the image are produced by the same shader program and rendering setup (light, texture, camera, etc). In this paper, we present an indirect shader synthesizer framework to combine different shader rendering effects to create a highly informative visualization of the input data. Our overall goals in this study include the examination of multi-shader rendering and demonstrate indirect shader synthesis technique in an interactive shader painting framework.

The remainder of this paper is organized as follows. Section 2 reviews related work in multiple shader rendering. Section 3 and 4 examines the shader operations and classifications, respectively. An indirect shader synthesizer framework is

described in section 5. Section 6 demonstrates a user-controlled shader painting framework based on our indirect shader synthesizer. We present the resultant images from our shader synthesizer in section 7 and conclude our study in section 8.

2 Related Work

In this section we will focus on previous work about multi-shader rendering and the synthesis between the different shading results.

McGuire [16] describes a shader framework called “SuperShader” that renders many effects on surfaces. It allows arbitrary combinations of the rendering effects to be applied to surfaces simultaneously. The effect shaders are generated and optimized at runtime. One key problem in multi-shader rendering system is to manage the rendering order of the shaders (a.k.a permutation problem). McGuire solves this problem by generating various shader source codes from source code snippets. Hargreaves [9] shows how to automatically expand large numbers of shader permutations from a smaller set of input shader fragments. McGuire et al. [21] present an “abstract shader tree” system for generating complex GPU shaders through automatic combination of primitive shading functions. In [17], McCool demonstrate an approach to connect and combine shader programs using algebraic operators. In a more recent work [33], Trapp and Döllner transform the shader source code fragment into an intermediate representation which is associated with the predefined semantics for combination at run time.

In volume visualization community, many multi-shader rendering work had been done in the field of focus+context rendering and importance-driven rendering. Kruger et al. [13] developed an interactive context preserving volume rendering to achieve focus+context rendering. They use multi-shaders to depict the different body parts like skin and bone. The final color is a weighted average of all shaders involved. Viola et al. [36] propose an importance-driven volume rendering. The voxel in the data is assigned an object importance which encodes a visibility priority. This property determines whether a more-important region is behind a less-important region. When this occurs, the less-important-region will be rendered with a reduced opacity. Thus the objects of interest are always clearly visible. They use Maximum Importance Projection (MIP) to determine the most important object location for each ray and the less-importance objects in the front are removed or become more transparent to achieve a cut-away view. Most recently, Plate et al. [29] developed a multi-volume shader framework to render the intersection between the datasets with different resolution. They define a set of convex polyhedral volume lenses associated with one or more volumetric datasets. The lenses can be interactively moved around while the region inside each lens is rendered using interactively defined multi-volume shaders. Their result shows the very promising shading behavior of the combination of the resultant imagery from multi-shaders.

Texture image synthesis has been well-studied in the past few years. We will focus the related work on the stitch or blend between multiple image patch on the arbitrary surface. Praun et al. [28] proposed a lapped texture technique for surface texture synthesis in 2000. They used an irregular texture patch and iteratively pasted it onto the surface. The placement of the patches is oriented according to a pre-defined vector

field on the surface. An alpha-mask for the patch is created for further alpha-blending to smooth the transition between the overlapped patches. Their approach can create very nice result. However, the method is sensitive to input textures and is unsuitable for textures with strong low-frequency components, boundary mismatches, or a singularity point. Most recently, Takayama et al. [34] extend this work into “lapped solid texture”. They classify the solid texture according to its tilability/anisotropy and a tensor field is created to match this tilability to guide the solid texture fill-in orientation inside the mesh. Similarly to 2D lapped texture, they created a 3D alpha mask for alpha blending between the lapped solid textures.

3 Shader Operations

We generalize a shader as a function mapping Φ :

$$\Phi : \mathbf{x} \times \mathbf{v} \rightarrow \mathbf{c} \quad (1)$$

where \mathbf{x} is the 3-tuple of the input pixel position, \mathbf{v} is a vector of attributes associated with each pixel, and \mathbf{c} is the output color for the input pixel. The attributes in equation 1 include, but are not limited to, normal, texture, texture coordinates, and view direction. These attributes are used in the shader program to compute the output pixel color \mathbf{c} .

By applying the unary or binary operations between one or more shaders attached to the same fragment, we can synthesize the result from multi-shaders and achieve different rendering effect. In our study, we explore the following operators:

- **Complement.** The resultant color is the complement of the shader output.
- **Min/Max.** The resultant color is the component-wise minimum/maximum between the outputs of the two shaders.
- **Over.** The resultant color is the blending composition between the outputs of the two shaders.
- **Replace.** One shader output replaces the other shader output as the final output.
- **Weight.** The two shader outputs are weighted and added together.

The outputs from the above operations are normalized (clamp) to fit the rendering pipeline. In addition, the shader operations can be cascaded.

4 Shader Classification

A pixel shader can be an arbitrary function mapping to generate the output color for any input pixel. We classify the shaders into several types such that we can use a different user interface to handle the synthesis between the shaders. Our classification is based on the functionality of the shaders. These shaders include null shader, photo-realistic shader, NPR shader, procedural shader, and volume rendering shader.

Null Shader. A null shader indicates no shading on the input pixel. The shader ID of 0 is reserved for the null shader in our shader synthesizer system. Null shader is especially useful to produce an erosion-like object [26].

Photorealistic Shader. Photorealistic shaders are most commonly used in graphics applications. The pixel color is computed based on local shading models [2, 27] or global illumination [4, 25]. Texture mapping can be applied in the shader program. In most cases, this shader tries to generate a photo-realistic effect image.

NPR Shader. Non-photorealistic rendering (NPR) focuses on enabling a wide variety of expressive styles to convey the most important information in the output image. In contrast to a photorealistic shader, which focuses on photorealism, NPR is inspired by artistic styles such as drawing, illustration, and animated cartoons [3, 6, 15, 20].

Procedural Shader. In a procedural shader, the pixel color is calculated by the procedural function such as the well-known Perlin's noise or turbulence functions [24, 26]. The procedural shader is very suitable for modeling and visualizing the realistic texturing of complex surfaces, especially simulating a sculpted appearance of objects. Marble color, wood ring, and the gaseous phenomenon can be generated from such procedural functions [8, 12].

Volume Rendering Shader. A volume rendering shader creates the image directly from volumetric data without generating the geometry. In volume rendering, by changing the blending mode to composite the samples along the ray casting into the volume, we can create an image with distinguished appearance to convey the different information of the data. Commonly-used shaders include direct volume rendering (DVR), X-Ray, and maximum intensity projection (MIP) [5, 14, 37]. A transfer function can be applied to perform post-classification [7] in DVR.

5 Indirect Shader Synthesizer

Indirect shader domain rendering is achieved via shader synthesizer.

5.1 Indirect Shader

We use the concept of *indirect shader* to perform shader synthesis. A shader is not directly associated with the pixel. In contrast, all shaders are stored in a lookup table and the proper shader is selected at run time using a shader ID. Our shader synthesizer framework is shown in Fig 1. A main shader is used to resolve the shader ID and the shading process is executed on the selected shaders. The outputs from the shaders are composited in the main shader according to the shader operators discussed in section 3. With the support of OpenGL 2.0 [31] or above versions, each shader is compiled into an independent shader object and linked into the main shader program.

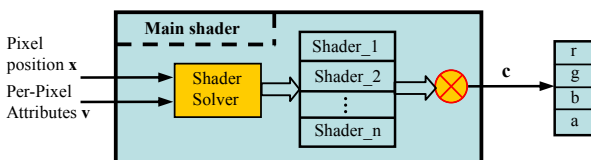


Fig. 1. The framework for indirect shader domain rendering

5.2 Shader Texture

A shader texture stores shader IDs and other metadata information for rendering. A simple shader synthesizer can be implemented using a 1D shader texture which functions as a look-up table of all applicable shader IDs. In the main shader, the shader solver in Fig 1 is simplified as a shader selector to index into the shader texture. Since the shader texture only contains the shader ID, the interpolation between the two adjacent entries in the texture is undefined and a NEAREST interpolation mode is used to locate the proper shader ID. Fig 2 Left shows a volume rendering image for a CT head dataset. The transfer function is used as the shader selector in which a silhouette enhanced NPR shader is applied if the opacity is greater than a given threshold otherwise a DVR shader is used. This simple shader texture technique may cause a sharp transition between the visual effects from different shaders (see Fig 2 Top Left).

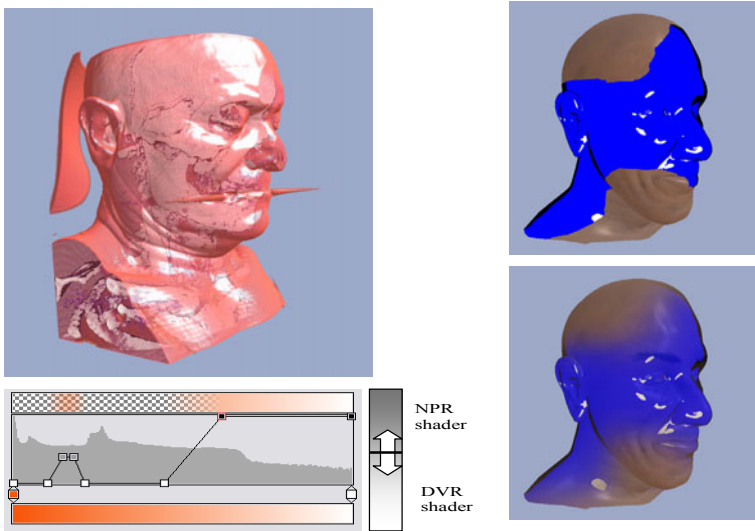


Fig. 2. Left: Silhouette enhanced NPR shader and DVR shader for a CT Head dataset. A transfer function is used as the shader selector shown bottom left. Right: An NPR shader (toon) and a photorealistic shader (texture mapping) are rendered on the Head model. A simple shader texture is used with hard edge between different shaders (top right); the layer-based synthesizer is used with the over operator to produce a smooth transition between shaders (bottom right).

5.3 Layer-Based Shader Synthesizer

In our layer-based shader synthesizer, an image layer is generated for each shader. These image layers are then blended together. Furthermore, we extend the shader operators defined in section 3 to gain more visual effects.

Multi-shaders are supported in our layer-based shader synthesizer. Each texel value in the shader texture is associated with a k -tuple defining the shader attributes. Generally speaking, a higher dimension of k -tuple can provide more properties for the underlying shader; however, it is limited by the hardware texture capability. In our

study, we resolve shader attributes via a triple: shader ID, shader operator ID, and an optional operator parameter. In our OpenGL implementation, this shader triple information can be encoded into a texture with the support of the integer internal format (GL_EXT_texture_integer) [23]. Integer texture guarantees the input integer data (texel value) will not be altered when downloaded into texture memory and fed into the shader. The integer texel value can be arbitrarily evaluated in the shader program. Fig 3 shows the diagram of a 4-channel 16-bit integer shader texture. The shader ID and operator ID are both encoded with 3 bits each, supporting up to 8 applicable shaders and 8 operators in the same pass rendering. The remaining 10 bits can be optionally encoded for the weight coefficients used in the weight operator or for other purposes. Fig 4 shows the pipeline of layer-based shader synthesizer and a smooth transition rendering between toon and photorealistic shaders using *over* operator (the enlarged resultant image shown in Fig 2 Bottom Right).

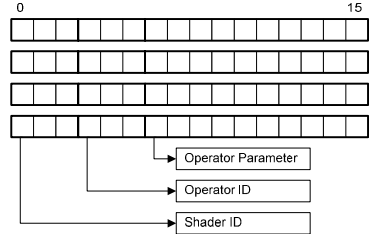


Fig. 3. A 4-channel shader texture. Each channel is encoded with shader ID, operator ID, and an optional parameter.

In volume shaders, there is no real image layer since we perform the shader operations between multi-shaders for each sample along the ray cast into the volume. We can imagine that there are virtual image layers of 1x1 pixels from the different shaders for all samples along the ray and perform a shader operation between these virtual image layers for all samples. The combination of the multi-shader output is finally blended into the frame buffer according to the rendering mode.

To improve the performance, we do not run all shaders on a given input pixel. Instead, only the non-zero shader ID specified in the shader texture is executed. The shader ID of 0 is viewed as a null shader and it is simply ignored.

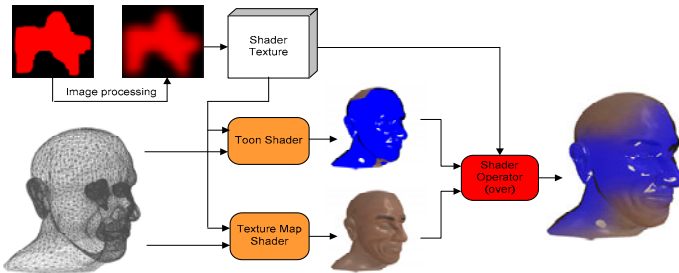


Fig. 4. The rendering pipeline uses a rich shader texture for the Head geometry and the over operator is applied between two image layers

6 Interactive User-Controlled Shader Painting

Our shader synthesizer system allows users to paint the shader onto a 2D surface or into volumetric data to produce desired shading effect. A shader brush is designed to associate different shader and operation parameters with painting area.

6.1 Paint on UV-Mapping

To support user-controlled shader painting on a surface, our system draws on a shader texture with UV-mapping. We use the VAMP mapping [30] in Right Hemisphere's Deep Paint 3D to create the UV-mapping. For simple geometry, an adaptive unwrapping technique [11] can be used to generate the UV-mapping directly.

Painting the shader ID onto the surface is equivalent to generate a 2D shader texture with the same as UV-Mapping, in which each texel contain the shader metadata as shown in Fig 3. We use an intermediate rendering buffer to record the shader texture coordinates (u, v) in the UV-mapping for every pixel in the frame buffer. We name this intermediate buffer as *I-buffer* which is only updated at painting time. When drawing on the surface, the shader texture is updated from the shader painting stroke. The shader ID in the shader texture is set to the new shader ID according to the painting stroke as shown in equation 2.

$$\text{ShaderTexture}[I_buffer[\text{PixelPos}]] = \text{shaderId} \quad (2)$$

The shader texture is then passed into the main shader to render the geometry with the new shaders. Fig 5 shows that different shaders are painted on the surface by user.

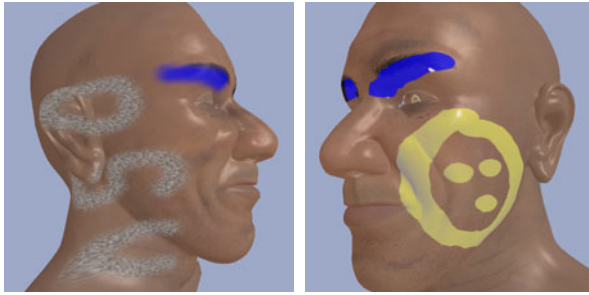


Fig. 5. The toon, granite, and gooch shaders are drawn on the head by user

6.2 Volumetric Painting

Multi-shaders can be tagged with pre-classified volumetric data to achieve rich rendering effects [36]. Here we describe a novel technique to associate shaders with the different brush strokes and paint on the homogeneous data.

6.2.1 Brush Stroke

Unlike surface geometry, there is no well-defined boundary inside a volumetric dataset. Instead, a transfer function is used to determine the opacity everywhere. It is difficult to determine the brush stroke depth when painting into a fuzzy volume. To solve this problem, we define the brush stroke as a ball-shaped mask with a fixed opacity value between 0 and 1 at the ball center. The opacity of the stroke ball is

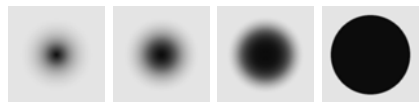


Fig. 6. The four brush stroke balls modulated by the different Gaussian functions

modulated by a three dimensional *Gaussian* function. Fig 6 shows the four brush strokes used in our system modulated by the different *Gaussian* functions.

6.2.2 Brush Stroke Placement

To find out the voxels painted by the brush, the brush (ball) center is placed at the sample point in the volume matching the brush opacity. Thus all voxels covered by the brush are associated with the brush’s shader ID.

To locate the point matching the brush’s center opacity, we use a ray caster to pick the matched point inside the volume. When the user paints into the volume using our painting widget, a single ray casting is performed starting from the volume boundary. Once the first sample opacity is hit with equal to or greater opacity than the stroke opacity, the ray casting is immediately stopped and the position of the sample is recorded. The voxels covered by the ball centering at the sample is associated with the shader ID of the brush stroke, as well as other information. Our method follows the instinct of “What You See Is What You Get” [10] in 3D painting.

6.2.3 Brush Stroke Union

The brush stroke is generated when the surface intersection point is determined or the point inside the volume is picked by ray caster. When there are overlapped region from multiple brush strokes, care needs to be taken to compute the shader parameters for the overlapped region. To preserve the continuity, we compute the union operation (see Equation 3) using the algebraic sum and algebraic product to represent the shader parameters within the overlapped region.

$$A \cup B = a(\mathbf{x}) + b(\mathbf{x}) - a(\mathbf{x})b(\mathbf{x}) \quad (3)$$

Here, $a(\mathbf{x})$ and $b(\mathbf{x})$ are the shader parameters from brush stroke A and B .

6.2.4 Painting Order

Painting order needs to be maintained for user-controlled shader painting within the overlapped region. The outputs of the shaders from shader texture will not be composited according to their order in the bit mask of the shader texture. An additional painting order parameter can be specified and stored into the shader texture as shown in Fig 3.

7 Experimental Results

All experimental images are generated on a PC with a GeForce 8800 graphics hardware and a Pentium Core 2 Q6600 2.6 GHz processor.

Fig 7 shows the combination between two shaders (granite and toon) using four operators: min, max, over, and weight. A smooth transition between two shaders only occurs for the *over* operator which combines the two shader results using alpha blending. The volumetric painting with multiple shaders is shown in Fig 8. Fig 8 (top) shows a CT head dataset is rendered with DVR shaders and then paint with a procedural granite shader and hader at different positions. Fig 8 (bottom left) shows MIP and NPR shaders are painted on DVR shader. A peeler shader (null shader) is presented in Fig 8 (bottom middle) to remove the soft tissue in the ROI.

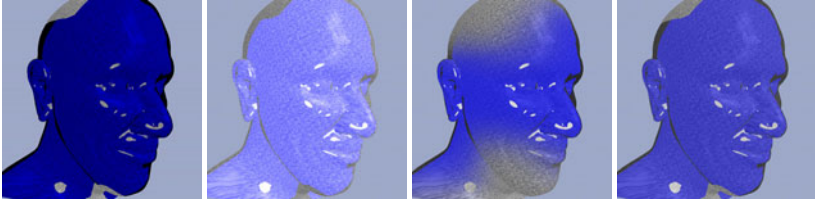


Fig. 7. The different operators between two shaders: granite procedure shader and toon NPR shader. From left to right: min operator, max operator, over operator, and weight operator with both coefficients as 0.5.

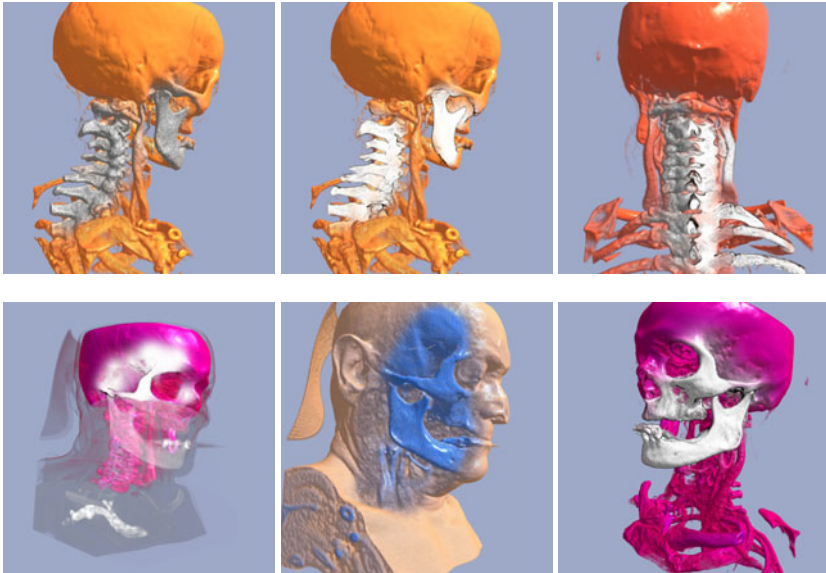


Fig. 8. Volumetric painting with multi-shaders. Top (left to right): DVR+Granite, DVR+NPR, DVR+Granite+NPR; Bottom (left to right): DVR+MIP+NPR, DVR+Peeler, R+granite+NPR.

8 Conclusions and Future Work

In this paper, we have presented indirect shader synthesis in an interactive shader painting framework. The resultant imagery indicates our indirect shader synthesizer provides a rich appearance control for the investigating data, including both surface geometries and volumetric datasets. The indirect shader synthesizer provides a novel and effective way to control the appearance of the rendering over the multi-shaders. The interactive volume painting instantly modifies the volume appearance which is specifically useful in surgery education and planning.

Our study only explored shaders focusing on fixed pixel position. We want to extend shaders with position- or time-dependent metadata information and thus create deformation and time-varying phenomenon rendering appearances.

References

1. ATI, <http://ati.amd.com/products/radeonhd4800/specs.html>
2. Blinn, J., Newell, M.: Texture and Reflection in Computer Generated Images. *Communications of the ACM* 19(10), 362–367 (1976)
3. Bruckner, S., Gröller, M.: Style Transfer Functions for Illustrative Volume Rendering. *Comput. Graph. Forum* 26(3), 715–724 (2007)
4. Dutré, P., Bala, K., Bekaert, P.: *Advanced Global Illumination*, 2nd edn. A K Peters Ltd., Wellesley (2006)
5. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume Rendering. In: *Computer Graphics, SIGGRAPH 1988* (1988)
6. Deussen, O., Strothotte, T.: Computer-Generated Pen-and-Ink Illustration of Trees. In: *Proceedings of SIGGRAPH 2000* (July 2000)
7. Engel, K., Kraus, M., Ertl, T.: High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. In: *Eurographics Workshop on Graphics Hardware 2001, ACM SIGGRAPH*, pp. 9–16 (2001)
8. Ebert, D., Musgrave, F., Peachey, D., Perlin, K., Worley, S.: *Texturing and Modeling: A Procedural Approach*, 3rd edn. Academic Press, New York (2003)
9. Hargreaves, S.: Generating shaders from hlsl fragments. In: Engel, W.F. (ed.) *ShaderX3: Advanced rendering with DirectX and OpenGL*, Thomson Learning (2005)
10. Hanrahan, P., Haeberli, P.: Direct WYSIWYG painting and Texturing on 3D Shapes. In: *Computer Graphics (SIGGRAPH 1990)*, vol. 24, pp. 215–223 (1990)
11. Igarashi, T., Cosgrove, D.: Adaptive Unwrapping for Interactive Texture Painting. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*, pp. 209–216 (2001)
12. King, S., Crawfis, R., Reid, W.: Fast Animation of Amorphous and Gaseous Volumes. In: *Volume Graphics 1999*, Swansea, UK, pp. 336–346 (1999)
13. Kruger, J., Schneider, J., Westermann, R.: ClearView: An Interactive Conetxt Preserving Hotspot Visualization Technique. In: *Proceedings of IEEE Visualization 2004* (2004)
14. Levoy, M.: Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics*, 245–261 (1990)
15. Lu, A., Morris, C.J., Ebert, D.S., Rheingans, P., Hansen, C.: Non-Photorealistic Volume Rendering Using Stippling Techniques. In: *Proceedings of IEEE Visualization 2002* (2002)
16. McGuire, M.: The SuperShader. In: *Shader X4: Advanced Rendering Techniques*, ch. 8.1, pp. 485–498 (2005)
17. McCool, M., du Toit, S., Popa, T., Chan, B., Moule, K.: Shader Algebra. In: *SIGGRAPH 2004*, pp. 787–795. ACM Press, New York (2004)
18. Microsoft. *DirectX9 SDK*
19. Microsoft. *DirectX11 SDK*
20. Markosian, L., Kowalski, M., Trychi, S., Bourdev, L., Goldstein, D., Hughes, J.: Real-Time Nonphotorealistic Rendering. In: *Proceedings of ACM SIGGRAPH 1997*, pp. 113–122 (1997)
21. McGuire, M., Stathis, G., Pfister, H., Krishnamurthi, S.: Abstract shade trees. In: *Symposium on Interactive 3D Graphics and Games* (March 2006)
22. Nvidia, http://www.nvidia.com/page/geforce_8800.html.
23. OpenGL, <http://www.opengl.org/registry>
24. Perlin, K.: An Image Synthesizer. In: *Proc. SIGGRAPH* (1985)
25. Pharr, M., Humphreys, G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco (July 2004)

26. Perlin, K., Hoffert, E.: Hypertexture. In: Proc. SIGGRAPH (1989)
27. Phong, B.T.: Illumination for Computer Generated Pictures. *Communications of the ACM* 18(6), 311–317 (1975)
28. Praun, E., Finkelstein, A., Hoppe, H.: Lapped textures. In: Proceedings of SIGGRAP 2000, pp. 465–470 (July 2000)
29. Plate, J., Holtkaemper, T., Froehlich, B.: A Flexible Multi-Volume Shader Framework for Arbitrarily Intersecting Multi-Resolution Datasets. In: Proc. of IEEE Visualization (2007)
30. Right Hemisphere Ltd. Deep Paint 3D (Texture Weapons), <http://www.righthemisphere.com>
31. Segal, M. and Akeley, K.: The OpenGL Graphics System: A Specification (Version 2.0 - October 22, 2004) (2004), <http://www.opengl.org/>
32. St-Laurent, S.: Shaders for Game Programmers and Artists, 1st edn. Course Technology PTR (2004)
33. Trapp, M., Döllner, J.: Automated Combination of Real-Time Shader Programs. In: Proceedings of Eurographics 2007, pp. 53–56 (2007)
34. Takayama, K., Okabe, M., Ijiri, T., Igarashi, T.: Lapped Solid Textures: Filling a Model with Anisotropic Textures. In: Proceedings of ACM SIGGRAPH 2008 (2008)
35. Tietjen, C., Isenberg, T., Preim, B.: Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning (2005)
36. Viola, I., Kanitsar, A., Gröllner, M.: Importance-Driven Volume Rendering. In: Proceedings of IEEE Visualization (2004)
37. Xue, D., Crawfis, R.: Efficient Splatting Using Modern Graphics Hardware. *Journal of Graphics Tools* 8(3), 1–21 (2003)

Visual Exploration of Stream Pattern Changes Using a Data-Driven Framework

Zaixian Xie, Matthew O. Ward, and Elke A. Rundensteiner

Computer Science Department
Worcester Polytechnic Institute
{xiezx,matt,rundenst}@cs.wpi.edu

Abstract. When using visualization techniques to explore data streams, an important task is to convey pattern changes. Challenges include: (1) Most data analysis tasks require users to observe the pattern change over a long time range; (2) The change rate of patterns is not a constant, and most users are normally more interested in bigger changes than smaller ones. Although distorting the time axis as proposed in the literature can partially solve this problem, most of these are driven by the user. This is however not applicable to streaming data exploration tasks that normally require near real-time responsiveness. In this paper, we propose a data-driven framework to merge and thus condense time windows having small or no changes. Only significant changes are shown to users. Juxtaposed views are discussed for conveying data pattern changes. Our experiments show that our merge algorithm preserves more change information than uniform sampling. We also conducted a user study to confirm that our proposed techniques can help users find pattern changes more quickly than via a non-distorted time axis.

1 Introduction

The term *data streams*, or *streaming data*, refers to data arriving at end-users in a continuous, unbounded, and normally very rapid way [1]. Many real-world examples exist, such as traffic monitoring, intensive care units in hospitals, and the stock market. Storing data, performing queries, and mining patterns are common tasks on data streams in order to retrieve useful information, understand associated phenomena, and provide support for decision-makers. Because of this wide usage, data stream analysis has attracted much attention in multiple areas of computer science, including database management and data mining [1].

As an efficient technique to help data analysis, visualization is also increasingly being employed to help users investigate data streams. This has resulted in some frameworks, algorithms, and techniques for preprocessing and visualizing streaming data, along with interaction techniques to explore it [2,3,4,5]. Although some of this work only focuses on time-series data, their techniques can often be applied to data streams. We can divide the tasks solved by these efforts into two categories: (1) maintaining and conveying data patterns of the current window; and (2) detecting and representing the pattern change over time. Not many researchers have focused on the second category, although this is an important part of stream analysis. Figure 1 shows one intuitive approach to visualizing pattern change in the traffic data. In this figure, each time window

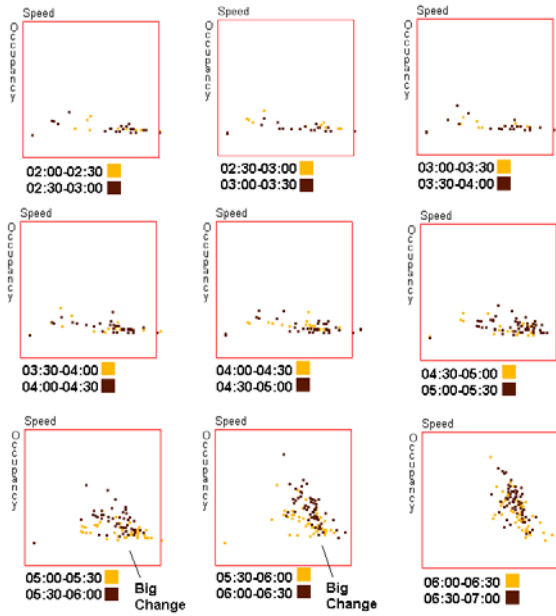


Fig. 1. A juxtaposed output using the traffic data of 5 hours from a specific sensor. 10 windows are shown in this figure. Each subfigure shows two contiguous windows. Newer data is in dark color, while the older data is light. Significant changes are buried in a lot of subfigures with few or no changes.

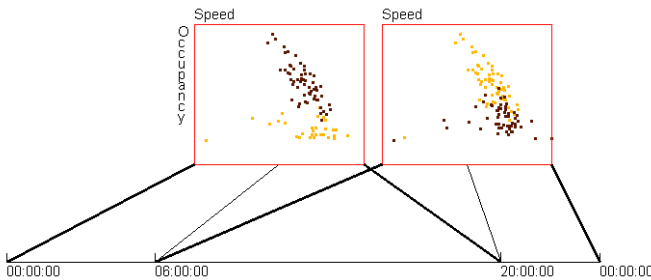


Fig. 2. 48 windows, containing the data in Figure 1, are merged to 3 windows and then shown with 2 scatterplots. Each scatterplot contains two windows, and is linked to the time axis via three lines to delimit the time range for these two windows. Newer data is in dark color, while the older data is light.

corresponds to 30 minutes; every subfigure shows two contiguous windows to convey the pattern change. This can help users detect changes of the fit line slope for the linear trend between *Occupancy* and *Speed*. However, because significant pattern change only happens when entering rush hours (see the subfigures with the label “big change”), such useful information is buried by other subfigures. This might result in a slow response rate, which is not acceptable for some applications. Moreover, the display canvas is wasted by a lot of subfigures with small or no changes.

In order to overcome the above shortcoming, our basic idea is to design algorithms to automatically merge windows with small or no changes and assign more screen space to periods having large pattern changes. Figure 2 shows a motivating example, where 48 original windows (24 hours) are merged to 3 windows and then visualized by 2 subfigures. Note that each subfigure contains the data in two windows, and is linked to the time axis via three lines (two thick and one narrow) to delimit the time ranges for these two windows. Obviously, Figure 2 reduces users' response time significantly, and merging maintains most of the information about recognizable changes of the fit line slope, the increasing at 6AM, and the decreasing at 8PM.

The main contributions of this paper include:

- We propose a framework to visualize data streams with the goal to show significant pattern changes to users. The main approach is to merge those windows with few or no changes when visualizing and storing recent as well as old data.
- The above framework is materialized using two important data patterns: linear trends and data range.
- We performed experiments to show that our merge algorithm can preserve more change information than uniform sampling. User studies were conducted to demonstrate that our techniques can significantly reduce users' response time when looking for significant pattern change in a data stream.

2 The Data-Driven Framework

Before describing the framework and algorithms, we give some terms.

Terms: If we merge windows $W_1, W_2, \dots,$ and W_k to a window W' , we call W' the **parent window** or **parent** of $W_1, W_2, \dots,$ and W_k , and $W_1, W_2, \dots,$ and W_k the **child windows** or **children** of W' . We also call $W_1, W_2, \dots,$ and W_k **original windows** and W' the **merged window**. **Current view** means the most recent n_0 windows over which users want to observe data pattern changes. Many multivariate data patterns can be described using a vector (v_1, v_2, \dots, v_r) , which we call a **pattern vector**. An example is the vector $(-2, 5)$ that describes a linear trend $y = -2x + 5$.

Merge Algorithm: We first explain how to merge n_0 windows to $n_m (\leq N_m)$ windows, where N_m is the maximal number of windows that the canvas can hold, and then some special requirements for handling streams will be discussed.

Data patterns in real applications normally need some complex presentations, such as a linear trend. However, in most cases, it is enough to represent the pattern change as a real number. For example, if we want to investigate how the slopes of fit lines change in a linear trend, we can define the pattern change as the slope difference between two contiguous windows. Based on this reasonable assumption, we use an example shown in Figure 3 to describe our merge algorithm. In this example, the pattern is described by a real number and the change is defined by the difference between two numbers. Actually we only consider the change instead of the patterns themselves. In Figure 3, the first row shows all 9 original windows with their pattern description. Recall that our approach is to merge contiguous windows if the change between them is small. Thus, the intuitive idea is to calculate all changes between contiguous windows and

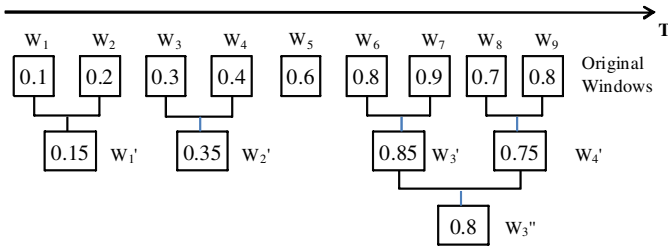


Fig. 3. An example to show how we do a multiple pass merge given a change magnitude $\delta = 0.1$. First, all pairs of windows are merged if their change magnitude is smaller than or equal to 0.1, and then the window list is scanned again to find window pairs that need to merge, until all changes between contiguous windows are bigger than 0.1.

merge those with small changes. In the first row (original windows) of Figure 3 we can find the following facts: (1) the change between neighbor windows is 0.1 or 0.2; (2) the pattern differences between W_1 and W_2 , W_2 and W_3 , W_3 and W_4 , all are 0.1, which is a small change compared to other changes. Can we merge W_1, W_2, W_3 and W_4 to one window? Absolutely no. The reason is that the data pattern is increasing steadily from W_1 to W_4 . The aggregate change is 0.3, which is not small. If we merge these four windows to one window, we will lose this important change. Therefore, in our merge algorithm, we only merge two windows at once. To explain this idea, assume that we are merging the window list $\{W_{n_0-1}, W_{n_0-2}, \dots, W_0\}$, where W_0 is the current window, and the change threshold is δ . We search the whole window list from the beginning, until we meet two contiguous windows, say W_j and W_{j-1} , having a change less than or equal to δ , and then merge them. After that, we do the same searching and merging from W_{j-2} until W_0 . For example, in Figure 3 in the first pass of searching and merging, we get four merged windows: W'_1, W'_2, W'_3 and W'_4 , given $\delta=0.1$. Note that a single pass scan is not enough because the change between a merged window and an original window, or two merged windows can be less than δ , e.g., W'_3 and W'_4 . Thus we need to do multiple pass searching and merging from the beginning of the window list for a given δ , until we cannot find a change less than δ in a complete pass. In Figure 3 we finally get a new window list $\{W'_1, W'_2, W'_3, W'_4\}$.

After multiple pass searching and merging given a threshold δ , the number of windows in the new window list is probably still bigger than N_m . Under this situation, we can increase δ and do searching and merging again. For flexibility, we allow users to provide a sequence, $\{\delta_i\}_{i=0}^p (\delta_i < \delta_{i+1})$. The searching and merging will be run on these δ values one by one until $n_m < N_m$. δ_p should be the maximal possible change to make this algorithm applicable to any input.

When we discuss the merge algorithm, we fix the number of original windows to n_0 . For data streams, if one new window arrives, the oldest window, namely the expired window, has to be removed from the current view before we add the new window to the visualization. For example, in Figure 3, if we have a new window W_{10} , then W_1 must be removed from the view. The easiest approach is to run the the merge algorithm again on this new window list $\{W_2, W_3, \dots, W_9, W_{10}\}$. Obviously this is not efficient

because the merge result of the last time period is not reused. Thus we handle the new arrival window via the following steps: (1) If the oldest window in the current view has been merged into other windows, decompose the oldest merged window and put all its child windows back to the window list. (2) Remove the oldest window from the window list. (3) Add the new window to the window list. (4) Run the merge algorithm on the new window list. Therefore, for the new window W_{10} , we run merge algorithm on $\{W_2, W_2', W_5, W_3'', W_{10}\}$ instead of $\{W_2, W_3, \dots, W_9, W_{10}\}$.

How to Merge Windows: We have two options to merge two windows: (1) We first do a union set operation on two windows and get a merged window. Then we apply uniform sampling to this merged window to reduce the number of datapoints to the size of one window. For example, if each window has 100 datapoints, we can get a window having 200 datapoints after union set operation. Then we apply uniform sampling to this merged window with 50% as the sampling ratio. (2) Once we get a complete time window in the data stream, we calculate and store its pattern in memory. Assume the patterns of two time windows W_1 and W_2 are denoted by V_{p_1} and V_{p_2} respectively. When we merge W_1 and W_2 to W' , we calculate the pattern of W' directly from V_{p_1} and V_{p_2} . This reduces the time to calculate the pattern for a merged window but can only be applied to some specific data patterns [6].

The two streaming datasets used in this paper are the following.

Traffic Data Stream: In Section 1, we showed a slice of this data stream, which is provided by Mn/DOT (the Minnesota Department of Transportation) [7]. There are more than one thousand sensors on highway entrance/exit ramps and main lanes throughout the Twin Cities metro area. Each detector can collect the following values every 30 seconds: (1) Volume: the number of vehicles passing the detector. (2) Occupancy: the percentage of time that the detector sensed a vehicle. (3) Speed: the average speed of vehicles passing the detector. We normally select one detector and retrieve its three measures during a specific time period, e.g., one day or one week, on the Mn/DOT website.

Sleep Data Stream: This data stream is a physiological dataset (Santa Fe time series competition data set B) selected from the PhysioBank archive [8]. It is recorded from a patient suffering from sleep apnea in a sleep laboratory. Since it is relatively long (about 4 hours at a frequency of 2Hz), we use it to simulate a data stream. This dataset has three measures: heart rate, chest volume (respiration force), and blood oxygen concentration.

3 Visualization of Patterns and Their Changes

In this section, we discuss the juxtaposed view to convey pattern changes over merged windows.

Assume that we have n windows, W_1, W_2, \dots , and W_n , we need to visualize. We generate $n - 1$ subfigures using standard multivariate visualization techniques, e.g., scatterplots or parallel coordinates. The first contains W_1 and W_2 ; the second shows W_2 and W_3 , and so on. This design is from our prior work [9] and enables users to quickly detect pattern changes in the visualizations.

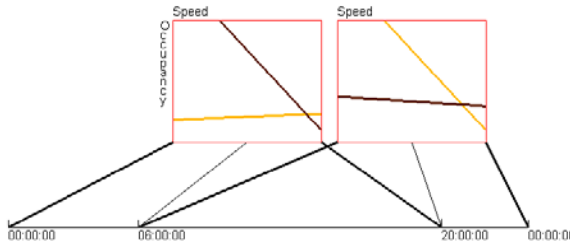


Fig. 4. A pattern outline view to visualize the pattern change in the traffic data slice used in Figure 2. Each line represents a linear model for a merged window.

In juxtaposed views, we develop two types of visualization techniques: (1) *juxtaposed full view* that uses traditional visualization techniques to show all datapoints in the windows (Figure 2); and (2) *juxtaposed pattern outline view* that shows only the outline of the discovered pattern for each window. Pattern outline view is specific to each pattern. For example, it can be a line for linear trends. Figure 4 shows a pattern outline view for the same data as Figure 2.

In Figures 2 and 4, all subfigures are placed on the canvas horizontally in the order of the timestamp. Because the time axis is evenly spaced and subfigures have different lengths of time range, we use lines to connect subfigures to the time axis. This can help users understand where the change is fast and where the change is slow. We call this a *1D even layout*.

The 1D even layout is intuitive to interpret, but it does not make full use of the canvas when the number of merged windows is large, especially for those visualization techniques that generate output in a shape close to square, such as scatterplots or parallel coordinates. In order to avoid this drawback, we propose a grid layout, in which we lay out all subfigures in a grid having n rows and n columns. If there are m subfigures, $n = \lfloor \sqrt{m-1} \rfloor + 1$. In grid views, the representation of the time axis is problematic. If we use the same method as the 1D even layout to connect the subfigures to the time axis via lines, we will encounter a lot of overlapping. We solve this problem using an interaction technique : when the mouse hovers over a subfigure, the corresponding time range will be highlighted on the time axis (Figure 5).

Figure 5 shows an example using the pattern outline view and grid layout. Each subfigure is a two-dimensional parallel coordinates. There are two bands in each subfigure. One band represents the data range in a time window. On dimension X, two corners of the rectangles correspond to $(\bar{X} + s)$ and $(\bar{X} - s)$ respectively. Note that \bar{X} is the mean value, and s is the standard deviation for dimension X in an arbitrary time window. In this figure, we can find two types of range: Type 1 (low heart rate and high blood oxygen concentration, e.g., the yellow band in the highlighted subfigure) and Type 2 (high heart rate and low blood oxygen concentration, e.g., the dark band in the highlighted subfigure). Our merge algorithm can automatically detect the shift between two types, as shown in Figure 5. From the time axis, we find that Type 2 normally only exists in a short time range, so it can be treated as an outlier. This might be associated with

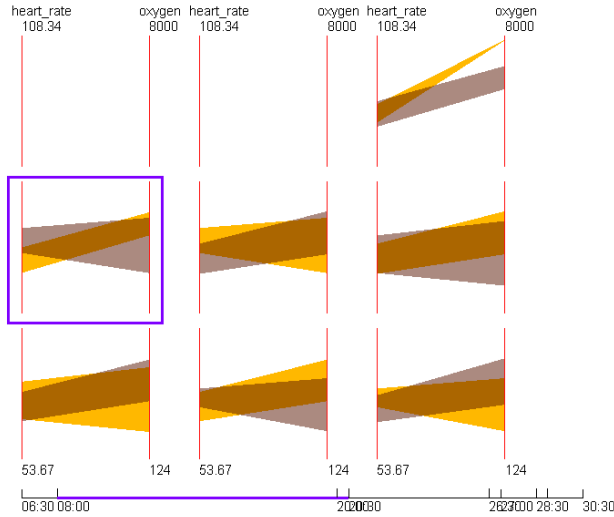


Fig. 5. A pattern outline view in grid layout to visualize the changes in data range for sleep data. A subfigure is highlighted with a purple border when the mouse hovers over it. The corresponding part of the time axis is highlighted as well.

sleep apnea (periods during which a patient takes a few quick breaths and then stops breathing for up to 45 seconds) [8].

4 Evaluation

In this section, we evaluate two important issues: (1) how well does the merge algorithm preserve the change information for data patterns? and (2) how much can the proposed techniques reduce users’ response time?

4.1 Measuring Result Quality of the Merge Algorithm

To the best of our knowledge, there are no existing algorithms designed and optimized for achieving the same goal as our proposed merge algorithm. Therefore, we chose uniform sampling as our competitor in this algorithm to evaluate the output quality. We first defined a measure for merging quality, and then ran our merge algorithm and uniform sampling on the traffic data stream to compute quality measures in different configurations.

In order to explain the quality measure, we show an example in Figure 6. The numbers in this figure have the same definition as Figure 3. In each subfigure, the top row represents the original windows; the merged windows are shown in the second row that will be visualized. Thus the actual change information perceived by users shown in the third row may be different from that in the original windows. For example, the original change between W_4 and W_5 is 0.4. In Figure 6(a) (our proposed merge algorithm),

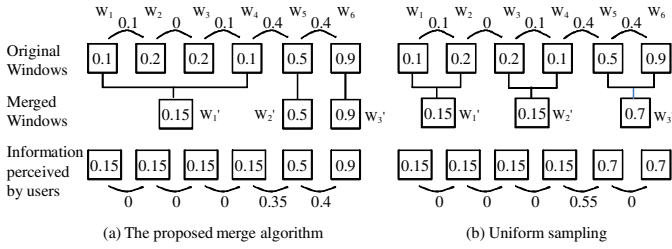


Fig. 6. This figure shows how to measure result quality of the merge algorithm and uniform sampling regarding the degree to which the change magnitude is preserved

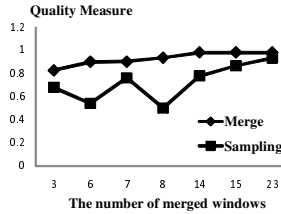


Fig. 7. The quality measures for merge algorithm and uniform sampling when merging a slice of traffic data in the current view with different number of merged windows

the perceived change by users is 0.35. This value becomes 0.55 in Figure 6(b) (uniform sampling). Obviously, regarding this change, our proposed merge algorithm has a better result than uniform sampling because $|0.35 - 0.4| < |0.55 - 0.4|$. Based on the above discussion, the formula to measure the result quality is given below:

$$Q = \frac{1}{n-1} \sum_{i=1}^{n-1} \left(1 - \frac{|\delta_i - \delta'_i|}{\delta_{max}} \right)$$

Note that n is the number of original windows; δ_i denotes the actual change between W_i and W_{i+1} ; δ'_i represents the perceived change; and δ_{max} means the maximal change. If $\delta_{max} = 1.0$, the quality measures for Figures 6(a) and 6(b) are 0.95 and 0.85, respectively. In addition, normally we are more interested in bigger changes than small ones, so we count only those changes bigger than a threshold δ_T when using the above equation. If we set $\delta_T = 0.2$, the quality measures for Figures 6(a) and 6(b) become 0.975 and 0.725, respectively. In this experiment, we chose $\delta_T = \pi/6$ and calculated quality measures for the results of uniform sampling and our proposed merge algorithm respectively. In this set of experiments, we aim to test merge algorithm by treating a slice of traffic data within one day as the current view. Hence the number of original windows is 48. Figure 7 shows the quality measures for seven different values as the number of merged windows.

We can make the following observations based on Figure 7: (1) The merge algorithm performs better than uniform sampling in all configurations of this experiment; (2) When we decreased the number of merged windows in the current view, the merge algorithm shows good stability, but uniform sampling does not.

4.2 Comparing Proposed Techniques with Uniform Time Axis

Our initial goal was to reduce users' response time for detecting pattern changes. To verify whether we have achieved this, we conducted a user study to compare users' response accuracy (RA) and response time (RT) on different visualization techniques. The techniques we tested included: (1) Juxtaposed views with the original windows; (2) Juxtaposed full views; (3) Juxtaposed pattern outline views. The first one is the competitor, and techniques 2 and 3 use the merged windows.

In this experiment, we chose the traffic data and set the length of the current view to one day. The target data pattern was linear trends. The length of one time window was 30 minutes. The number of merged windows is set to 6. We picked 2 sensors and generated 2 figures for each technique, resulting in 10 figures. Every participant was asked to observe each figure on a laptop monitor and answer: "When did the biggest change of the fit line slope happen?" Note that one figure using technique 1 contains 47 scatterplots, so we allowed users to apply zooming on figures when explore them. 8 graduate students in computer science participated in this user study. Since there was no significant difference for the RA using the three techniques, we only calculated the average RT shown in Figure 8 with 95% confidence interval, and compared the RT of different techniques using a paired samples t-test. The statistical result revealed that our proposed techniques (Techniques 2 and 3) have significantly shorter response time than the visualizations of the original windows ($p < 0.01$).

Based on the experiment results, we conclude that our proposed visualization techniques combined with the merge algorithm can significantly reduce users' response time when exploring linear trend changes on streaming data. In the future, we plan to introduce other data patterns, such as data range, into this experiment. More participants will also be invited.

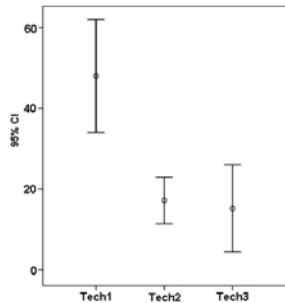


Fig. 8. The response time for three techniques with 95% confidence interval. Tech 1: juxtaposed views with the original windows; Tech 2: juxtaposed views (full view); Tech 3: juxtaposed views (pattern outline).

5 Related Work

In order to deal with large time-series datasets, some abstraction algorithms have been introduced into time-series visualization. These can be categorized into two approaches: user-driven [10,5] and data-driven [11,3]. Hao et al. [10] introduced DOI (degree of interest) functions to determine the sampling rate. The DOI function is used to represent how users are interested in different portions of a time-series dataset. The subset with a higher DOI value is visualized using a higher sampling rate. In another paper by Hao et al. [5], they used variable resolution density displays to visualize univariate data. These user-driven approaches can show more details for important data, but are not applicable to data streams whose requirements normally are close to real-time.

Miksch et al. [11] developed an abstraction algorithm for temporal univariate data that aims to transform numerical values to qualitative descriptions. It can smooth data oscillation near thresholds. *BinX* [3] is a real-time system to visualize time-series data on the fly. It uses an aggregation algorithm to adapt large datasets to a limited canvas and supports online adjustment for the levels of aggregation. Both of these techniques only handle univariate data trends, while our framework is applicable to more complex data patterns.

If using an abstraction algorithm, a distorted timeline might be necessary to give important data more space. This technique is used in many research efforts. For example, Bade et al. designed an intensive care unit monitoring system in which multiple timelines are displayed. Users can select a subrange at the bottom timeline, then rescale the time range and show it in the middle and top timelines [12]. We borrowed the ideas from this paper to represent the unevenly spaced timeline in our work.

6 Conclusions

This paper addresses the problem of how to efficiently visualize pattern changes on a data stream given the fact that the pattern change rate is not constant. Distorting the time axis can partially solve this problem, but most existing techniques are user-driven. This is not applicable to data streams that normally need quick responses. We proposed a data-driven approach to automatically merge adjacent time windows with few or no changes in the current view. Our experiments show that our proposed merge algorithm can preserve more change information than uniform sampling. We proposed two types of visualization techniques: juxtaposed full views and outline views. The former keeps the data details while the latter aims to convey only the data pattern users want to observe. We conducted a user study to confirm that our visualization techniques together with the merge algorithm can significantly reduce the time cost to detect pattern changes over data streams.

References

1. Lukasz, G., Özsu, M.T.: Tamer: Issues in data stream management. *SIGMOD Rec.* 32, 5–14 (2003)
2. Wong, P., Foote, H., Adams, D., Cowley, W., Thomas, J.: Dynamic visualization of transient data streams. In: *Proc. IEEE Symposium on Information Visualization*, pp. 97–104 (2003)

3. Berry, L., Munzner, T.: Binx: Dynamic exploration of time series datasets across aggregation levels. In: IEEE Symp. Information Visualization Poster, pp. 215.2 (2004)
4. Albrecht-Buehler, C., Watson, B., Shamma, D.A.: Visualizing live text streams using motion and temporal pooling. *IEEE Computer Graphics and Applications* 25, 52–59 (2005)
5. Hao, M.C., Keim, D.A., Dayal, U., Oelke, D., Tremblay, C.: Density displays for data stream monitoring. *Comput. Graph. Forum* 27, 895–902 (2008)
6. Chen, Y., Dong, G., Han, J., Wah, B.W., Wang, J.: Multi-dimensional regression analysis of time-series data streams. In: VLDB, pp. 323–334 (2002)
7. Minnesota Department of Transportation: Mn/DOT traveler information (2009), <http://www.dot.state.mn.us/tmc/trafficinfo/> (accessed on February 25, 2009)
8. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* 101, e215–e220 (2000)
9. Xie, Z., Ward, M.O., Rundensteiner, E.A.: Visual analysis of multivariate data streams based on doi functions. Technical Report TR-10-06, Worcester Polytechnic Institute, Computer Science Department (2010)
10. Hao, M.C., Dayal, U., Keim, D.A., Schreck, T.: Multi-resolution techniques for visual exploration of large time-series data. In: EuroVis 2007: Joint Eurographics - IEEE VGTC Symp. on Visualization, pp. 27–34 (2007)
11. Miksch, S., Horn, W., Popow, C., Paky, F.: Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants. *Artificial Intelligence in Medicine* 8, 543–576 (1996)
12. Bade, R., Schlechtweg, S., Miksch, S.: Connecting time-oriented data and information to a coherent interactive visualization. In: CHI, pp. 105–112 (2004)

RibbonView: Interactive Context-Preserving Cutaways of Anatomical Surface Meshes

T. McInerney and P. Crawford

Dept. of Computer Science, Ryerson University, Toronto, ON, Canada, M5B 2K3

Abstract. We present an interactive visualization tool that provides users with the capability of cutting away the surfaces of enclosing objects to reveal interior or occluded objects. This tool, known as *RibbonView*, is applied by mimicking the sweeping action of a paint roller across the surface of a smooth object. This virtual analogy of a familiar real-world action not only removes the occluding surface but also generates a series of polygonal strips or *ribbons* that provide an effective contextual outline view of the removed material. The cutaway tool can be used for many types of cuts, uses a single, consistent interaction style, and the cut region is easily editable. We apply *RibbonView* to several human anatomy data sets to demonstrate its ease of use and effectiveness.

1 Introduction

Visualizing, examining and measuring anatomical structures embedded in 3D medical images are key tasks in radiology. In surgical planning, surgeons must ascertain the complex shape and organization of these structures to find optimal approaches to a target structure or to plan a series of surgical actions. In general, the explosion of large 3D datasets in medicine and other application domains such as architecture, entertainment, engineering and manufacturing, has resulted in the need for effective visual communication styles as well as simple, intuitive interactive data exploration tools which allow users to not only view the data, but comprehend it.

When inspecting and manipulating complex systems of solid 3D objects, it is critical to provide users with the capability of seeing through the surfaces of enclosing objects to reveal interior objects or parts. Users also need to understand and measure the spatial relationship between objects in the system and may need to spatially overlay two or more similar objects in order to compare their shape and size. Providing these capabilities is a challenging interactive 3D visualization problem. Ideally one would like to mimic the real world actions of simply “reaching out” and “cutting away” occluding material in order to examine a hidden region or object sub-part. Furthermore, it is imperative that this region *focusing* capability is not achieved at the expense of a *contextual* view of the surrounding or occluding structures.

While the use of semi-transparency on an outer surface can reveal the interior of an object, the generated images tend to be noisy and do not effectively convey the spatial relationship or depth of interior objects, especially if there are multiple layers of transparency. Simply cutting away parts of the occluding objects is debatably a more effective approach, but the user is forced to mentally “fill in the gap” of the removed material and estimate its shape. Ideally it would be desirable to cut away the outer

surface while still rendering some outline of it that simultaneously minimizes occlusion and provides effective visual depth and spatial relationship cues of the interior. Furthermore, if this outline is aligned with the natural geometry of the outer object surface (e.g. along primary medial axes, ridge lines, primary curvature lines etc.) then the user can more readily mentally reconstruct the missing geometry.

Although these classic focus-plus-context visualization issues are important, allowing users to easily *generate* object-aligned cutaway regions is equally important. Radiologists, surgeons and medical technicians are very knowledgeable about anatomy. However, given their considerable workloads, using the complex and restrictive interfaces too often associated with common medical visualization software is counterproductive. The goal is to have these expert users focus on their specific visualization task and not on how to use the visualization tools. To realize this goal requires the creation of simple but powerful interaction metaphors.

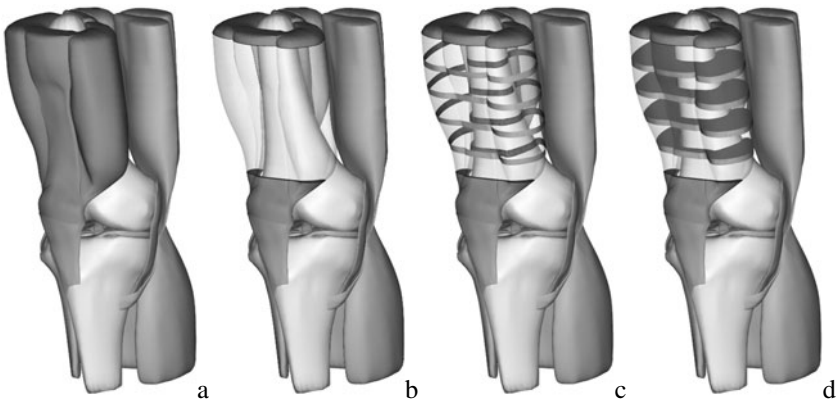


Fig. 1. Knee data set showing: a) surfaces of muscles, tendons, and bone, b) cutter extruded along muscle and resulting cutaway material shaded transparently, c) ribbon view of cutaway material, d) solid “slice” view of cutaway material

In this paper we describe RibbonView - a cutaway visualization tool that mimics the sweeping action of a hand-held paint roller across the surface of a smooth object. In RibbonView, this virtual analogy of a familiar real-world action does “double duty”. Sweeping the cutaway tool along an object surface not only removes the occluding surface but also generates a series of polygonal strips or *ribbons* outlining the removed material (Fig. 1c). The sweeping action is simple and intuitive - a user simply pulls a rectangular strip over the surface of the object along a curving path aligned with the natural shape lines of the object. As the strip is pulled across, it is continuously extruded to form a quadrilateral cutter mesh orthogonal to this path. Thus, the very action of using the cutaway tool, coupled with its explicit quadrilateral mesh-based representation, automatically establishes a contextual outline view of the enclosing object. The cutaway tool can be used for many types of cuts while adhering to a single, consistent interaction model. Furthermore, since the cutter mesh corresponds directly to a cutaway region, cuts are not only easily created, but easily edited as well. In the remainder of this paper, we describe RibbonView and its interaction model in more detail. We

apply it to several human anatomy data sets to demonstrate its effectiveness and provide a discussion of its ease of use and efficiency compared with other techniques.

2 Related Work

Anatomical structures often have complex shapes and systems of these structures have complex spatial interrelationships. Medical volume images containing these structures are commonly viewed through direct volume rendering or by segmenting and extracting the surfaces of the anatomical structures from the volume, generating polygonal surface meshes. Although we concentrate on polygonal meshes in this paper, it is nonetheless insightful to first briefly review cutaway techniques and context-preserving volume rendering algorithms for volume images. We will then review recent cutaway techniques for polygonal meshes.

2.1 Volume Image Cutaway and Rendering

A simple and common technique to reveal hidden structures in medical volume images is the use of one or multiple clip planes, generating image slices. Radiologists are adept at scanning through volume images on a slice by slice basis and mentally reconstructing 3D shape. While the use of 2D slice images obviously does not directly convey 3D shape, slices are useful for exact measurements of depth and/or distance between objects.

Clip planes support only a limited range of cut geometries. Weiskopf et al. [1] extend the range of volume clipping by interactively applying convex and concave clipping objects, carving the away chunks of the volume image. Konrad-Verse et al. [2] present a virtual resection tool based on a deformable cutting mesh. Chen et al. [3] proposes a variety of complex volume manipulation tools that allow drilling, lasering, and peeling operations.

Another approach to viewing interior structures in volume images is through the use of illustrative volume rendering techniques [4], [5], [6]. Essentially these techniques attempt to reduce the opacity in less important regions in order to visualize interior and exterior structures of a focal region while also preserving and enhancing important shape cues. This "context-preserving" volume rendering model is often a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity.

Several research groups have also been exploring the use of deformations to visualize volume images [7], [8], [9]. McGuffin et al. [7] proposed an interactive system to browse pre-labeled iso-surfaces in volume data by deforming them according to simple interaction metaphors. Users can cut into and open up, spread apart, or peel away parts of the volume in real time using widgets, attempting to retain surrounding context. Mensmann et al. [8] also use a deformation approach to simulate surgical cutting of tissue to reveal tissue underneath. Finally, Correa et al. [9] use a space warping technique to simulate peelers, retractors, pliers, and dilators.

2.2 Polygonal Mesh Cutaway

Many segmentation algorithms are commonly used to label the voxels associated with anatomical structures embedded in medical volume images. The surfaces of these

structures can then be extracted; resulting in systems of 3D closed (i.e. "solid") polygonal surface meshes. Another class of segmentation algorithms known as deformable models, can typically directly generate polygonal meshes.

The primary techniques used to examine occluded interior structures, whether through direct volume rendering of the volume images or through the use of extracted polygonal meshes, are cutaway views and ghosted views (i.e. semi-transparency). For example, Diepstraten et al. [10] used transparency for technical illustrations.

The most common technique to implement cutaways of polygonal meshes is Constructive Solid Geometry (CSG). RibbonView uses the GTS library [11] to perform CSG operations as it allows us to maintain explicit mesh representations of both the object and the cutaway piece.

Several researchers have developed cutaway tools for surface meshes. Coffin et al. [12] present an interactive technique to cut holes in occluding geometry using a user-defined cutout shape or a standard shape. Diepstraten et al. [13] presented different methods to create cutaways in polygonal data. In Knodel et al. [14] users generate cutaways using simple sketching actions and then refine the shape of the cut using widgets. Li et al. [15] presented a system for authoring and viewing interactive cutaway illustrations of complex 3D models, targeted towards medical and technical education rather than for radiology or surgical planning.

3 RibbonView

In our cutaway approach we attempt to provide a single intuitive interaction metaphor that can be used for many different types of cuts. We employ a paint roller metaphor (Fig. 2) where the user grabs the "handle" and pulls the "roller" across the surface of the object, *extruding* a quadrilateral cutter mesh to define the cutaway region. This surface-constrained interaction is simple, fast and comfortable.



Fig. 2. A paint roller is used as the underlying interaction metaphor of the RibbonView tool

One advantage of the extrusion interaction model is the user is provided with immediate visual feedback of the cutter mesh – and hence the cutaway region – as it is created. Furthermore, users can stop at any time, back up (un-extrude) or make adjustments to the cutter height and depth (see Sect. 2.3).

However, the primary advantage of the extrusion process is the leading quadrilateral edge (the "roller") of the cutter is constrained to remain orthogonal to the extrusion path defined by the cursor (the "handle") (Fig. 2, right) as the user pulls it along with the mouse. The extrusion path can be easily made to coincide with medial axes of objects or ridge lines, or other curving feature lines. The result is a cutter mesh that

essentially consists of a connected series of quadrilateral polyhedrons (loosely referred to as “boxes”) oriented orthogonally with respect to these shape feature curves. These boxes provide a convenient framework for the construction of the context preserving ribbons.

The ribbon alignment and ribbon shading provide an effective visual outline of the cutaway region, allowing users to more easily mentally reconstruct the original surface while they are simultaneously viewing the interior. In the following sections, we describe the extrusion process and the subsequent construction of our cutter mesh. We then describe the various types of cuts that can be performed as well as the cutaway region rendering styles supported. Finally, we describe the editing operations that can be performed on the cutter mesh.

3.1 Cutter Model Representation and Construction

The cutter mesh is represented as a 3D closed mesh of quadrilaterals (Fig. 3d). It can be visualized as a connected series of polyhedrons (“boxes”) (Fig. 6a). The user also has the option of using the cutter as a control mesh for an interpolating subdivision surface [16] (Fig. 3e). The subdivision surface is used when performing freeform cuts (Fig. 3h) or rounded cuts. Maintaining an explicit mesh representation of the cutter as well as of the cutaway region allows us to maintain complete control over the rendering styles and also allows for subsequent manipulation and measurements (Sect. 3.3).

To create a cutter – and hence define the cutaway region – the user first establishes the height of the “roller” by clicking on a surface point of the object¹ and stretching out an initial narrow and thin rectangular box (Fig. 3a). The user can slide the free end of the initial box around (Fig. 3a–b) to fine tune the roller height and orientation.

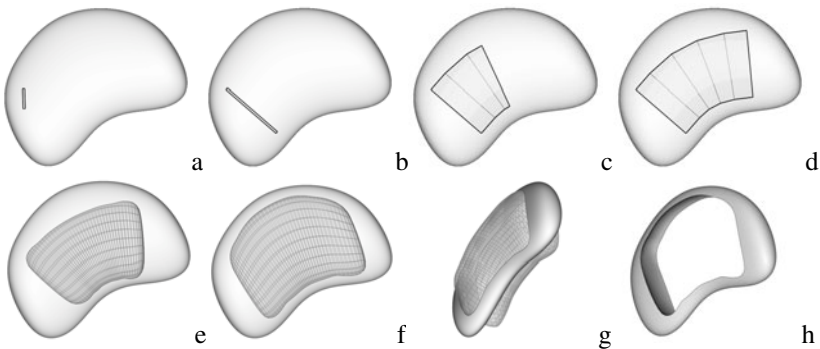


Fig. 3. To create a cutter mesh, the user extends and orients an initial thin rectangular box “strip” (a–b) and then extrudes it by pulling the right face across the object surface (c–d). New boxes are automatically created and connected to the cutter mesh. The mesh is thickened and can be subdivided and edited, if desired, resulting in a freeform shape cut (e–h).

¹ The user can also click on a point off of the object to create a cutter that completely surrounds a region of the object.

Once the initial cutter box is established, the user pulls on the right-hand face and smoothly widens it (Fig. 3c). When the box reaches a user-defined width (set via a slider), a new box is created and connected to the cutter mesh (Fig. 3c–d). As the new box is pulled and widened across the surface of the object, the front face of the box may “collide” and intersect with the object surface. This collision is detected and the front face is rotated around its shared front edge until it no longer collides. In effect, this action causes the front face to be constrained to “stick” to the object surface. This widening, rotation and box generation process happens quickly and smoothly - the user is unaware of the underlying collision detection and response algorithm and sees only the continuous cutter extrusion.

Once the cutaway region has been defined with the cutter mesh, the mouse scroll wheel is used to “thicken” the cutter to the desired depth (Fig. 3g). The cut is then activated with a key press (Fig. 3h). Since parts of the object may be thicker than others, a small amount of unwanted object material may remain after the cut is made. The user can easily edit the depth of the cutter mesh in the thicker region and re-cut.

3.2 Cutaway Types Supported

Various cuts are naturally supported by the combination of the extrusion process, the quadrilateral cutter mesh and the subdivision process. Each cut is described below:

Curving Path: The default cutaway supported is a curving path cut. As described previously, the curving path cut is used to cut away material along the natural shape lines of an object. The cut can either wrap around a section of the object (Fig. 1), cut away part of the object (Fig. 4a, b), or cut a “window” in the object (Fig. 3h).

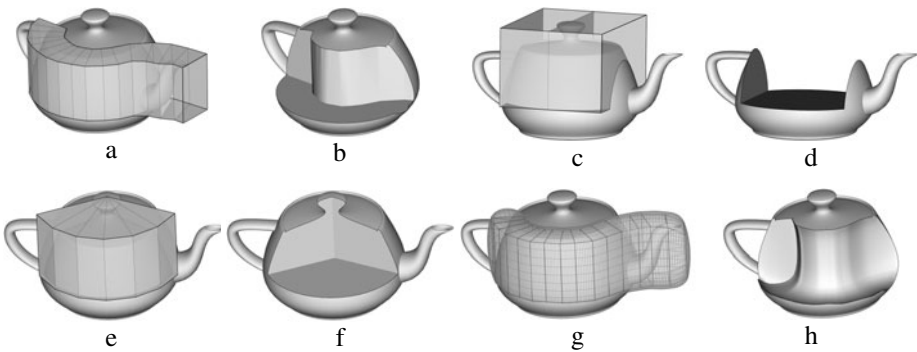


Fig. 4. Cut types supported: (a,b) curving path cut, (c,d) straight path cut (“box” cut), (e,f) surface of revolution cut (“wedge” cut), (g,h) freeform cut

Straight Path: With the click of a button, the surface-sticky curving cutter extrusion path can be constrained to continue along a 3D straight line. This simple constraint supports the creation of standard volumetric cuts such as box cuts (Fig. 4c,d). The cutter can also switch at any time between curving paths and straight line paths.

Wedge Cut: Many anatomical structures are disc-shaped or tubular, with a well-defined single primary medial axis. By simply clicking and constraining one end of

the initial cutter strip, surface of revolution or “wedge” cutters can be extruded to open up these structures (Fig. 4e,f). The free end of the cutter strip can be swept over an arbitrary range of angles, from narrow wedges to complete surfaces of revolution.

Freeform Cut: Freeform cuts are performed using a subdivision surface cutter (Fig. 3h). Rather than using a tedious and error-prone tracing process, freeform cuts are performed by first extruding a rough cutter shape across the target region of an object surface as usual (Fig. 3d). With the click of a button, the cutter mesh is then used as a control mesh and automatically subdivided with a tightly-interpolating subdivision process. The control points of the mesh can then intuitively pulled, in a constrained manner, to quickly refine the freeform shape (Fig. 3f).

Curved or “Rounded” Cut: The subdivision surface can also be used to generate cutter meshes that are rounded (Fig. 4g, h). As the depth of the cutter can be locally controlled along its length, rounded cuts can be generated such that an interior object is fully exposed without cutting all the way through the exterior object.

3.3 Context-Preserving Rendering Styles of the Cutaway Region

The extrusion/sweeping interaction model combined with the explicit quadrilateral mesh representation of the cutter and the use of GTS, provide the means for very flexible, context-preserving rendering styles. For example, since GTS outputs an explicit mesh representation of the cutaway region, we can render the cutaway region as semi-transparent (Fig. 5c). In addition, GTS partitions the cutaway region into several components, including the part of the object contained within the cutter mesh and vice versa. This flexible scheme allows us to render each of these components separately or in combination. For example, the part of the object contained within the cutter mesh – the “walls” – can be shaded and outlined in a different color (Fig. 5a).

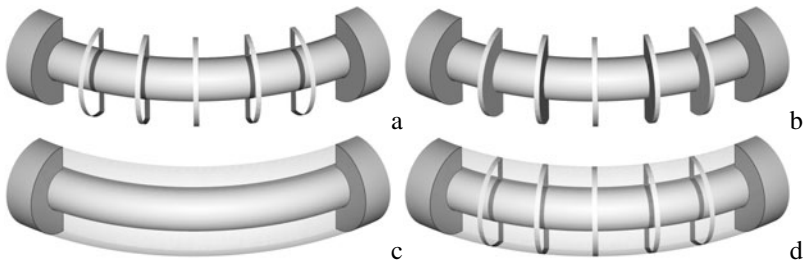


Fig. 5. The extrusion/sweeping interaction model combined with the explicit quadrilateral mesh representation of the cutter and the use of GTS provides a convenient framework upon which to construct different context-preserving rendering styles

The quadrilateral parameterization of the cutter mesh allows us to construct a series of ribbons. Ribbons are constructed by interpolating cross-sectional “slices” between the cutter mesh boxes and subsequently constructing thin, quadrilateral box meshes centered on each slice. The width of the box mesh can be varied to control ribbon width. The thin box meshes and the object are then fed to GTS, causing only the part of the object contained within each of these boxes to be output. This has the effect of

removing only ribbon-shaped sections of the geometry. These ribbons can then be rendered alone (Fig. 5a) or combined with a transparent intersection surface (Fig. 5d). Furthermore, the explicit mesh representation of the ribbons provides a means for separately rendering the outside and inside of the ribbon mesh (Fig. 5a).

Finally, we are also able to generate solid cross-sectional slices (Fig. 5b). Solid slices are constructed in a manner similar to ribbons except that the complete intersection geometry is computed by GTS, rather than just the outermost surface.

In summary, maintaining explicit representations of both the input and output of the CSG operations, combined with the parameterization of these representations, provides a framework upon which effective visual communication styles can be built. We also plan to explore non-photorealistic rendering styles for additional visual cues.

3.4 Cutter Mesh Editing

As described in section 3.1, the cutter is a closed quadrilateral mesh and can be loosely thought of as a series of joined boxes (Fig. 6a). This representation supports constrained editing of the cut region, preventing the user from creating ill-defined cutters while still providing cutter shape flexibility. For example, to locally modify the height of the cutter, the user selects a cutter cross-sectional “slice”, corresponding to a box inner boundary face (or end-cap face), and drags the top edge of the slice to extend, shrink, or reposition it. During this interaction the slice rectangle is constrained such that it cannot intersect a neighboring slice rectangle, while at the same time always remaining planar.

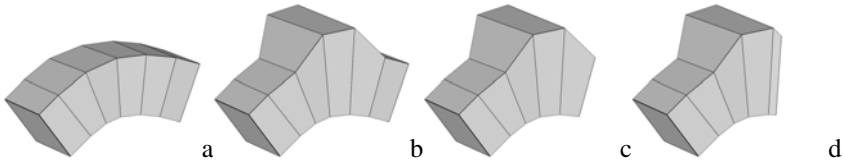


Fig. 6. The shape of the cutter mesh (a) can be intuitively edited in a constrained fashion to prevent the creation of degenerate cutters. In (b) the height and depth have been locally modified. In (c–d), the cutter has been un-extruded.

Similarly, the user can use the mouse scroll wheel and quickly “thicken” the entire cutter mesh, cutting more deeply into, or entirely through, the object. The cross-sectional slices of the cutter may be angled with respect to each other (if, for example, the cutter was extruded along a sharply curving path). As the cutter is thickened these faces may intersect, resulting in a non-simple closed cutter mesh. An iterative procedural algorithm is continually invoked as the cutter is thickened to prevent cutter self-intersection. This algorithm detects intersecting slice rectangles and rotates them slightly away from each other, using the front edge of each slice rectangle as its rotation axis. In addition to global cutter thickening, the user is also able to select a cutter slice and thicken locally (Fig. 6b). Slice orientation and position may also be adjusted.

A critical factor of any cutaway tool is the ability to quickly and easily undo input actions. The RibbonView tool supports a simple but effective un-extrude action. The user simply presses a key and then un-extrudes the cutter by pulling the leading edge of the cutter in the opposite direction. The orientation and width of the current box

can be adjusted or the entire cutter can be undone, box by box. A quick undo option uses the Ctrl-z keys to remove the last box of the cutter.

4 Experimental Results and Discussion

We have performed a series of experiments to demonstrate the effectiveness and ease of use of the RibbonView tool. All region selection operations (initialization, extrusion and thickening) were performed in real-time and most take only a few seconds to generate (Table 1). Fine tuning the region of interest adds a few more seconds. Once the region has been selected (and edited if desired), a key press activates the actual cut. Currently the GTS-supported cutaway operations are not GPU accelerated. However, a spatial data structure is used to speed up the cutting. Only a few seconds are required to perform complex cuts on an object with up to 500000 triangles.

In the first experiment we cutaway a portion of a muscle to expose the femur bone (Fig. 1). Unlike the use of transparency alone (Fig. 1b), note how the ribbons clearly show the wrapping of the muscle around the thigh bone (Fig. 1c). In the second experiment we perform a freeform cut of the skin to expose the skull (Fig. 7a). We then perform another freeform cut to cutaway part of the skull. This type of cut is used for planning some cranio-facial surgical procedures. In the third experiment we have extruded the cutter along the primary medial axis of the jaw and combined ribbons with transparency to show the shape of the mandible (Fig. 7b). Solid slices can also be generated and used to compare and measure the shape of this jaw with another jaw data set or with the same post-surgical jaw. In the final experiment we create a wedge cut of the vertebra to reveal the disc underneath (Fig. 7c).

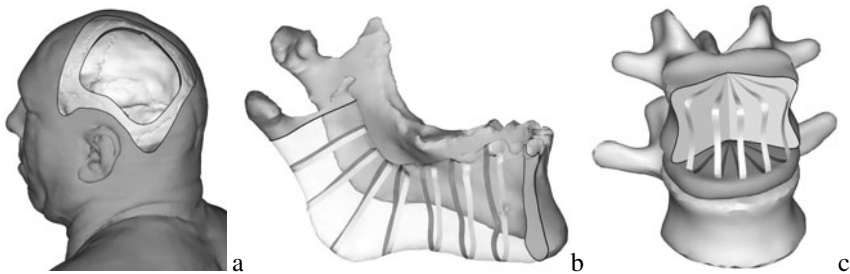


Fig. 7. Results of different cuts using the RibbonView tool. In a) a freeform cut of the skin and skull has been performed. In b) a curving cut along the primary medial axis of the jaw is demonstrated. In c) a wedge cut exposes the relationship of the vertebra to the disc underneath.

We also performed an informal user study on the ribbons and solid slices versus transparency. The study asked users to choose the visualization style that best illustrated the shape of the cutaway occluding surface, as well as the spatial interrelationship between the occluding and occluded surfaces. Several sets of images were presented to approximately 40 participants. Each set contained the original uncut image of a unique system of anatomical structures, and three cutaway images – each using one of transparency, ribbons, or solid slices. Overall, ribbons or solid slices were preferred on average 80 percent of the time.

Table 1. User interaction times required to generate results in Figures 1 and 7

Fig.	Extrusion (sec)	Edit Time (sec)	No. Edits	Total Time (sec)
1	5	6	2	11
7-a	10	17	10	27
7-b	8	8	4	17
7-c	4	3	1	7

4.1 Discussion

In many cut away algorithms, the ability to easily, efficiently and effectively construct, define and modify the cut region to obtain the optimal view of the interior is often inadequate. Our contention is that this inability is due to the choice of interaction model. For example, cutting- or tracing-based interaction models attempt to mimic the real world action of scissors (or a marker) to cut out (or outline) the region of interest. While this type of interaction may be useful for freeform cutout shapes, it is difficult to perform more structured, symmetric or volume-based cuts. Editing or modifying the cut region is also problematic. Backtracking can be performed or it may be possible to convert the traced region to a spline curve for further editing. Tracing with a mouse can also be tedious – pen based systems or touch screens may be more efficient but there is the new problem of occlusion by the user’s hand.

Sketch- or gesture-based interaction is another metaphor that has been used in cut-away systems. In this scenario the user draws different curve shapes on the object surface and each curve is then interpreted by the algorithm to generate different cuts. For example, Knödel et al. [14] define four different curve gestures: line, circle, corner, and ribbon. One of the basic problems with this interaction metaphor is that users can't immediately see the shape of the cutaway region they are creating. Another problem is after the initial cutting phase, the modification of a cutaway region is typically performed using a widget that is constructed from the sketched input curve. The implication is that users are now faced with two interaction models. Furthermore, widgets typically have rather restrictive editing capabilities.

Sculpting based interaction is another popular interaction model [1], [3]. Sculpting may be better suited to the shaping (i.e. generation) of objects, for example by digital artists, rather than the visualization of existing structures. The difficulty aligning or positioning the sculpting tool along natural shape lines of the object and the subsequent inability to easily or efficiently edit the cutaway region may limit the effectiveness of this interaction model. For example, it may be tedious to interactively explore/remove a region of the data around a target anatomical structure using unconstrained 3D positioning, and each cutting action is independent of the others.

Finally, widget based interaction models [7] are often limited to cutting away simple convex shapes and it may be difficult to produce elongated, curving cuts that are aligned with the natural shape of the object. Furthermore, several widget types are often provided and the user is forced to shift their focus away from the visualization task to decide which widget is most appropriate – an often nontrivial task.

The RibbonView interaction model is an attempt to design a model that minimized the deficiencies noted above. In our opinion, the two most critical design factors are the ability to easily position the cutter along natural shape features of the object using simple constrained input actions, and the ability to easily modify/fine tune the cut.

5 Conclusion

Using either semi-transparency or cutaways separately to examine occluded interior structures often results in an ineffective visual communication style. Some synthesis of the two techniques is perhaps necessary in order to understand the spatial interrelationships of a complex system of anatomical structures. In this paper, we have utilized ribbons to achieve this synthesis. The shaded ribbons coupled with a semi-transparent cutaway region not only maintains a contextual view but also clearly shows the spatial relationship of the back of the occluding surface and the front of the interior surface.

Furthermore the simple, fast and intuitive generation of a desired data view is equally as important as the visual communication style itself. In this paper, we have adhered to a single, consistent and intuitive extrusion interaction model and realized this interaction model by sliding, expanding, creating, and joining boxes along the surface of an occluding object. This design allows us to efficiently generate a cutaway that is aligned with the natural shape features of an occluding object, with most cutaway operations completed within 10 – 15 seconds (Table 1).

We are currently exploring several improvements and extensions to the RibbonView system. For example, we are experimenting with the use of a multi-touch input device for RibbonView such that the user can literally reach out with their hands and sweep their fingers across the object surface to indicate the cutaway region. We also plan to transfer the cutting operation to the GPU to provide real-time cutting.

References

1. Weiskopf, D., Engel, K., Ertl, T.: Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics* 9(3), 298–312 (2003)
2. Konrad-Verse, F.O., Preim, B., Littmann, A.: Virtual Resection with a Deformable Cutting Plane. In: *Proceedings of Simulation und Visualisierung*, pp. 203–214 (2004)
3. Chen, H.L.J., Samavati, F.F., Costa Sousa, M.: GPU-based Point Radiation for Interactive Volume Sculpting and Segmentation. *The Visual Computer* 24(7-9), 689–698 (2008)
4. Zhou, J., Döring, A., Tönnies, K.D.: Distance based enhancement for focal region based volume rendering. In: *Proceedings of Bildverarbeitung für die Medizin 2004*, pp. 199–203 (2004)
5. Viola, I., Kanitsar, A., Gröllner, M.E.: Importance-driven feature enhancement in volume visualization. *IEEE Trans. on Visualization and Comp. Graphics* 11(4), 408–418 (2005)
6. Bruckner, S., Grimm, S., Kanitsar, A., Gröllner, M.E.: Illustrative context-preserving exploration of volume data. *IEEE Trans. on Vis. and Comp. Graphics* 12(6) (2006)
7. McGuffin, M.J., Tancau, R., Balakrishnan, R.: Using Deformations for Browsing Volumetric Data. In: *Proceedings of IEEE Visualization, Seattle, Wash.*, pp. 401–408 (2003)
8. Mensmann, J., Ropinski, T., Hinrichs, K.H.: Interactive Cutting Operations for Generating Anatomical Illustrations from Volumetric Data Sets. In: *Journal of WSCG – 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, vol. 16(1-3), pp. 89–96 (2008)
9. Correa, C., Silver, D., Chen, M.: Feature Aligned Volume Manipulation for Illustration and Visualization. *IEEE Trans. on Vis. and Computer Graphics* 12(5), 1069–1076 (2006)

10. Diepstraten, J., Weiskopf, D., Ertl, T.: Transparency in Interactive Technical Illustrations. *Computer Graphics Forum* 21, 317–326 (2002)
11. SourceForge, The GNU Triangulated Surface Library (2006), <http://gts.sourceforge.net>
12. Coffin, C., Höllerer, T.: Interactive Perspective Cut-away views for general 3D scenes. In: *Proc. of IEEE Symposium on 3D User Interfaces*, Alexandria, VA, pp. 25–28 (2006)
13. Diepstraten, J., Weiskopf, D., Ertl, T.: Interactive Cutaway Illustrations. *Computer Graphics Forum* 21, 22(3), 523–532 (2003)
14. Knödel, S., Hachet, M., Guitton, P.: Interactive Generation and Modification of Cutaway Illustrations for Polygonal Models. In: *Proc. Of the 10th Int. Symposium on Smart Graphics*, Salamanca, Spain, pp. 140–151 (2009)
15. Li, W., Ritter, L., Agrawala, M., Curless, B., Salesin, D.: Interactive Cutaway Illustrations of Complex 3D Models. In: *Proc. of Int. Conf. on Computer Graphics and Interactive Techniques ACM Siggraph*, San Diego, CA (2007)
16. Schaefer, S., Warren, J.: A Factored Interpolatory Subdivision Scheme for Quadrilateral Surfaces. In: *Curve and Surface Fitting: Saint Melo 2002*, pp. 373–382 (2003)

Interactive Visualisation of Time-Based Vital Signs

Rhys Tague, Anthony Maeder, and Quang Vinh Nguyen

School of Computing and Mathematics, University of Western Sydney, Australia
{R.Tague, A.Maeder, Q.Nguyen}@uws.edu.au

Abstract. Vital signs interpretation is an important element of patient monitoring. The use of visualisation could enhance this interpretation. We present a model for time-based visual analytics visualisation of vital signs. Our model allows for multiple vital signs to be charted along a time-based axis. The patient management care team then can apply lenses to alter the view being presented of the data, allowing the viewer to understand the meaning and improve the interpretation. The lenses applied will allow for generalisation, refinement, and traversing.

1 Introduction

Measurements of physiological factors play an important role in health care. Basic vital signs such as heart rate, blood pressure, temperature and respiratory rate, are frequently the focus of these measurements. These factors can be measured easily, reliably and repetitively with relatively inexpensive and simple medical devices. Their successive measurement over time enables various clinical processes and decisions to be undertaken with a higher degree of confidence as characteristic patterns or significant changes are detected. These processes may include such common tasks as function or condition assessments, patient health status monitoring, and effect of or response to treatments. The examination of vital signs would help the carers to discover patterns and irregularities as well as make predictions. As a result, many such patients can be better managed by the collection and analysis of vital signs readings [1].

The task of managing data for records of vital signs over time in a “patient data history” can be achieved with standard database techniques, provided adherence to appropriate standards for the data collection, representation and communication have been observed. Interpretation and understanding of the resulting information can be very complex, yet there is a lack of standards for this aspect. This situation is worsened by the complexity of interrelatedness for such data, even for basic vital signs. For instance, the clinician may need to determine whether a particular combination of heart rate and blood pressure changes are anomalous, relative to “typical” human characteristics, or relative to past performance of the patient over a given time period. This question would currently require a significant amount of numerical analysis and multiple comparisons of data sets to be performed. Furthermore, presentation of the results in a form which is readily intelligible and assimilable by a wide range of different clinical workers, from specialist clinicians to supporting health carers, is challenging in that it requires different views of the same data to be presented appropriately.

Much current research on vital signs analysis focuses on the data mining aspect in their analysis such as those in [2-5]. Although these are capable tools for analysis, the lack of an interactive visualisation could reduce their effectiveness. The authors believe that one of the best approaches to tackling the above limitations is to utilise both human and computer strengths with a mixed-initiative visualisation where both can collaborate in the exploration and creation of knowledge. The rich and adaptive visualisation environment can help the human analyst to gain knowledge of the data through our powerful visual perception and reasoning skills, ideally driving the system toward more focused and more adequate analytical techniques. The visualisation should provide not only a simplified abstract view of the entire record of the patient data history, but also display in detail the information at a particular focus point.

These limitations can be addressed by providing a rich and adaptive visualisation environment for working with data of this type. The fundamental components of this visualisation are (i) representing portions of time-based vital signs data sets for selected time points or periods of interest, and (ii) annotation and cueing of anomalous or critical configurations for both single and multiple data items. In each case, the mode of visualisation adopted should be meaningful to the level of sophistication of different user types, yet consistent through these levels of simplification (in a similar way to geographical map presentation at different scales). This paper proposes a new design approach to model such a visualisation and analysis aid.

2 Conventional Visualisation and Visual Analytic Approaches

The technology now exists to capture and store physiological vital signs in patient data histories. However, vital signs signals are very complex because of the interdependence between parameters, and irregularities in signal variation over time. Examination of the data through the generalised visualisation of patient data histories rather than simply detailed display of all the signal data, could provide better insight for the patient condition, and in turn support better diagnosis and treatment for the patient. Although much effort has been applied to collect, standardise and manage vital signs [6,7] for further analysis, this data is commonly presented as a text-based document, spreadsheet table, database table or text flow chart. Better tools and models are needed to provide more effective visualisations and analysis of such multivariate time series data and related analysis.

Visualisations have been successfully used to enhance the navigation and analysis of personal health histories [8, 9]. These techniques provide an interactive visualisation platform for clinical patient records in which health problems, diagnoses, test results and medications are represented as dots and lines. These small items can be zoomed to show more detail of the information. Attribute properties, such as line colour and line thickness, are also employed to illustrate the relationships and significance. Bui et al presented a *TimeLine* system [10] that provides a temporal visualisation for medical records, in which the data can be reorganised around medical disease entities and conditions. Zheng et al. investigates user interactions with electronic

health records (EHRs) to uncover hidden navigational patterns in the data sets [11]. Another interesting visual user interface for medical data is *HotBox* [12] which links 3D anatomy and knowledge bases of the anatomy physiological response and other medical informatics resources.

Although these above visualisation techniques and systems have been introduced, there is little research work on the creation and validation of new effective models and techniques to satisfy the current needs for interactive visualisation of vital signs in entire patient data histories, which arise from health systems reforms such as new models of care and expanded care teams. Conventional software solutions usually display each individual vital sign as static line graphs [13,14] which makes it really difficult for human viewers to compare multiple vital signs through the extensive and very detailed historical records.

The most closely related works to this research are those techniques described in [15] and [16]. Fonseca et al. [15] presents a *multivariate time series* representation. The technique is aimed at providing a set of desired state for patients for comparison purposes. Data processing is also employed to enhance the analysis, such as to categorise the parameter and apply linear interpolation methods across the missing values in the data. Its visualisation technique uses Star-Plots in an attempt to display concurrently vital signs. Unfortunately, the use of Star-Plots can create an overlapping problem as multiple lines are drawn along the circular axes. Andry et al. [16] present a web-based *highly interactive graphics environment* that provides an easy and pleasant way for viewers to browse through patient historical data. This technique cleverly applies design principles that present a selected vital sign or multiple vital signs as multiple graphs. The data is also separated into two groups for easy management and analysis, based on health targets (such as heart rate, blood pressure, temperature and respiratory rate) and activities (such as exercise, diet and medication). Unfortunately, the authors do not provide an interactive visualisation implementation with a concurrent view of all vital signs in the patient data histories. The iteration of its development with real clinicians is not considered, and thus the effectiveness cannot be justified through continuous feedback.

3 Proposed Visual Analytics Model

For ease of adoption and validation, ideally new and existing methods for visual analytics should be used in conjunction with one another. This conjunction allows the visualised data to express maximum meaning, as this meaning varies depending on the vital signs being visualised and the specific choice of visualisation technique. The visual analytics will no longer provide static displays of the vital signs, but robust platforms for discovery, data manipulation and decision making. Figure 1 shows our proposed model for knowledge discovery using intelligent visualisations and automated analysis. The new method we propose here uses the following components in conjunction: data analysis, visual cues, basic decision making, and user interaction to analyse the data.

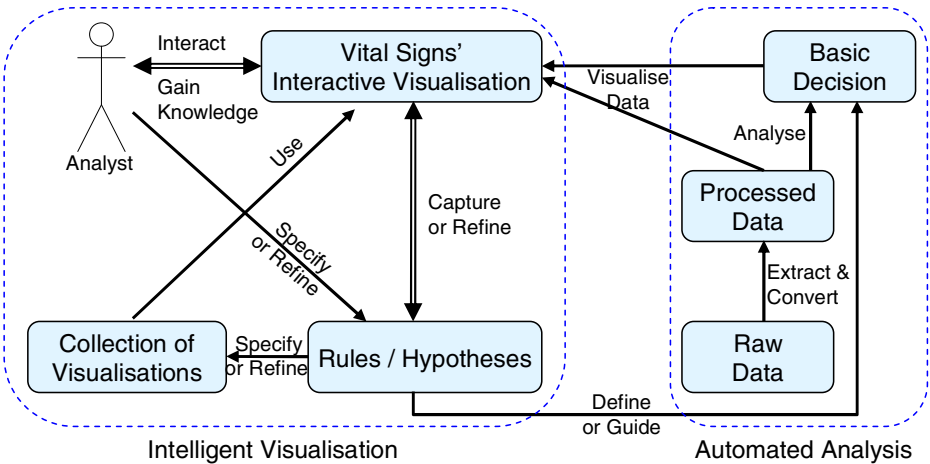


Fig. 1. A Model for Visual Analytics of Vital Signs

1. *Data Processing*: temporal vital signs data can accumulate to an extremely large quantity, especially if recordings from a patient are taken every day over a time-span of many years. Depending on the device which took the recording and the software managing the EHR, the data is often stored in the form of an archetype [17]. Archetypes need to be known and interpreted by the visualisation to extract the raw data. These areas of interest typically arise when a vital sign recording contains an anomaly from the normal [18]. The way in which this anomaly is determined is usually through *control limits* or ranges that have been set by a physician [18]. These control limits are physiological limits that are different in every patient, which necessitates physician involvement to determine them. The limits can also change depending on the context of the patient’s physical state at the time of recording. Each vital sign data series therefore has different control limits which need to be determined before the visualisation can be rendered.
2. *Basic decision making* is supported by automated data analysis techniques, such as data mining, data management and data filtering. These techniques are used to discover new knowledge, filter out irrelevant data, capture important attributes of the information, find relational structures, and/or decompose large structures into sub-structures. Basic decision making provides initial and automated findings that can be verified by medical staff.
3. *Rules and hypotheses* are used to produce new visualisation instances and also to support basic decision making, which can be learned automatically or manually through data exploration based on the user’s behaviour and preferences. The capture rules and the hypotheses can be recorded using a visualisation language. As a result, they can be portable amongst analysts (eg. medical staff) for use with the same application or different data sets.
4. *Collection of visualisation algorithms*: computer-generated layouts play an important role in the interactive visualisation because they can produce the layout of a large data set within a limited amount of time. It could take days or more for

a human to produce a similar readable display. The collection of layout and navigation algorithms would effectively enable the system to provide a suitable output corresponding to the nature of the data sets and user preferences. The list of algorithms can be updated and refined through interaction.

5. *Interactive visualisation* uses the collection of layout and navigation algorithms in the constraint of rules and hypotheses. The interactive visualisation provides the most suitable display for a data set and an analyst. In other words, it should interactively create layouts and interactions based on the nature of the applications, the data sets, the analyst's preferences and the underlying sense-making rules. The innovation also lies in the creation of an effective mechanism to integrate different interactive visualisation techniques so that it can provide a best-possible display corresponding to the analyst's preferences and behaviour.

4 Vital Signs Visualisation

The graphical representation of the vital sign series is important to the overall comprehension of the visualisation. The longitudinal nature of vital signs data requires the representation of the data to be adaptive enough to accommodate the large amounts of data. This can be done through the use of different 'lenses' in the visualisation, such as *generalisation* and *refinement*. Generalisation will allow the user to see the entire patient's vital sign history, and refinement will allow the user to see shorter time periods (if wanting to see days or weeks). These different lenses accommodate the different user types that will use the visualisation. Although different lenses will provide different views of the data to enhance meaning, only basic graphical elements will be used to represent the data. Horizontally-aligned lines will be used when data points are spread over a period of time, which helps the user to navigate with their eye [19], and plotted points will be used when a data point is presented in a moment in time.

As a vital sign data trace can change rapidly over a period of time, the rate of change needs to be represented in the visualisation. This representation has to be done consistently and intuitively to avoid incorrect comprehension [20] which can arise from the user believing that one rate of change is similar to another. We adapt a "traffic light" notification system to allow the user to quickly understand various notification states which exist in the data: *green being "neutral"*, *amber being "area of interest"* (warranting some warning), and *red being "area of concern"* (warranting some intervention). For the notification system to work there has to be decision making associated with the visualisation. This decision making could be rule-based and basic (see *Table 1*) as it is merely to aid in the decision process. These decisions will be made using the raw data after it has been extracted, and then it will be annotated to mark notification areas before it is rendered in the visualisation.

Interaction is an important part of this visualisation, as it allows the user to traverse and manipulate the visualisation to gain a better comprehension. This interaction will cater for all user types that will interact with the visualisation. The users will be able to interact with the visualisation through standard user interface components and computer input devices, e.g. computer mouse. Every time a user performs an interaction with the visualisation, the visualisation rendering will be updated to accommodate the changes

Table 1. Basic Rules applied to data

	1 Std Dev	2 Std Dev	1Std Dev	1 Std Dev
Vital Sign (1)	X	x	x	x
Vital Sign (2)			x	x
Vital Sign (n)				x
	Warning	Concern	Concern	Concern

made by the user. These changes could be in the form of generalisation, refinement, time traversing, and annotating. These interactions allow every user type to look at the data with different lenses, which could help them comprehend the data more.

5 Vital Signs Visualisation

Currently a design has been developed for a prototype version of the new visualisation system described above. *Fig 2* shows the concept of the visualisation that can be interacted with. Currently there are two lenses present at all times: the left lens shows a time span and the right lens shows a moment in time. These two lenses will be interacted with to inspect the data. The time span lens will also allow generalisation and refinement, so more experienced users can look at data points over longer or shorter periods of time. The moment in time lens currently shows data points for the current moment in time. This lens will also have the capability of ‘onion rings’ to show the data points changing over time. The further in time the data point is, the more transparent the data point appears.

Interaction is an important part to this visualisation. All lenses have interaction capabilities for the user to use. The time span lens will allow the users to select segments of time to zoom in on for refinement, and it will also allow zoom out for generalisation. The moment in time lens will allow users to select data points visually presented to go to that moment in time. Both lenses will operate in synchrony, meaning

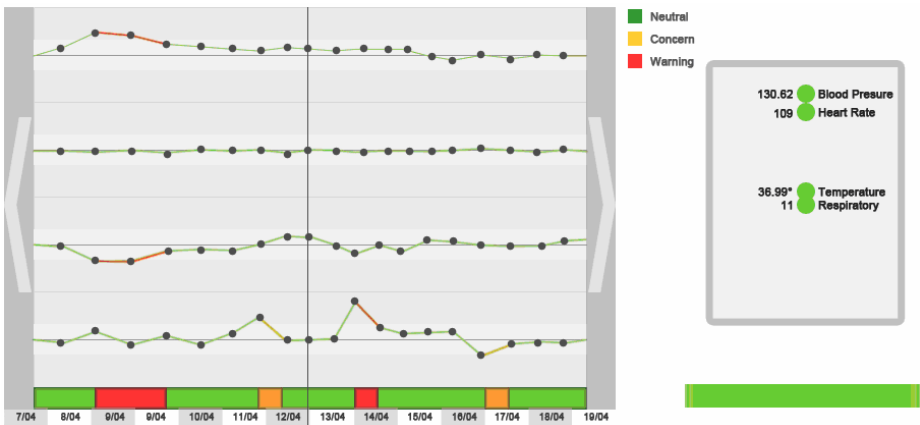


Fig. 2. Concept of Visualisation

one lens cannot change the data index without showing change in the other. As the visualisation is interacted with, the rule based decisions are constantly performed, to show anomalies if the data is manipulated in any way. This continual decision making will allow for continual notification to the user as he/she navigates the data.

The concept graphically represents more than one vital sign. This allows an overview look upon the patient's health status. This also allows for the view of relationships between the vital signs. These relationships are determined by the basic rule based decisions, which could possible help in the decision making process regarding the patients health. To apply the above visualisation model we can take advantage of technologies that enable the visualisation to be used inside an internet browser. Having the visualisation in an internet browser will not only allow the largest reach for potential users, but there will also be no need to install the visualisation onto an individual's computer as it can be rendered remotely and displayed locally. Having remote visualisation could also facilitate tele-monitoring, which could help in streamlining the overall patient monitoring process by spreading this task across the various care team members.

The technology that is being used to develop the prototype implementation is the Adobe® Flash® platform. This platform is widely available on desktop computers and mobile devices. Therefore if the visualisation needs to be rendered on different devices, we have the capability of doing this via the platform software. The data that will be used will be from a vital sign database obtained locally from our collaborating medical researchers. Initially this data will be retrieved by the visualisation system locally and then will be visualised locally. In the future the visualisation will be able to be accessed over the internet, to be rendered in any web-browser.

Once the visualisation is completed it will be validated with potential users. There will be 3 groups of different types of users, with a sample size of 10 users per group. Each user will use the visualisation individually, so there are no cross influence. The 3 groups will be: (1) medical staff, (2) family and friends, (3) chronic disease sufferers. The participants of the study will be interviewed on their understanding of the vital signs with and without the visualisation. The measure for the level of understanding is the accuracy of the participant's interpretation of the data in terms of locating and classifying situations of interest. The questions will be the same for both the raw data and visualised data. Questionnaires will also be after the user has interpreted the data, to allow analysis of the experience that the user had during the visualisation process.

6 Conclusion

This paper has discussed a visual analytics approach in telehealth situations where a large amount of complex vital signs data needs to be understood by a range of different user types. It has been proposed that this need is met by using the two major attributes of the approach, namely the use of generalisation and refinement lenses, and the use of visual cues for data of interest. The expected results from this study will show the extent to which this approach to interactive data visualisation of large amounts of longitudinal vital signs data will improve the overall interpretation of patient data histories by all types of carers, for patients undergoing health care

management using vital signs monitoring. It will also be expected that the use of basic rule-based decision making for notifications will aid in the user interpretation of the vital signs, which is in turn expected to improve the overall management of patient monitoring as a health outcome.

References

1. Harries, A.D., Zachariah, R., Kapur, A., Jahn, A., Enarson, D.A.: The Vital Signs of Chronic Disease Management. *Transactions of the Royal Society of Tropical Medicine and Hygiene* 103(6), 537–540 (2009)
2. Caraca-Valente, J.P., Lopez-Chiavarrias, I.: Discovering Similar Patterns in Time Series. In: *Proceedings of 6th ACM SIGMOD International Conference on Knowledge Discovery and Data Mining 2000*, pp. 126–133 (2000)
3. Seely, A., Macklem, P.: Complex Systems and the Technology of Variability Analysis. *Critical Care* 8, 367–384 (2004)
4. Saeed, M., Mark, R.: A Novel Method for the Efficient Retrieval of Similar Multi parameter physiologic time series using wavelet-based symbolic representations. In: *Proceedings of AMIA Symposium*, pp. 679–683 (2006)
5. Sorani, M.D., Hemphill III, J.C., Morabito, D., Rosenthal, G., Manley, G.T.: New Approaches to Physiological Informatics in Neurocritical Care. *Neurocritical Care* 6, 1–8 (2007)
6. McDonald, C.J., Tierney, W.M.: Computer-stored Medical Records. Their Future Role in Medical Practice. *JAMA* 259(23), 3334–3440 (1998)
7. HL7 Organisation (2010), <http://www.hl7.org> (access July 18, 2010)
8. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B.: LifeLines: Using Visualization to Enhance Navigation and Analysis of Personal Records. In: *Proc. of AMIA Symposium*, pp. 76–80 (1998)
9. Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B.: LifeLines: Visualizing Personal Histories. In: *Proc. of CHI 1996*, pp. 221–227. ACM, Vancouver (1996)
10. Bui, A.A.T., Aberle, D.R., Kangarloo, H.: Timeline: Visualizing Integrated Patient Records. *IEEE Transactions on Information Technology in Biomedicine* 11(4), 462–473 (2007)
11. Zheng, K., Padman, R., Johnson, M., Diamond, H.: An Interface-driven Analysis of User Interactions with an Electronic Health Records System. *Journal of the American Medical Informatics Association* 16(2), 228–237 (2009)
12. Ward, R.C., Pouchard, L.C., Beckerman, B.G., Dickson, S.P.: The HotBox: A Visual User Interface to Medical Data. *Information Visualization* 5, 290–295 (2006)
13. Visualizations: Line Graph of Vital Signs, <http://manyeyes.alphaworks.ibm.com/manyeyes/visualizations/line-graph-of-vital-signs> (accessed 18 July 2010)
14. Fonseca, T., Ribeiro, C., Granja, C.: Vital Signs in Intensive Care: Automatic Acquisition and Consolidation into Electronic Patient Records. *Journal Medical System* 33, 47–57 (2009)
15. Ordonez, P., desJardins, M., Feltes, C., Lehmann, C. U., Fackler, J.: Visualizing Multivariate Time Series Data to Detect Specific Medical Conditions. In: *Proc. AMIA 2008 Symposium*, pp. 530–534 (2008)

16. Andry, F., Naval, G., Nicholson, D., Lee, M., Kosoy, I., Puzankov, L.: Data Visualization in a Personal Health Record Using Rich Internet Application Graphic Components. In: Proc. HEALTHINF 2009 – International Conference in Health Informatics, pp. 111–116 (2009)
17. Garde, S., Knaup, P., Hovenga, E., Heard, S.: Towards semantic interoperability for electronic health records: Domain knowledge governance for openEHR Archetypes. *Methods of information in medicine* 46(3), 332–343 (2007)
18. Gao, T., Greenspan, D., Welsh, M., Juang, R., Alm, A.: Vital signs monitoring and patient tracking over a wireless network. Paper presented at the Proceedings of the 27th Annual International Conference of the IEEE EMBS (2005)
19. Friel, S.N., Curcio, F.R., Bright, G.W.: Making Sense of Graphs: Critical Factors Influencing Comprehension and Instructional Implications. *Journal for Research in Mathematics Education* 32(2), 124–158 (2001)
20. Cleveland, W., McGill, R.: Graphical perception and graphical methods for analyzing scientific data. *Science* 229(4716), 828–833 (1985)
21. Tarasewich, P., Campbell, C., Xia, T., Dideles, M.: Evaluation of visual notification cues for ubiquitous computing. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) *UbiComp 2003*. LNCS, vol. 2864, pp. 349–366. Springer, Heidelberg (2003)

Using R-Trees for Interactive Visualization of Large Multidimensional Datasets

Alfredo Giménez, René Rosenbaum, Mario Hlawitschka, and Bernd Hamann

Institute for Data Analysis and Visualization (IDAV), Department of Computer Science, University of California, Davis, CA 95616-8562

Abstract. Large, multidimensional datasets are difficult to visualize and analyze. Visualization interfaces are constrained in resolution and dimension, so cluttering and problems of projecting many dimensions into the available low dimensions are inherent. Methods of real-time interaction facilitate analysis, but often these are not available due to the computational complexity required to use them. By organizing the dataset into a level-of-detail (LOD) hierarchy, our proposed method solves problems of both inefficient interaction and visual cluttering. We do this by introducing an implementation of R-trees for large multidimensional datasets. We introduce several useful methods for interaction, by queries and refinement, to explain the relevance of interaction and show that it can be done efficiently with R-trees. We examine the applicability of *hierarchical parallel coordinates* to datasets organized within an R-tree, and build upon previous work in *hierarchical star coordinates* to introduce a novel method for visualizing bounding *hyperboxes* of internal R-tree nodes. Finally, we examine two datasets using our proposed method and present and discuss results.

1 Introduction

As measuring instruments advance technologically, datasets increase in both dimension and quantity. It becomes very difficult to interactively explore and analyze large, multidimensional datasets because of the high computational complexity required. Visualizing these dimensions also becomes a major problem for large dimensionalities and large datasets since standard visualization interfaces are constrained to a small number of dimensions and resolutions.

In fields of algorithms and complexity, efficient methods for data processing are often introduced through the use of hierarchical data structures. We have applied a hierarchical structure to large multidimensional datasets by generating an R-tree that contains the dataset. We utilized the efficiency of R-trees by implementing several interactive operations for analysis, and also used the hierarchical properties of R-trees to visualize the data at increasing levels-of-detail (LODs) in order to reduce visual clutter.

To achieve appropriate low-dimensional visualization of high-dimensional data within a hierarchy, we have implemented existing methods of hierarchical multidimensional visualization and have extended upon one of these methods in order

to accommodate it for more beneficial and efficient use within an R-tree structure. Specifically, we have examined *hierarchical parallel coordinates* and built on previous work to develop a new method for *hierarchical star coordinates*.

2 Previous Work

2.1 Multidimensional Visualization

Multidimensional visualization explicitly involves the problem of how to project d dimensions onto the a small number of dimensions available on visualization interfaces. Popular methods to do so include using visual cues, multiple visualizations, and alternative coordinate systems.

Chernoff [1] introduced a method using *visual cues* which involved transforming individual features of a face geometrically, and visualizing each multidimensional element as the resulting face. However, he stated that this technique is constrained to a small number of dimensions. This is an inherent problem; *visual cues* must be explicitly defined for each dimension.

Wright [2] introduced the use of multiple visualizations, scatterplot matrices, where a matrix of two-dimensional scatterplots is displayed such that every dimension is plotted against every other dimension. Similar multiple visualization schemes have been developed as well; however, with all these techniques, either the number of dimensions is constrained by the screen space available for multiple plots, or the visualization cannot display all dimensions at once.

Alternative coordinate systems attempt to provide a visualization for any number of dimensions. We have implemented and built on two of these techniques, specifically *parallel coordinates* [3] and *star coordinates* [4].

2.2 Hierarchical Visualization of Multidimensional Data

In order to generate a visualizable hierarchy from a dataset, several proposed methods involve hierarchical clustering algorithms. Fua [5] presented one of these algorithms based on proximity information and Linsen [6] presented another one based on density functions. Though effective for generation of a hierarchy, they both involve an added preprocessing step to cluster the data. These approaches have high computational complexity for generation and interactive operations, since they are not guaranteed to be balanced trees.

3 Main Idea

We introduce a method to generate a hierarchical structure of data which allows for efficient interactive operations as well as methods for visualization of data within this hierarchical structure. In contrast to previous work, our method provides a great degree of efficiency and requires minimal data-specific information, while also adding functionality for analysis.

We propose using R-trees to generate this hierarchy and examine the benefits for doing so in section 4. R-trees provide functionally visualizable aggregate items within an LOD-hierarchy while also increasing efficiency, which improves upon the problems encountered in some previous proposals (see subsection 2.2).

We examine methods to visualize aggregate items as well as data items within the R-tree in section 5. Some of these methods are already well-known, and we introduce a new method to visualize multidimensional R-tree aggregate items based on some existing proposals. We examine two types of interactive operations, queries and refinement, in section 6. Finally, we apply our proposed methods on real datasets in section 7.

4 R-Trees: An Effective Data Structure for Interactive Visualization of Large Multidimensional Datasets

In order to provide 1) a scalable hierarchy for large multidimensional datasets, 2) visualizable and accurately representative aggregate items within that hierarchy, and 3) efficient interactive operations on the structure, we propose organizing datasets into R-trees.

4.1 Generation of an LOD-Hierarchy

R-trees generate an LOD-hierarchy of aggregate and data items in a “bottom-up” fashion. All individual data elements are inserted into the bottom level, and nodes are split into two new ones when their respective number of children

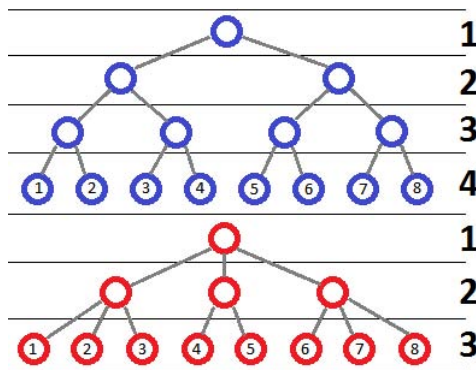


Fig. 1. Two R-trees with the same 8 elements inserted, but with different user-specified values for m (maximum number of children per node). The top has an m value of 2 and 4 levels-of-detail (LODs), while the bottom has an m value of 3 and 3 LODs. The top requires storage of 7 *hyperboxes* (one per internal node), while the bottom requires storage of 4. The ability to define m allows for dynamic LODs and storage space, and in turn, different quantities of refinable *hyperboxes* and regions-of-interest (ROIs) within each level.

exceeds the maximum number of children, m . Whenever the root node is split, a new LOD in the hierarchy is introduced. Every internal node contains a number of children and a region which bounds all of its children. Node splitting in R-trees is a widely covered research topic, as the optimal solution requires factorial time complexity [7]. Our implementation uses linear splitting, a method which delivers accurate enough results for our application as well as linear time complexity.

R-trees allow for alteration of their internal tree depth, and different tree depths directly affect both user preferences and storage space. We can control the depth of the hierarchy by specifying different values for m . A larger value of m corresponds to fewer splits and, therefore, fewer levels within the hierarchy. With more children per node, there are more available refinable nodes per level and less total aggregate items. This means that there are more detailed specification of ROIs within a level, and less total internal storage space is required. However, having too many children per node contributes to visual clutter and less LODs within the hierarchy. For smaller datasets, a small value of m is useful, because more LODs organize the data more efficiently for interactive operations and introduce more LODs. Lower values do, however, increase the storage space. An illustration of the differences between high and low m values is shown in figure 11.

4.2 Effective, Visualizable Aggregates

In order to allow for a scalable representation, we require a structure that not only organizes the data into an LOD-hierarchy, but allows for appropriate visualization of levels within it. For this reason, it is crucial that we generate aggregate items that are accurately representative to the actual data, as well as usable in various visualization schemes. An item that is accurately representative of the data is one which does not remove semantic information from the dataset.

The R-tree aggregate items are ranges of values for each of the total d dimensions, which we will denote as *hyperboxes*. In one dimension, a *hyperbox* is a range of points, or an extension of a single point, which is a line segment. In two dimensions, a *hyperbox* is a range of lines, or an extension of a single line segment, which is a rectangle. We continue this process of extending lower-dimensional *hyperboxes* in order to generate *hyperboxes* of unlimited dimensionality.

These *hyperboxes* are accurately representative aggregate items for visualizing internal levels of a hierarchical data structure because they denote where the children of their respective nodes are as well as how sparse or dense the elements within that *hyperbox* are, due to their bounding property. These characteristics allow the user to draw conclusions about what values within the dimensions of the data are most common as well as how varied the dimensional values are in comparison with each other.

4.3 Efficiency for Real-Time Interaction

It is crucial for our application to interactively operate on datasets that are not only large in quantity of elements, but large in dimensionality as well; therefore, generation, queries, and refinement operations must be low in computational

complexity. The use of R-trees allows us to execute hierarchical generation and interactive operations very quickly, even with large datasets of many dimensions.

R-trees are inherently balanced trees, which provides a great deal of efficiency. Every time a node is split, its children are distributed amongst the new nodes in order to maintain the same depth throughout the R-tree and avoid empty nodes. This property allows for insertions, deletions, and searches to be made in worst-case $O(md \log_m n)$ time for n data elements of d dimensions. Our proposed method to execute queries requires even less time than searches, as we will explain in detail in section 6.2. This method for generation of a hierarchy improves upon Linsen's [6], which does not maintain tree balance and generates many empty nodes. Furthermore, while Linsen's [6] method applies a more accurate automatic generation of clusters, it necessitates specification of a density function and introduction of another preprocessing step to evaluate densities and quantities of clusters.

5 Visualization of Datasets within R-Trees

The visualization of massive multidimensional datasets as organized within R-trees requires a transformation from the d dimensions of the R-tree data into the two dimensions available on screen space, as well as effective methods of visualizing both aggregate items and individual data elements.

We examined and implemented two alternative coordinate systems for multidimensional visualization. We show that R-trees are visualizable using *hierarchical parallel coordinates*, and introduce a method which builds upon Kandogan's [4], which we denote as *hierarchical star coordinates*. In both cases, we describe how to represent multidimensional data elements as well as bounding *hyperboxes*.

5.1 Hierarchical Parallel Coordinates

Parallel coordinate visualization was defined by Inselberg [3], and has been extended to represent multidimensional value ranges, *hyperboxes*, by Fua [5]. In *parallel coordinates*, each dimension is denoted by a single line such that all lines are unique and parallel to each other, and points are represented as polygonal lines with values plotted on each respective dimensional line. To represent *hyperboxes*, we simply plot two data elements in this fashion, the maximum and minimum, and fill the area between both segments, so that we attain a polygon which covers all values within the range of the *hyperbox*.

5.2 Hierarchical Star Coordinates

For single elements within *star coordinates*, the technique is, again, explicitly defined [4], and we propose extending this idea to also represent *hyperboxes*, as Fua [5] did with *parallel coordinates*. The dimensional axes are represented by a set of lines which all emanate from a single point (the *star coordinate* origin). The data elements in *star coordinates* can either be represented as a polygonal

line which connects dimensional values, or as a single point which is translated in the direction of each dimensional line by the magnitude of the value. We use the latter. In order to represent the *hyperboxes*, we cannot simply plot the minima and maxima of the range as with *parallel coordinates*, because the area between the minimum point and maximum point no longer accurately represents the range. Instead we introduce a method that plots all possible combinations of the minimum and maximum values in each dimension—the corners of the *hyperbox*—and fills the area between those points. We fill the area by calculating the convex hull of these points and constructing its respective convex polygon.

6 Interactive Operations for Visualization and Analysis

6.1 Refinement Methods for Dynamic Removal of Clutter

The fact that R-trees are an LOD-hierarchy allows for several methods to remove clutter, both programmatically and interactively. Clutter is defined as the ratio of LOD to available screen resolution; thus, LOD corresponds directly to the amount of clutter in the visualization. In an LOD-hierarchy, it is possible to refine down the hierarchy and therefore alter the LOD of the visualization dynamically. Dynamic alteration of LODs, in turn, allows for dynamic removal of clutter.

To be more explicit, refinement means breaking down certain regions within the R-tree into their more detailed components. This is done by removing a *hyperbox* from the visualization and replacing it with its child *hyperboxes* or data elements. This provides us with a more accurately detailed visualization.

Refinement can be done uniformly or non-uniformly as well as programmatically or interactively, with different benefits for each.

Uniform Programmatic. We introduce one uniform programmatic method for refinement: a simple breadth-first search (BFS). This method refines all *hyperboxes* of a single level within the hierarchy. In this way, it is possible to alter the LOD uniformly—all elements visualized have the same LOD at all times. In this way, the user can draw initial conclusions about the dataset as a whole and determine which areas are more of interest than others. When the user determines a region within the dataset that is particularly of interest, the ability to refine non-uniformly and interactively becomes crucial.

Non-Uniform Interactive. To facilitate interactive non-uniform refinement, we introduce a method to execute queries. These queries allow the user to define which dimensions and regions are of interest, and then refine the corresponding *hyperboxes* as desired.

6.2 Interactive Queries for Real-Time Analysis

The user may construct and execute two types of queries on the R-tree: 1) bounded and 2) overlap. Both query methods iterate over nodes of the R-tree and execute comparisons between the constructed query and each node processed.

Both also require the same input: a set of 3-tuples, which each specify 1) a dimensional index, from 1 to d inclusively, 2) a value within that dimension, and 3) a margin value.

Bounded Queries. *Bounded queries* find nodes whose *hyperboxes* completely encompass the query values and margins in the specified dimensions.

This type of query facilitates interactive searching for programmatically generated clusters of data—because it tests for nodes that completely encompass the query region, this is an effective way for the user to find bounded clusters created by the R-tree generation.

Overlap Queries. *Overlap queries* find all nodes which overlap any part of the query values and margins in the specified dimensions.

Because *overlap queries* allow searching for any data within the specified range, they are useful for drawing conclusions about the data regardless of the internal R-tree structure, and therefore is based on the data elements rather than the data structure.

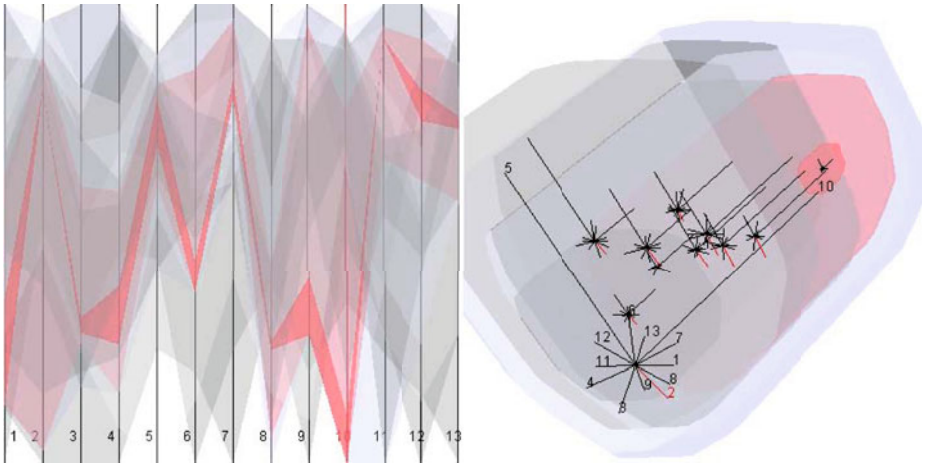


Fig. 2. We highlight visualized hyperboxes in parallel coordinates (left) and star coordinates (right). By increasing the LOD of a large hyperbox containing sparsely distributed data, we obtained detail of a small hyperbox (at the end of the axis numbered 10) within that region containing densely distributed data (the small red hyperbox).

7 Visual and Interactive Data Exploration and Analysis

7.1 Isolating Outliers

It is essential in visual data analysis to isolate and either extract more detail from or eliminate outliers. With our system, it is possible to discover outliers quickly and either decrease their significance or refine them in order to examine

them more closely. We explain the process by example, with a dataset of regional wines [8] of 13 dimensions and 178 elements.

As 178 elements is a fairly small number of data, we choose a small value for m , 2, in order to increase LODs available. Next, we execute BFS refinements to draw initial conclusions about where outliers may lie. From this step, we can see distributions of values in each dimension. Some are densely packed around certain values, like dimension 10 and dimension 5. The outliers in each dimension are those values which lie outside of the densely packed regions. In order to show the efficacy of removal as well as examination, we remove the outliers in dimension 5 and examine in detail the outliers of dimension 10.

In order to remove an outlier, we construct a query which contains it. As explained in section 6, when we are looking for specific elements, like outliers, *overlap queries* are more effective. After running the *overlap query* and coloring the result white, the outliers in dimension 5 barely contribute to the visualizations.

To examine an outlier, we execute an *overlap query* in red followed by a number of overlap refinement operations until we obtain the LOD required. After just a few overlap refines, we achieve a very specific outlying region visualized in both the *parallel coordinates* and *star coordinates*, while avoiding clutter due to the region-specific refinement operations.

7.2 Examining Correlations

We can examine correlations between dimensions and between individual clusters/elements by performing refinement operations until we achieve the desired LOD in a ROI, and arranging the visualization to show correlations. As example we analyze a dataset of forest fires within the northeast region of Portugal [9].

After initial setup, we determine a good ROI and begin rearranging the visualization methods in order to analyze correlations. We rearrange the *parallel coordinate* axes to observe dimensional correlations: high values in dimension 4 correlate with low values of dimensions 11 and 12. In the *star coordinate* view, we increase the magnitude and vary the direction of certain dimensions, in this case 4, 7, 11, and 12, shown in figure 3. As we can manipulate these axes, we observe to what extent the shape of the aggregates is affected. The blue aggregates are fairly unaffected by manipulation of dimensional axis 11 and highly affected by manipulation of dimensional axis 4; therefore, these aggregates have low values in dimension 11 and high values in dimension 4. Furthermore, large hyperboxes represent very sparse distributions of data, observed in red, whereas small hyperboxes represent dense distributions, observed in blue. We conclude that a large quantity of the data has fairly high values in dimension 4 and extremely low values in dimension 11.

8 Possible Drawbacks

One principal drawback of our *hyperbox* visualization method is that it requires $O(2^d)$ complexity to calculate the *hyperbox* corners. The effects are rather detrimental if visualization of 20 or more dimensions is required, so improved methods

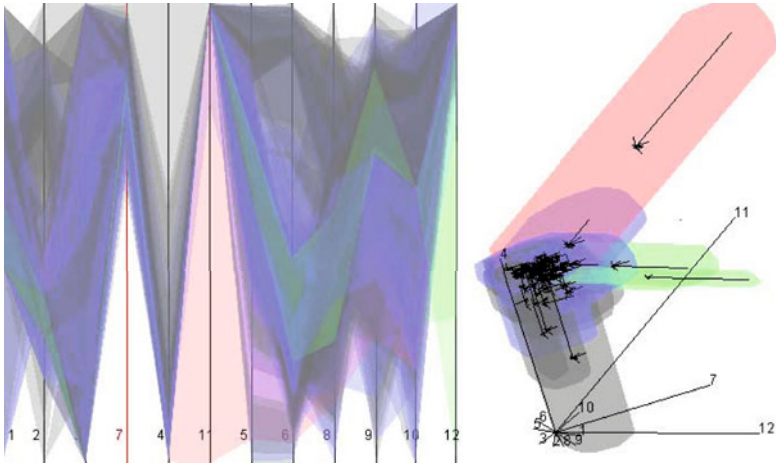


Fig. 3. We show hyperboxes as colored regions in parallel coordinates (left) and star coordinates (right) for a dataset of forest fires. Each numbered line in the star coordinate visualization corresponds to a dimension. These numbered lines can be manipulated in direction and length. If we extrude the line numbered 11, the red hyperbox is extruded more than any other hyperboxes. Therefore, the red hyperbox contains data with a high range of values in dimension 11.

would be necessary to provide real-time visualizations at this level of dimensionality. Note that this complexity applies to the visualization, rather than the interactive operations.

9 Conclusions and Possible Future Research

We have implemented and built upon several existing methods for multidimensional visualization and visualizable hierarchical structuring of multidimensional datasets. We have introduced a novel method to generate an efficient LOD-hierarchy for large, multidimensional datasets using R-trees, we have examined methods to visualize *hyperboxes* and elements within that LOD-hierarchy, and we have examined the use of interactive operations on the data to facilitate analysis. We have used existing visualization schemes, *parallel* and *star coordinates*, in order to introduce a new method for visualizing *hyperboxes*, while retaining the ability to use existing visualization methods as well. Our method for LOD-hierarchy generation provides a great deal of efficiency and functionality in contrast to previous ones, and in combination with the introduced visualization schemes and interactive operations, added benefits for analysis and exploration of data.

A possible improvement to the drawback of complexity mentioned in section 8 could be to apply Linsen's [6] splat-based ray-tracing method to these *hyperboxes*, in which case the complexity would be constrained by screen resolution, rather

than the data dimensionality. Another possible improvement, for more accurate hierarchical cluster generation, could be to develop new node-splitting algorithms based on factors other than proximity.

Future implementations of our method could significantly influence areas which use progressive refinement, such as Rosenbaum's [10] technique for device adaptation. As progressive refinement methods require generation of LOD-hierarchies for many types and sizes of multidimensional data, our method provides much of the necessary functionality.

Acknowledgements

René Rosenbaum was supported by the German Research Foundation Deutsche Forschungsgesellschaft (DFG), and Mario Hlawitschka was supported in part by NSF grant CCF-0702817. We thank our colleagues from the Institute of Data Analysis and Visualization (IDAV) at UC Davis.

References

1. Chernoff, H.: The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association* 68, 361–368 (1973)
2. Wright, D.B.: Scatterplot matrices. *Encyclopedia of Statistics in Behavioral Science* 4, 1794–1795 (2005)
3. Inselberg, A.: The plane with parallel coordinates. *The Visual Computer* 1, 69–91 (1985)
4. Kandogan, E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001*, pp. 107–116. ACM, New York (2001)
5. Fua, Y.H., Ward, M.O., Rundensteiner, E.A.: Hierarchical parallel coordinates for exploration of large datasets. In: *Proceedings of the Conference on Visualization 1999: Celebrating Ten Years*, pp. 43–50. IEEE Computer Society Press, Los Alamitos (1999)
6. Linsen, L., Long, T.V., Rosenthal, P., Rosswog, S.: Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 1483–1490 (2008)
7. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: *International Conference on Management of Data*, pp. 47–57. ACM, New York (1984)
8. Forina, M.: An extendible package for data exploration, classification and correlation (2010)
9. Cortez, P., Morais, A.: A data mining approach to predict forest fires using meteorological data. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) *EPIA 2007. LNCS (LNAI)*, vol. 4874. Springer, Heidelberg (2007)
10. Rosenbaum, R., Hamann, B.: Progressive presentation of large hierarchies using treemaps. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnaç o, M.L., Silva, C.T., Coming, D. (eds.) *ISVC 2009. LNCS*, vol. 5876, pp. 71–80. Springer, Heidelberg (2009)

Combining Automated and Interactive Visual Analysis of Biomechanical Motion Data

Scott Spurlock¹, Remco Chang², Xiaoyu Wang¹, George Arceneaux IV¹,
Daniel F. Keefe³, and Richard Souvenir¹

¹ Department of Computer Science, The University of North Carolina at Charlotte

² Department of Computer Science, Tufts University

³ Department of Computer Science and Engineering, University of Minnesota

Abstract. We present a framework for combining automated and interactive visual analysis techniques for use on high-resolution biomechanical data. Analyzing the complex 3D motion of, e.g., pigs chewing or bats flying, can be enhanced by providing investigators with a multi-view interface that allows interaction across multiple modalities and representations. In this paper, we employ nonlinear dimensionality reduction to automatically learn a low-dimensional representation of the data and hierarchical clustering to learn patterns inherent within the motion segments. Our multi-view framework allows investigators to simultaneously view a low-dimensional embedding, motion segment clustering, and 3D visual representation of the data side-by-side. We describe an application to a dataset containing thousands of frames of high-speed, 3D motion data collected over multiple experimental trials.

1 Introduction

As CT technology becomes more mature, scientists are now able to capture high speed motion of bones and joints at the rates of 250 to 500 frames per second with sub-millimeter accuracy [1]. This new imaging modality has allowed the scientists to examine animal kinematics movement in ways that have not been possible before. However, with such rich information, new methods for analyzing biomechanical motion data become increasingly important.

Some work has already considered exploring this data both visually and interactively [2,3]. These systems allowed the scientists to examine raw biomechanical data, but with limited automated analyses. Visualizations have been developed to reanimate the motion data by displaying bones moving in space and overlaying additional data, such as instantaneous helical axes computed from the motion [2]. Recently, Keefe et al. developed an interactive system that combined these 3D motion visualizations with complementary 2D visualizations to better capture the higher dimensionality of the data [3]. These two systems have aided scientists in characterizing the data and finding patterns within the animals' behaviors. There are limitations to these systems, however, based on their reliance on user interactions and visual inspection. The sheer volume and complexity of the data may obscure patterns or relationships from manual discovery.

In this paper, we extend the previous work on biomechanical motion analysis to incorporate automated methods for analysis. Specifically, we incorporate dimensional reduction and clustering techniques to reduce complexity and highlight patterns within the data. We then integrate the result of these unsupervised learning approaches into an interactive tool to enable visual analysis.

While these automated analysis techniques for biomechanical motion are common in the machine learning and computer vision fields, they have rarely been integrated with highly interactive visual analytical systems. By leveraging a blended model of learning approaches with interactive visual analysis, we aim to enable a new style of hybrid investigation [4,5]. Using this system, an investigator can utilize automated computational methods to identify hidden temporal patterns embedded within the data while retaining full interactive exploration capabilities to analyze the data either in raw or post-processed form. In our system, a low-dimensional graphical representation of the data can be viewed concurrently with clustering output, and side-by-side with relevant video clips. We believe that this synthesis is an advancement over previous systems in which the burden of investigation is solely dependent upon the user, and that by using this hybrid approach the user can identify new patterns more quickly and in a repeatable fashion.

2 Related Work

Much research has focused on analyzing biomechanical motion through 3D visualizations. Most of these 3D visualization systems focus on rendering the spatial changes of different parts of motion data, for example, the trajectory of jaws or the rotation of joints [3]. Using direct manipulations, most of these systems allow the user to control the viewpoint and to focus on movements of specific parts. In recent work, Keefe et al. [2] presented an effective 3D visualization framework for biomechanical motion analysis with interactive visualizations with detailed anatomical features.

The other major aspect of biomechanical motion analysis is the temporal patterns. Visualizing trends in time-varying and multi-variate data has been considered in depth within the information visualization community [6]. However, for visualization of biomechanical motion over time, 3D views of the model are often animated, and additional data attributes are often visualized using color, texture, streamlines, and 3D data glyphs [2,7,8]. While these annotated 3D views can be quite powerful, it has been suggested that understanding trends over time through animation may not be the most effective strategy [6]. Our system follows the framework developed by Keefe et al. [3] in that we also utilize a multiple coordinated visualization approach and support analysis using both the 3D model view and 2D information visualizations.

3 Biomechanical Motion Data

The techniques and example application in this paper are presented with the goal of applying broadly across different types of motion analysis of interest in the



Fig. 1. Example images from the pig chewing motion data set

biological sciences. The specific datasets driving the framework presented here come from a study at Brown University that made use of marker-based X-ray Reconstruction of Moving Morphology (XROMM)¹ to capture measurements of the lower jaw movements of miniature swine (Sinclair strain) during mastication.

Although these data describe the motion of just two bones, these bones connect to each other at two joints and also whenever the teeth come into contact with each other; thus, the data serve as an ideal springboard for complex high-dimensional, multi-joint analyses of other biomechanical structures. The chewing motion itself is quite unique among animals, hence the interest from an evolutionary biology perspective in studying the coordinated motion of these bones under different experimental conditions.

From the raw data collected from multiple high-speed fluoroscopic videos captured experimentally, 4×4 transformations can be derived to describe the rigid body transformations (translation and rotation) of the pig's mandible in relation to its skull (see Figure 1). Previous research has identified groupings of particular sequences of frames into segments, which identify related, temporally proximate frames.

4 Automated Analysis of Biomechanical Motion Data

As described in Section 3, the biomechanical motion data is represented as 4×4 transformation matrices describing the positions of the individual bones. This high-dimensional data describes the configuration of the animal at a particular timestep and the time-series describes the biomechanical motion. In order to better understand and visualize the motion and discover any underlying patterns, we employ *dimensionality reduction* and *data clustering*. In this section, we describe the steps for the automated analysis of this data.

4.1 Dimensionality Reduction

Most data analysis techniques on high-dimensional points and point sets do not work well. One strategy to overcome this problem is to find an equivalent lower (typically 2 or 3) dimensional representation of the data. Dimensionality reduction is the technique of automatically learning a low-dimensional representation for data. The most well-known techniques are Principal Component Analysis

¹ <http://xromm.org>

(PCA) [9] and Independent Component Analysis (ICA) [10]. These methods seek to represent data as linear combinations of a small number of basis vectors. However, many data sets, including the transformation matrices of the biomechanical motion data considered in this work, tend to vary in ways which are very poorly approximated by changes in linear basis functions.

Techniques in the field of manifold learning embed high-dimensional data points which lie on a *nonlinear* manifold onto a corresponding lower-dimensional space. There exists a number of automated techniques for learning these low-dimensional embeddings, such as Isomap [11] and LLE [12]. These methods have been used in computer vision and graphics for many applications, including medical image segmentation [13] and light parameter estimation from single images [14]. In this paper, we use the Isomap algorithm, but the general approach could be applied with any of the other nonlinear dimensionality algorithms.

Isomap embeds points in a low-dimensional Euclidean space by preserving the geodesic pair-wise distances of the points in the original space. To estimate the (unknown) geodesic distances, distances are calculated between points in a trusted neighborhood and generalized into geodesic distances using an all-pairs shortest-path algorithm. With most manifold learning algorithms, discovering which points belong in the trusted neighborhood is a fundamental operation. Typically, the Euclidean distance is used, but in certain cases other distance measures have been shown to lead to a more accurate embedding of the original data [15]. Due to the structure of the transformation matrices (most notably the rotational component) used in our data, we use a distance metric based on exponential matrix mapping described in [16].

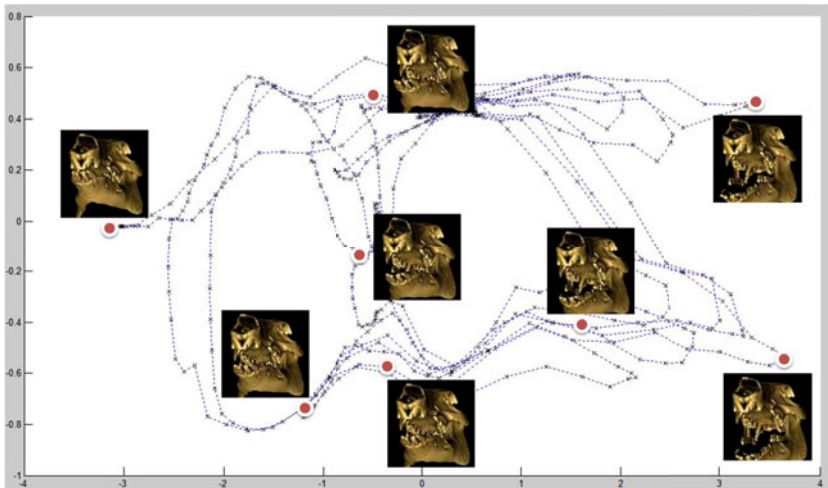


Fig. 2. 2D Isomap embedding of the pig chewing data set. Each point corresponds to an original frame of data. For the indicated points, the corresponding image is shown.

Figure 2 shows the Isomap embedding of one of the data sets. Each 2D point represents one of the original frames from the video data and the corresponding images of selected points are shown. One can observe a *perceptual organization* of the original data in the embedded space. The x - and y -axes correlate with the two major components of the motion: opening and closing of the mouth and lateral motion of the mandible, respectively.

This embedding step describes the relationship between each of the frames in the original data set, but doesn't yet describe the known structure within the data. Each of the biomechanical motions considered in the work are comprised of multiple, short, temporal segments that correspond to distinct phases of motion. The shape of these segments in the embedded space can be used compare multiple motion patterns. In order to discover the similarity among segments within a data set, we apply hierarchical clustering.

4.2 Clustering

The process of comparing and grouping temporal segments from the embedding into clusters can be automated using well-known unsupervised data clustering techniques. We use agglomerative clustering to iteratively and hierarchically merge similar segments into clusters. Depending on the goal of the analysis, single-link, complete-link, or average-link can be used. In all three methods, the distance measure [17] applied between segments uses a combination of curve fitting and string matching techniques to calculate the similarity of two trajectories. The metric is scale-, rotation-, translation-, and duration-invariant. Figure 3 shows four segments from the data shown in Figure 2. Our clustering process iteratively groups the segments shown from left to right as being the most similar in shape. From this process we generate a dendrogram, a hierarchical tree structure, to allow the end-user to interactively choose the level of grouping that most meaningful to the investigation.

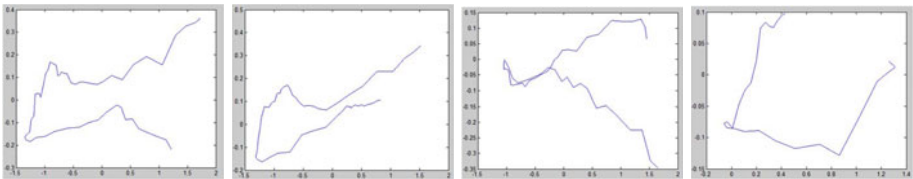


Fig. 3. Separate phases of motion represented as segments from the Isomap embedding. The two on the left were measured to be the closest in shape.

5 Interactive Visual Analysis of Patterns in Motion Data

In this section, we describe an interactive visual interface for analyzing 3D motion data. This interface displays the raw data using an animated 3D model view, as well as the automated analysis output described in section 4. By integrating

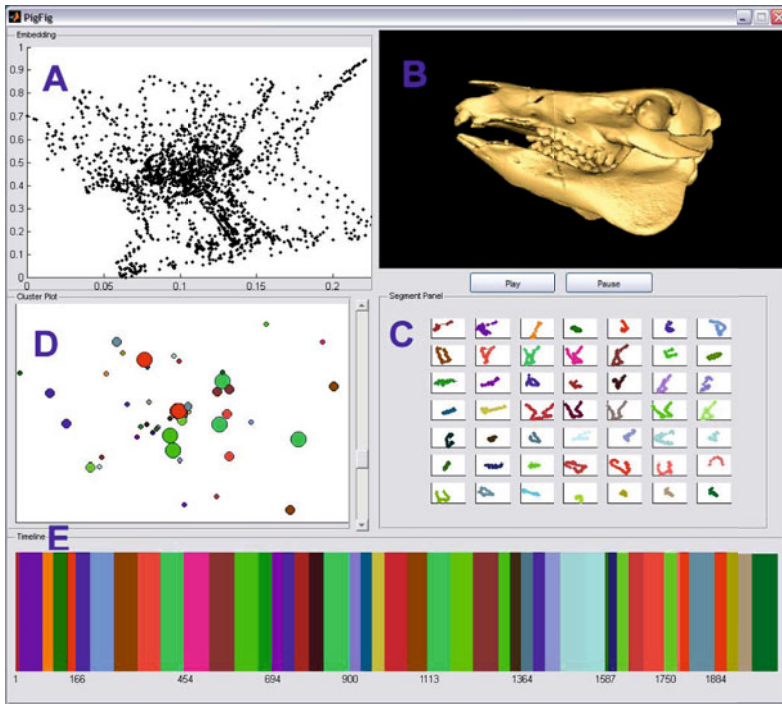


Fig. 4. The overview of our interactive visual analytics system. (A): The embedding view. (B): The 3D visualization view. (C): The segment view. (D): The cluster view. (E): The timeline view.

these views using a multiple-coordinated-views approach [18], our system provides methods to perform interactive analysis across modalities. Shown together in Figure 4, these views allow the user to explore the 3D motion sequence in space and time, in its embedding space, as well as at a clustering level.

5.1 Exploring the Motion Space

Given our initial dataset of 3D transformations, we utilize an interactive 3D visualization and timeline view to help the user analyze the motion data from both spatial and temporal perspectives.

3D Visualization for Motion Exploration. Following related work on biomechanical motion visualization, our system supports motion playback, viewpoint adjustment, and motion sequence comparison. Figure 4 (B) shows the 3D visualization in our system. The user can interactively change the viewpoint of the model and focus on the movement at each time step of the data. Coordinated with the embedding space, the 3D visualization enables the users to explore the precise spatial location of each motion sequence. If any specific motion sequences

have been selected, our system will also show a side-by-side playback panel to help the user compare the motion difference between these sequences.

Timeline Visualization for Temporal Patterns. To aid in finding temporal patterns, a timeline visualization shows time in relation to segments of motion frames, as well as hierarchical clusters of segments. Figure 4 (E) shows grouping of the motion data based on the predefined segments. These segments are dynamically color-coded to correspond to the cluster into which they have been grouped. As the user traverses the clustering hierarchy, the number of clusters will change accordingly. Higher thresholds results in fewer clusters, but allows the user to identify high-level behavioral patterns in the timeline; whereas a lower threshold produces more clusters, but allows the user to examine the segments in greater detail.

5.2 Analyzing the Embedding Space

The low-dimensional embedding space and the hierarchical clustering provide a foundation for further interactive analysis. When a dataset is first loaded, an overview of the data is displayed using the three coordinated views seen in Figure 4: a small-multiples view, an embedding view, and a 2D plot of each segment whose positions are based on their similarities. These three views have been carefully chosen for their analytical capabilities in analyzing different aspects of a 3D motion sequence.

Small Multiple Views. We designed the small-multiples view to represent individual motion sequences in the embedding space. Because biomechanical motion data typically contains cycles, a key feature of our system is to help group and analyze the cyclic motions. Figure 4 (C) shows how these motion segments are assigned to small-multiple images corresponding to a trajectory in the embedding space. Each segment represents one cycle of frames, where each frame is a point in the embedding. Clicking on a segment will draw the trajectory on the relevant points in the embedding and show an animation of the corresponding frames in the 3D visualization view (see figure 4 (B)). The background color of each multiple encodes the data segment and corresponds to those in the cluster and timeline views (see Figure 4 (D)).

Embedding Space View. Within the embedding (unlike the original high-dimensional data space), the Euclidean distance represents the dissimilarity between data points. To highlight the data distribution and correlation between different motions, we naturally display this as a 2D scatterplot where each frame of the motion data is encoded as a point. Previously defined segments encapsulate particular sequences of motions, which correspond to a sequence of points in the embedding. Using standard mouse interactions, the user can analyze the low-dimensional representation to explore the perceptual organization of a particular data set. The user can examine each frame by mousing over the points, which will automatically update the 3D visualization.

Zoomable Cluster View. Our system provides an interactive, zoomable cluster view. This view is a graph visualization that shows a point for each cluster. The points are positioned using multi-dimensional scaling (MDS) with the distance metric described in Section 4. Each bubble in this view indicates one cluster and is color-coded in accordance with the small-multiple and timeline views. Each cluster varies in size with the number of frames it encapsulates. As shown in Figure 4 (D), the cluster view enables user to interactively zoom in and out on different levels of grouping. The user can use the scroll bar to choose different levels of the grouping results, which automatically updates the size of individual clusters and corresponding coloring scheme.

5.3 Connecting the Embedding Space and the Data Space

The interaction and linked views are the keys for the user to simultaneously explore and analyze both the high-dimensional motion space and the lower-dimensional embedding. Since all views are coordinated visually and interactively, they collectively provide a cohesive exploration environment and support analysis of both spatial and temporal perspectives. For example, the embedding space view may be animated either through interaction in the embedding view or by selecting individual segments from the small multiple view. Also, the timeline can depict the temporal relationships at multiple clustering levels.

6 Discussion

In this paper, we introduce an integrated system that combines automated analytical methods with interactive visual analysis. Compared to existing work, our approach is innovative in that automated analysis can reduce the amount of ambiguity introduced through a user's interactions. For instance, one of the most important features in the work by Keefe et al. [3] is the small-multiples view showing a trace of a point on the pig's teeth plotted over time. This view is similar to the small-multiples view shown in Figure 4 (C). However, the key difference is that the tracer view requires the user to manually interact with the 3D view such that the front of the pig's model is facing the user's viewpoint. Only using this particular perspective can the tracer view show a pig's bilateral chewing behavior, which is a sideways grinding motion during a pig's chewing cycle. However, examined from the side-view, this subtle motion in the chewing would have been unnoticed. In contrast, with our method, we learn an embedding of the pig's motion, which is analytically justified and is without the ambiguity of a user's interactions. Since every user and analysis session presents the same embedding visualization, multiple analyses will be more consistent, and the users are more likely to detect the same behaviors, thus providing a more defensible analysis result each time [19].

The agglomerative clustering we use provides the user a hierarchical structure to explore possible repeating phases within the motion. The user can interactively choose an appropriate threshold given a specific analytical goal. For instance, the user can choose a high threshold in the clustering view, which will

produce fewer clusters of similar chewing segments to discover high-level behaviors. On the other hand, to identify low level differences between the segments, a lower threshold can be used to examine which of the segments are the most similar. When used in conjunction with the 3D comparison view, the user can then discover minute difference between the segments.

Without the combined use of automated methods and interactive visual representations, such analyses and discoveries may not be possible. Even with systems that allow for highly interactive visual analysis, the types of analyses are usually limited to visual comparisons by the user, which can be ambiguous depending on the user's selections of viewpoints or segments of interest. Furthermore, the user cannot perform analyses across multiple levels of abstractions such as the features that our system can provide through clustering and interactive selection of thresholds.

7 Conclusions and Future Work

With new advances in scientific imaging, an increasing amount of high-resolution, high-quality biomechanical movement data is becoming available. With this opportunity comes the challenge of enabling scientists to make sense of information that is complex, temporal, multiphase, and cyclic. We presented a framework targeted at helping researchers meet this challenge. By combining machine learning methods with interactive visualization techniques, we provided users with a multi-pronged, hybrid approach to investigation. We demonstrated the combination of multiple, simultaneous views of the data where each view supports independent interaction, but work in concert to support more complex analysis.

This system is a positive first effort towards tighter integration of the user experience with the underlying analysis methods. In the future, we plan to conduct case studies with domain experts and apply our system to broader studies across data sets from more diverse domains. In addition, we plan to investigate additional analysis methods for the automated analysis of this type of temporal data. For example, Hidden Markov Models (HMM) could be used for data where the segments are not explicitly defined but can be learned based on the original data sequence. These potential additions combined with the general approach of blending automated and multi-view, interactive visual analysis open the door to new insights from scientific data analysis and exploration.

Acknowledgements. We wish to thank Elizabeth Brainerd and the XROMM group at Brown University for the insight, infrastructure, and data that enabled us to explore this research within the context of the pig mastication application. Thanks also to David Laidlaw for growing the initial interdisciplinary collaborations that made this research possible.

References

1. You, B., Siy, P., Anderst, W., Tashman, S.: In vivo measurement of 3-d skeletal kinematics from sequences of biplane radiographs: Application to knee kinematics. *MedImg* 20, 514–525 (2001)

2. Keefe, D.F., O'Brien, T.M., Baier, D.B., Gatesy, S.M., Brainerd, E.L., Laidlaw, D.H.: Exploratory visualization of animal kinematics using instantaneous helical axes. *Computer Graphics Forum (EuroVis Special Issue)* 27, 863–870 (2008)
3. Keefe, D.F., Ewert, M., Ribarsky, W., Chang, R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics* 15, 1383–1390 (2009)
4. Thomas, J., Kielman, J.: Challenges for visual analytics. *Information Visualization* 8, 309–314 (2009)
5. Ribarsky, W., Fisher, B., Pottenger, W.: Science of analytical reasoning. *Information Visualization* 8, 254–262 (2009)
6. Robertson, G., Fernandez, R., Fisher, D., Lee, B., Stasko, J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 6, 1325–1332 (2008)
7. Marai, G.E., Laidlaw, D.H., Andrews, S., Grimm, C.M., Crisco, J.J.: Estimating joint contact areas and ligament lengths from bone kinematics and surfaces. *IEEE Transactions on Biomedical Engineering* 51, 790–799 (2004)
8. Jan, S.L.V.S., Clapworthy, G.J., Rooze, M.: Visualization of combined motions in human joints. *IEEE Computer Graphics and Applications* 18, 10–14 (1998)
9. Jolliffe, I.T.: *Principal Component Analysis*. Springer, Heidelberg (1986)
10. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley and Sons, Chichester (2001)
11. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
12. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 2323–2326 (2000)
13. Zhang, Q., Souvenir, R., Pless, R.: On manifold structure of cardiac MRI data: Application to segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 1092–1098. IEEE Computer Society, Los Alamitos (2006)
14. Winnemöeller, H., Mohan, A., Tumblin, J., Gooch, B.: Light waving: Estimating light positions from photographs alone. *Comp. Graphics Forum* 24, 433–438 (2005)
15. Souvenir, R., Pless, R.: Image distance functions for manifold learning. *Image Vision Comput.* 25, 365–373 (2007)
16. Alexa, M.: Linear combination of transformations. *ACM Trans. Graph.* 21, 380–387 (2002)
17. Hsieh, J., Yu, S., Chen, Y.: Motion-based video retrieval by trajectory matching. *IEEE Trans. Circuits and Systems for Video Technology* 16, 396–409 (2006)
18. Roberts, J.C.: State of the art: Coordinated & multiple views in exploratory visualization. In: *CMV 2007: Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, Washington, DC, USA, pp. 61–71. IEEE Computer Society, Los Alamitos (2007)
19. Thomas, J.J., Cook, K.A.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr (2005)

Human Activity Recognition: A Scheme Using Multiple Cues

Samy Sadek¹, Ayoub Al-Hamadi¹, Bernd Michaelis¹, and Usama Sayed²

¹ Institute for Electronics, Signal Processing and Communications
Otto-von-Guericke-University Magdeburg, Germany

² Department of Electrical Engineering
Assiut University, Assiut, Egypt

{Samy.Bakheet, Ayoub.Al-Hamadi}@ovgu.de

Abstract. In this work, a schematic model for human activity recognition based on multiple cues is introduced. In the beginning, a sequence of temporal silhouettes of the moving human body parts are extracted from a video clip (i.e., an action snippet). Next, each action snippet is temporally split into several time-slices represented by fuzzy intervals. As shape features, a variety of descriptors both boundary-based (Fourier descriptors, Curvature features) and region-based (Moments, Moment-based features) are then extracted from the silhouettes at each time-slice. Finally, an NB (Naïve Bayes) classifier is learned in the feature space for activity classification. The performance of the method was evaluated on the KTH dataset and the obtained results are quite encouraging and show that an accuracy on par with or exceeding that of existing methods is achievable. Further the simplicity and computational efficiency of the features employed allow the method to achieve real-time performance, and thus it can provide latency guarantees to real-time applications.

1 Introduction

Recognizing human activities from video has emerged as one of the most important concerns in the field of pattern recognition and artificial intelligence over the last two decades. In spite of the voluminous existing literature on the analysis and interpretation of human motion motivated by the rise of security concerns and increased ubiquity and affordability of digital media production equipment, research on human activity and event recognition is still at the embryonic stage of development. Therefore much additional work remains to be done to address the ongoing challenges. It is clear that developing good algorithms for solving the problem of action recognition would yield huge potential for a large number of potential applications, e.g., human-computer interaction, video surveillance, gesture recognition, robot learning and control, etc. In fact, the non-rigid nature of human body and clothes in video sequences resulting from drastic illumination changes, changing in pose, and erratic motion patterns presents the grand challenge to human detection and action recognition [1]. In addition, while the real-time performance is a major concern in computer vision, especially for embedded computer vision systems, the majority of state-of-the-art action recognition systems often employ sophisticated feature extraction and/or learning

techniques, creating a barrier to the real-time performance of these systems. This clearly suggests that there is a trade-off between accuracy and speed.

The structure of the remainder of the paper is as follows. Related work is discussed in Section 2. Next the architecture of the proposed methodology, its core components and the workflow are described in Section 3. In Section 4, experimental results are reported and compared with that of other competing techniques. Finally, Section 5 concludes the paper and outlines future work.

2 Related Literature

For the past decade or so, many papers have been published in the literature, proposing a variety of methods for human action recognition from video. Human action can generally be recognized using various visual cues such as motion [2–4] and shape [5, 6]. Scanning the literature, one notices that a large body of work in action recognition focuses on using keypoints and local feature descriptors [7–10]. The local features are extracted from the region around each keypoint. These features are then quantized to provide a discrete set of visual words before they are fed into the classification module. Another thread of research is concerned with analyzing patterns of motion to recognize human actions. For instance, in [3] the authors analyze the periodic structure of optical flow patterns for gait recognition. Likewise, in [4], periodic motions are detected and classified to recognize actions. Like us, some other researchers have opted to use both motion and shape cues. For example, in [11], Bobick and Davis use temporal templates, including motion-energy images and motion-history images to recognize human movement. In [12] the authors detect the similarity between video segments using a space-time correlation model. While Rodriguez *et al.* [13] present a template-based approach using a Maximum Average Correlation Height (MACH) filter to capture intra-class variabilities, Jhuang *et al.* [14] perform actions recognition by building a neurobiological model using spatio-temporal gradient. Additionally in [15], actions are recognized by training different SVM classifiers on the local features of shape and optical flow. In parallel, a great deal of work focuses on modeling and understanding human motions by constructing elaborated temporal dynamic models [16]. Finally, there is also a fertile and broadly influential area of research that uses generative topic models for modeling and recognizing action categories based on the so-called Bag-of-Words (BoW) model. The underlying concept of a BoW is that the video sequences are represented by counting the number of occurrences of descriptor prototypes, so-called visual words [17].

3 Proposed Methodology

This section includes the details of the proposed method developed for human action recognition. A schematic block diagram depicting the major components of the method is shown Fig. 1. As shown in the block diagram, the backgrounds are first subtracted from each video clip by using a Gaussian mixture background model to extract the silhouettes of the moving human body parts. For

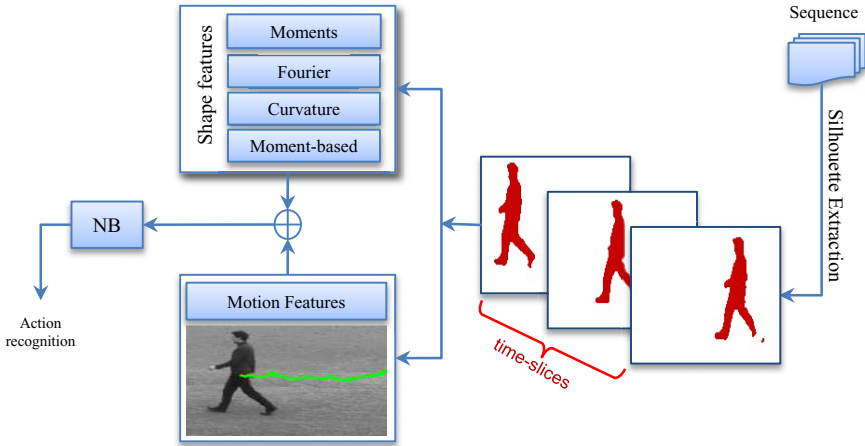


Fig. 1. Simplified schematic diagram of the proposed action recognizer

this method to be more robust against time warping effects, action snippets are temporally divided into a number of overlapping segments defined by fuzzy membership functions. Then local features are extracted from each temporal segment based on a variety of shape descriptors. As the motion features intuitively appear to be more relevant and appropriate to the current action recognition task, the final features fed into classifiers are constructed using both shape and motion features. These steps are detailed in the next subsections.

3.1 Preprocessing and Background Subtraction

For later successful feature extraction and classification, it is important to preprocess all video sequences to remove noisy, erroneous, and incomplete data, and to prepare the representative features that are suitable for knowledge generation. To wipe off noise and weaken image distortion, all frames of each action snippet are first smoothed by Gaussian convolution with a kernel of size 3×3 and variance $\sigma = 0.5$. For background subtraction, a GMM background model analogous to that described in [18] is used. In this model, each pixel in the scene is modeled by a mixture of K Gaussian distributions. Thus the probability that a certain pixel has intensity x_t at time t is given by

$$p(x_t) = \sum_{i=1}^K w_i * \eta(x_t; \mu_i, \Sigma_i) \tag{1}$$

where w_i, μ_i, Σ_i are the weight, the mean, and the covariance of the i -th distribution at time t respectively, and η is the Gaussian probability density function:

$$\eta(x_t; \mu, \Sigma) = \frac{1}{(2\pi)^{1/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x_t - \mu)^T \Sigma^{-1} (x_t - \mu)} \tag{2}$$

3.2 Feature Extraction

Feature extraction is generally viewed as the core in any action recognition system, but is also the most challenging and time-consuming part. A variety of features can be used for human action recognition. In this work, the features that primarily describe the shape of the segmented silhouettes of moving human body parts are used, in order to represent the action poses, which are the fundamental source of information regarding the interpretation of a specific human action. Moreover the information of motion can be also extracted by following the trajectory of the motion centroid, as described by end of this section. Before starting the feature extraction process, we temporally split each action snippet into several time-slices. These time-slices are defined by linguistic intervals. A Fuzzy membership function is used to describe each of these intervals,

$$f(t; \alpha, \beta, \gamma) = \frac{1}{1 + \left(\left|\frac{t-\alpha}{\beta}\right|\right)^\gamma} \quad (3)$$

where α , β , and γ are the center, width, and fuzzification factor of the interval, respectively as shown in Fig. 2. All the membership functions are chosen to be of identical shape on condition that their sum is equal to one at any instance of time t . It is experimentally found that using such fuzzy functions allows not only the local shape features to be extracted precisely, but also makes the performance decline resulting from time warping effects negligible. For shape features, we consider here a variety of invariant descriptors such as Fourier descriptors, curvature features, invariant shape moments, etc. The next subsections describe in more detail how such features are defined and extracted.

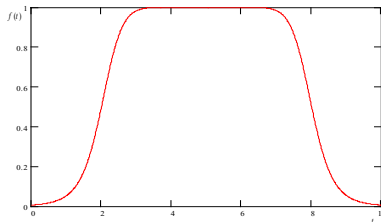


Fig. 2. An example of a membership function used to represent the temporal interval, with $\alpha = 5$, $\beta = 2$, and $\gamma = 10$

Fourier descriptors. Fourier descriptors for a given shape rely on the notion that any shape border (i.e., contour) can be represented by a periodic complex function: $z_i = x_i + jy_i$, where $x_i, y_i, i = 0, 1, \dots, n-1$ are the spatial coordinates of the boundary points. The k^{th} Fourier transform coefficient is calculated as

$$a_k = \frac{1}{n} \sum_{i=0}^{n-1} z_i \exp\left(-\frac{j2\pi ik}{n}\right), \quad k = 0, 1, \dots, n-1 \quad (4)$$

The Fourier descriptors are easily derived from the coefficients a_k by truncating the first two coefficients, a_0, a_1 and dividing the rest of the coefficients by $|a_1|$

$$b_k = \frac{|a_{k+2}|}{|a_1|}, k = 0, 1, \dots, n - 3 \tag{5}$$

It can be easily verified that such a choice of coefficients guarantees that the resulting shape descriptors are invariant to shape translation, rotation and scaling, and they are independent of the choice of the starting point on the contour.

Shape moments. Invariant moments are widely used in several computer vision applications for representing global and invariant shape characteristics of image features. The central moments of order $(p + q)$ of a shape $f(x, y)$ is defined by

$$\mu_{pq} = \iint (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \tag{6}$$

where (\bar{x}, \bar{y}) is the shape centroid. Thus the normalized central moments are given by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \gamma = \frac{p + q}{2} + 1 \tag{7}$$

On the basis of the normalized central moments, it is possible to calculate a set of seven moments [19], which is invariant under translation, changes in scale, and also rotation as follows,

$$\begin{aligned} h_1 &= \eta_{20} + \eta_{02} \\ h_2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \\ h_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{03} - \eta_{21})^2 \\ h_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \\ h_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] \\ h_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) \\ h_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] \\ &\quad + (\eta_{30} - 3\eta_{12})(\eta_{03} + \eta_{21})[3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] \end{aligned}$$

Curvature features. In a way similar to the extraction of the Mel Frequency Cepstral Coefficients (MFCC) features from voice signals, a set of other shape descriptors based on the cepstrum of the shape curvature can be also extracted. The mechanism works as follows. First we extract the shape curvature by using Freeman chain code [20]. Then, the cepstrum of the curvature signal is obtained, and the largest n coefficients are chosen to be added to the feature vector.

Moment-based features. Besides the pervious features, a set of other features derived from the central moments, can also be added into the feature vector. The existent analogy between moments and mechanical moments provides a deeper understanding of the central moments of second order, μ_{11}, μ_{02} and μ_{20} . Such

three central moments construct the components of the inertial tensor of the object rotation about the center of mass:

$$\mathcal{J} = \begin{bmatrix} \mu_{20} & -\mu_{11} \\ -\mu_{11} & \mu_{02} \end{bmatrix} \tag{8}$$

Based on the inertial tensor analogy, several further features could be derived from the 2nd order central moments. For example, the main inertial axis could be obtained by calculating the eigenvalues of the inertial tensor as follows

$$\lambda_{1,2} = \sqrt{\frac{1}{2}(\mu_{02} + \mu_{20}) \pm (4\mu_{11}^2 - (\mu_{02} - \mu_{20})^2)^{1/2}} \tag{9}$$

Notably the main inertial axes correspond to the semi-major a and semi-minor axes b of the ellipse, which can be regarded as an approximation of the object.

The orientation of the object defined as the tilt angle between the x -axes and the axis, around which the object can be rotated with minimal inertia, can be calculated by

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \tag{10}$$

where θ is assumed to be the angle between the x -axis and the semi-major axis a and the principal value of the *arc tangent* is picked such that $-\frac{\pi}{2} \leq \arctan x \leq \frac{\pi}{2}$. Other parameters such as the roundness κ and eccentricity ε that give information about the shape roundness seem to be very close. The roundness κ of a shape can easily be obtained by dividing the square of the perimeter p with the area of the object A . As for the simple geometric fact that the circle has the maximum area for a given perimeter p , κ can be scaled as follows

$$\kappa = \frac{p^2}{4\pi A} \tag{11}$$

Hence κ for a circle is equal 1, for other objects > 1 . The eccentricity ε can be calculated from the second-order central moments by

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 - 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2} \tag{12}$$

All the feature vectors of an action snippet are then normalized to fit a zero-mean and a unit variance distribution. The normalized vectors obtained can be used as shape contextual information for classification and matching. Many approaches in various object recognition applications directly combine these vectors to get one final vector per video and classify it using any classification algorithm. It is worth mentioning that concatenating all the feature vectors extracted from all frames of an action snippet will result in a very large feature vector that might be less likely to be classified correctly. To resolve this problem and to reduce the dimensionality of the resulting vector, all feature vectors of an action snippet at a time-slice are weighted and averaged

$$\boldsymbol{\mu} = \frac{1}{\tau} \sum_{t=1}^{\tau} w_t \mathbf{x}_t \tag{13}$$

where $w_t = f(t; \alpha, \beta, \gamma)$ is the weighting factor and τ is the number of the feature vectors at the time-slice. Then all the vectors resulting at each of the time-slices are catenated to yield the final feature vector for a specific action snippet.

3.3 Action Classification

In this section, we formulate the action recognition task as a multi-class learning problem, where there is one class for each action, and the goal is to assign an action to an individual in each video sequence. There are various supervised learning algorithms by which an action recognizer can be trained. Naïve Bayesian (NB) classifier is used in our framework. The main advantage of the NB classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. In spite of their naive design and apparently over-simplified assumptions, NB classifiers have shown to work quite well in many complex real-world situations [21].

Given a final feature vector X extracted from a test action snippet, posteriori probabilities are calculated using training action snippets. This is accomplished using Bayes rule, which is a fundamental formula in decision theory. Mathematically speaking, it is expressed as

$$p(\omega_i|X) = \frac{p(X|\omega_i)p(\omega_i)}{p(X)} \quad (14)$$

where $p(X) = \sum_{i=1}^K p(X|\omega_i)p(\omega_i)$. $p(\omega_i|X)$ is the posteriori probability of observing the class ω_i given the feature vector X . $p(\omega_i)$ is the priori probability of observing the class ω_i , $p(X|\omega_i)$ is the conditional density and K is the total number of classes. For this recognition task, it is assumed that each action snippet is uniquely described by the value of its a posteriori probability. Further, all the priori probabilities are assumed to be equal, and thus find the density functions for each of the classes, where each class refers to an action. Thus, K such densities are found, corresponding to the K different actions. Having obtained these K values for each of the classes, the most likely action is given by

$$P = \max[p_1, p_2, \dots, p_K] \quad (15)$$

where P is the probability of the most likely class and p_1, p_2, \dots, p_K are the probabilities of K different actions.

4 Experimental Results

In this section, the proposed method is experimentally evaluated using KTH dataset [8]. To assess the reliability of the method, the results obtained are compared with those reported in the literature for action recognition [10, 22, 14, 23, 9, 13, 24]. A total of about 2391 sequences are involved in the KTH dataset, which include six types of human actions (i.e., walking, jogging, running, boxing, hand waving and hand clapping). The actions are performed by

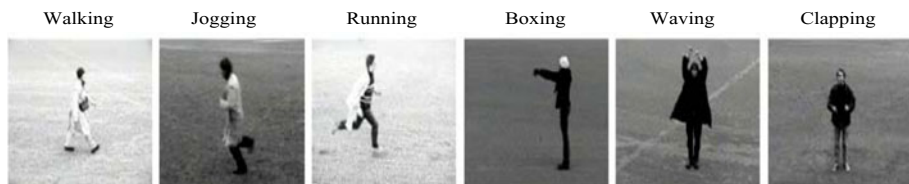


Fig. 3. Examples of the KTH dataset actions used in the validation process

a total of 25 individuals in four different settings (i.e., outdoors, outdoors with scale variation, outdoors with different clothes, and indoors). All sequences were acquired by a static camera at 25fps and a spatial resolution of 160×120 pixels over homogeneous backgrounds. There are, to the best of our knowledge, only very few similar datasets already available in the literature with sequences acquired on different environments. Fig. 3 shows an example frame for each action from the KTH dataset. In order to prepare the simulation and to provide an unbiased estimation of the generalization abilities of the classification process, the sequences, for each action, were divided with respect to the subjects into a training set and a test set with approximately equal numbers. This was done such that both sets contained actions from all persons. The NB classifier was trained on the training set while the evaluation of the recognition performance was performed on the test set. The confusion matrix depicting the results of action recognition obtained by the proposed method is shown in Table 1. From the figures in Table 1 a number of points can be drawn. The majority of actions are correctly classified. Additionally, there is a clear distinction between arm actions and leg actions. Most of the mistakes where confusions occur are between "jogging" and "running" actions and between "boxing" and "clapping" actions. This is intuitively plausible due to the fact of high similarity between each pair of these actions. To assess the efficiency of the proposed method, the results obtained have been compared with those of other previously published studies in the literature as shown in Table 2. From this comparison, it turns out that our method performs competitively with other state-of-the-art methods and its results compare favorably with previously published results. Notably all the methods that we compared our method with have used similar experimental setups, thus the comparison seems to be meaningful. Finally, the proposed

Table 1. Confusion matrix of the proposed method

ACTION	walking	jogging	running	boxing	waving	clapping
walking	0.99	0.01	0.00	0.00	0.00	0.00
jogging	0.07	0.80	0.13	0.00	0.00	0.00
running	0.01	0.08	0.91	0.00	0.00	0.00
boxing	0.00	0.00	0.00	0.94	0.01	0.05
waving	0.00	0.00	0.00	0.00	0.99	0.01
clapping	0.00	0.00	0.00	0.02	0.00	0.98

Table 2. Comparison with some well-known studies in the literature

Method	Accuracy
Our method	93.5%
Liu <i>et al.</i> [10]	92.8%
Wang <i>et al.</i> [22]	92.5%
Jhuang <i>et al.</i> [14]	91.7%
Rapantzikos <i>et al.</i> [23]	88.3%
Dollár <i>et al.</i> [9]	81.2%
Rodriguez <i>et al.</i> [13]	88.6%
Ke <i>et al.</i> [24]	63.0%

action recognizer runs at 25fps on average (using a 2.8 GHz Intel dual core machine).

5 Conclusion and Future Work

This paper has introduced an approach to human activity recognition using multiple cues. Although our model might seem to be similar to previous models of visual recognition, it differs in two important aspects resulting in an improved performance. First temporal shape contextual information in this model is obtained using a variety of descriptors, both border-based and region-based. Secondly, partitioning action snippets into several time-slices in a fuzzy manner makes the model more robust to shape deformations and time wrapping effects. The results obtained are at least as encouraging as those obtained through much more sophisticated and computationally complex methods. Furthermore the method achieves real-time performance and thus it can provide timing guarantees to real-time applications. However it would be advantageous to explore the empirical validation of the method on more complex realistic datasets presenting many technical challenges in data handling. Such issues are crucial and will be addressed in the scope of our future work.

Acknowledgment

This work is supported by Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by DFG and Forschungspraemie (BMBF-Fröderung, FKZ: 03FPB00213). The financial support of the Egyptian government for the PhD scholarship of the first author is also gratefully acknowledged.

References

1. Chakraborty, B., Bagdanov, A.D., González, J.: Towards real-time human action recognition. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 425–432. Springer, Heidelberg (2009)

2. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proc. ACM Int. Conf. Image and Video Retrieval, vol. 8, pp. 494–501 (2007)
3. Little, L., Boyd, J.E.: Recognizing people by their gait: The shape of motion. *International Journal of Computer Vision* 1, 1–32 (1998)
4. Cutler, R., Davis, L.S.: Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on PAMI* 22, 781–796 (2000)
5. Thurán, C., Hlaváč, V.: Pose primitive based human action recognition in videos or still images. In: *IEEE Computer Society Conference on CVPR* (2008)
6. Lu, W.L., Okuma, K., Little, J.J.: Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing* 27, 189–205 (2009)
7. Sadek, S., Al-Hamadi, A., Michaelis, B., Sayed, U.: Toward robust action retrieval in video. In: Proc. of BMVC 2010 (2010)
8. Laptev, I., Pérez, P.: Retrieving actions in movies. In: Proc. ICCV, pp. 1–8 (2007)
9. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. IEEE Workshop on VS-PETS, pp. 65–72 (2005)
10. Liu, J., Shah, M.: Learning human actions via information maximization. In: *IEEE Int. Conference on Computer Vision and Pattern Recognition* (2008)
11. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *IEEE Transactions on PAMI* 23, 257–267 (2001)
12. Shechtman, E., Irani, M.: Space-time behavior based correlation. *Computer Vision and Pattern Recognition* 1, 405–412 (2005)
13. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In: Proc. CVPR (2008)
14. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: *IEEE International Conference on Computer Vision*, pp. 257–267 (2007)
15. Schindler, K., Van Gool, L.: Action snippets: How many frames does action recognition require? In: Proc. CVPR, pp. 1–8 (2008)
16. Laxton, B., Lim, J., Kriegman, D.: Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In: Proc. CVPR, pp. 1–8 (2007)
17. Blei, D.M., Lafferty, J.D.: Correlated topic models. *Advances in Neural Information Processing Systems (NIPS)* 18, 147–154 (2006)
18. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR, pp. 246–2527 (1999)
19. Hu, M.: Visual pattern recognition by moment invariants. *IRE Tr. on. Inf. Theory*, 179–187 (1962)
20. Alajlan, N., Kamel, M.S., Freeman, G.: Multi-object image retrieval based on shape and topology. *Signal Processing: Image Communication* 21, 904–918 (2006)
21. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning* 29, 103–137 (1997)
22. Wang, Y., Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: CVPR (2009)
23. Rapantzikos, K., Avrithis, Y., Kollias, S.: Dense saliency-based spatiotemporal feature points for action recognition. In: CVPR, pp. 1–8 (2009)
24. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: Proc. ICCV, vol. 1, pp. 166–173 (2005)

A Platform for Monitoring Aspects of Human Presence in Real-Time

X. Zabulis¹, T. Sarmis¹, K. Tzevanidis^{1,2}, P. Koutlemanis¹,
D. Grammenos¹, and A.A. Argyros^{1,2}

¹ Institute of Computer Science - FORTH Herakleion, Crete, Greece

² Department of Computer Science, University of Crete

Abstract. In this paper, the design and implementation of a hardware/software platform for parallel and distributed multiview vision processing is presented. The platform is focused at supporting the monitoring of human presence in indoor environments. Its architecture is focused at increased throughput through process pipelining as well as at reducing communication costs and hardware requirements. Using this platform, we present efficient implementations of basic visual processes such as person tracking, textured visual hull computation and head pose estimation. Using the proposed platform multiview visual operations can be combined and third-party ones integrated, to ultimately facilitate the development of interactive applications that employ visual input. Computational performance is benchmarked comparatively to state of the art and the efficacy of the approach is qualitatively assessed in the context of already developed applications related to interactive environments.

1 Introduction

Advances in several research and technological fields have increased the likelihood of creating interactive environments that adapt to various aspects of human presence. Towards such environments, vision based estimation of the location and geometric attributes of the human body is of interest, as it unobtrusively conveys information on user activities. Multiview scene observation enables accurate and robust 3D reasoning, particularly in environments that are imaged distantly and at which occlusions are frequent and spatially extended. The large volume of generated data combined with high framerate requirements, calls for distributed and parallel image acquisition and processing, as well as efficient communication strategies. A multiview platform is proposed in this paper, whose architecture aims at the reduction of computational and communication costs. The platform provides functionalities for synchronized and distributed image acquisition and visual processing. Moreover, its complexity is encapsulated utilizing a middleware infrastructure so that the output of multiview visual computation can be communicated to third-party applications.

Upon this platform, a set of key visual processes are implemented that estimate aspects of human presence in real time, such as the volumetric occupancy of the monitored environment, the textured visual hull of persons, their

location and their head pose. The corresponding implementations capitalize on the proposed platform to pipeline processes and reduce communication costs. By encapsulating the platform functionalities implemented in a middleware infrastructure, pilot applications can be developed in simple programming environments, transparently to the implementation of the platform.

The remainder of this paper is organized as follows. In Sec. 2, related work is reviewed. Sec. 3 presents the setup and architecture of the proposed platform. The implementation of key components of the system is described in Sec. 4. In Sec. 5, the performance of the proposed platform is benchmarked and qualitatively assessed through pilot applications. Sec. 6 summarizes contributions and provides directions for future research.

2 Related Work

Although the use of camera clusters has been increasing, there exist only a few platforms that facilitate the development, pipelining and integration of parallel and distributed visual processes. Even fewer are the platforms that focus on estimating person locations and geometric attributes of the human body. Some multi-camera platforms gather and display input from multiple video streams, and either process each stream individually (e.g. [1-3]), or perform simple multiview operations (i.e. stream mosaicing [2]). Such platforms are typically based on centralized architectures, as they exhibit moderate computational requirements.

Multiview computations require significantly more the computational power raising the need for parallel computation. Parallel and distributed platforms (i.e. [4]) have been utilized to reconstruct a generic scene from multiple stereo views computed in parallel. More relevant to this work are systems [5-7] that reconstruct persons in the scene through their volumetric occupancy or the visual hull [8]. Although the reconstruction is approximate, i.e. concavities are not represented, the approach is sufficient for tasks such as person detection and localization. The systems in [6, 7] compute the visual hull volumetrically, enabling massive parallelization of computation and direct application of 3D linear operations (i.e. noise filtering) on the reconstructed volume. In contrast, [5] proposes view-based parallelization, based on the silhouette extracted from each view, and results in a mesh whose nodes are irregularly arranged in 3D space. Parallelization is massive in [6, 7] which run on GPU(s), while [5] parallelizes computation in dual-core CPUs and leads to increased hardware requirements.

3 Software Platform

3.1 System Setup

A typical physical setup of the system involves a $6 \times 6 \times 2.5m^3$ room, including a large (i.e. $5 \times 2m^2$) backprojection display providing visual feedback. The

camera cluster consists of 8 *Dragonfly2 Point Gray* cameras, mounted near the ceiling of the room viewing it peripherally and employs 2 or 4 computers with an Intel *i920* quad-core CPU and an *NVIDIA GTX275* GPU each. Cameras are evenly distributed to *host* computers and have a maximum framerate of 30 *fps* at 1280×960 image resolution. Synchronized image acquisition uses an additional *FireWire* bus across computers and timestamps, guaranteeing a maximum of $125\mu\text{sec}$ temporal discrepancy among images with the same timestamp. Henceforth, the set of the N simultaneously acquired images I_i is referred as a *multiframe*, the projection matrices for each view $i \in [1, 2, 3, \dots, N]$ denoted as P_i and corresponding camera centers as κ_i . The computers in each system are connected by 1 *GB* Ethernet in a star network topology, where one computer is declared the *central* workstation and the rest as *satellites*.

Cameras are automatically calibrated intrinsically and extrinsically, employing a checkerboard detector [9] to find reference points and passing them to a standard calibration toolbox [10]. Corresponding reference points across views are used to increase calibration accuracy, via bundle adjustment [11].

3.2 Architecture

The complexity of camera control and synchronized image acquisition is encapsulated in a software platform. The platform supports the synchronized communication of images and intermediate computation results across processing nodes through a shared memory. Results of visual computations become available to applications via integration with a middleware infrastructure.

A broad spectrum of camera types is supported, connected to host computers by Direct Memory Access to RAM. Each host workstation maintains a fixed RAM buffer for every view in which it stores captured frames after converting them from Bayer Tile to RGB format and rectifying them for lens distortion. The operations are implemented in the GPU of host computers with image storage rate matching the camera framerate. Images are stored together with associated timestamps and, as new frames arrive, older ones are removed.

Each time a new image enters a buffer, its timestamp is notified to the central workstation. During the creation of a multiframe, the central workstation selects the appropriate timestamps for each buffer, local or remote. Then, it broadcasts timestamp queries to the satellite workstations and acquires the queried frames, while for local buffers it fetches the frames from its RAM. This way, a frame that is dropped at a view does not disturb synchronization and, also, the transmission of a frame for a multiframe that will be eventually rejected is avoided.

Both images and intermediate computational results can be stored in a shared memory at the central workstation where multiple processes may have simultaneous access. Processes can concurrently read the data from the shared memory without copying them to the process' address space. By adding another computer, the central workstation can be relieved from image acquisition and preprocessing.

The platform is further integrated with a middleware infrastructure to facilitate the development of new visual processes through an API that supports

the *C/C++*, *.NET*, *Java*, *Python*, *Delphi*, and *Flash/ActionScript* programming languages. Through this middleware, the output of such visual processes (i.e. those in Sec. 4), can be communicated to applications in the form of event notifications, hiding the details of network connections and data serialization. The same middleware is also employed in the control of actuating components of the environment such as displays, illumination and sound. These capabilities simplify the development of interactive applications and enable the integration of vision processes with the reasoning and actuating components of the environment.

4 Key Vision Processes

Four processes that compute basic aspects of human presence have been implemented. These constitute the core visual processes for developing pertinent interactive applications. The processes compute the volumetric occupancy of persons in the images environment, their textured visual hulls, their locations and motion trajectories and their 3D head poses. To meet framerate requirements, these implementations are designed to be distributed and massively parallel.

4.1 Volumetric Occupancy

Volumetric occupancy of the imaged scene is represented in a voxel grid V . Along with image rectification, background subtraction of images is performed locally on the GPU of host computers. A pixelwise background subtraction [12] is employed to parallelize the operation. In contrast to [6, 7] we do not perform morphological operations to compensate for background subtraction errors, but consider them in reconstruction. The results, images B_i , are transmitted in Run Length (RL) encoding to shared memory.

The scene is reconstructed on the GPU of the central workstation by assigning a thread per voxel \mathbf{v} . A voxel $V(\mathbf{v})$ is labeled as 1 if found to be occupied and 0 if not. To process volumes of arbitrary size, computation of V is partitioned and results concatenated in shared memory, as opposed to [6, 7] where the dimensions of V are constrained by the GPU's memory capacity.

The value at $V(\mathbf{v})$ is computed amongst the views $i' \in C \subseteq [1, 2, 3, \dots, N]$, that \mathbf{v} projects within their visual field. Ideally, occupied voxels should project in foreground regions in *all* $B_{i'}$ and vice versa. To compensate for errors in background subtraction, a more lenient rule is applied, and a voxel is considered as occupied if at least half of the views in which it is visible concur that it projects in a foreground region. That is, $V(\mathbf{v})$ is 1 if

$$\sum_{i'} (B_{i'}(P_{i'} \cdot [\mathbf{v}; 1]^T)) > \text{card}(C)/2 \quad (1)$$

and 0 otherwise. Fig. 1(left), illustrates the process for a challenging scene. Optionally, V can be filtered with a 3D kernel to suppress voxelization artifacts; see Fig. 1(right).

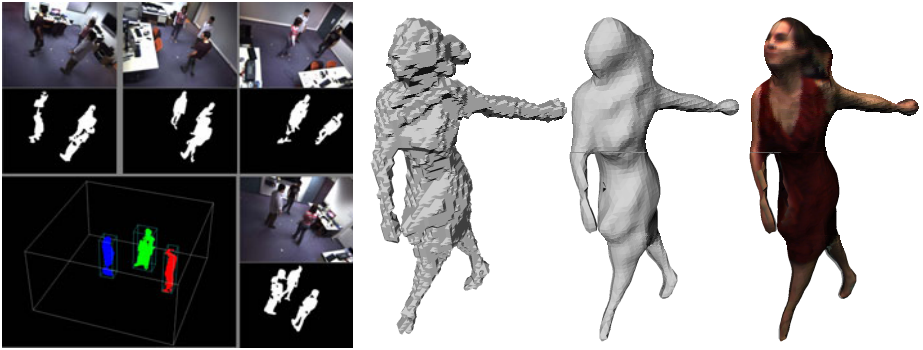


Fig. 1. Left: Original images, background subtraction and volumetric reconstruction, for a scene imaged by 4 cameras. Persons are imaged against cluttered background and uneven illumination, resulting in inaccurate background subtraction. Persons occlude each other in all views. Reconstruction is not accurate, but sufficient for person tracking. Right: Visual hull (left), smooth visual hull (middle) and textured visual hull (right), for a benchmark dataset [13].

4.2 Person Localization and Tracking

A multiview approach is adopted for person localization as such approaches typically outperform single-view [14, 15], due to the systematic treatment of occlusions. As in [16–20], we employ a planar homography constraint to map imaged persons to the ground plane, but we consider the occupied volume instead of the projection area. As in [19], we also utilize volumetric occupancy to increase the localization robustness, but do not require that the number of tracked persons is a priori known.

A GPU process projects V on the ground plane and sums occupied voxels along the direction perpendicular to the floor. This results in a 2D buffer F , registered to the ground plane. Image F is transferred to shared memory where it is collected by a tracker [21] that establishes temporal correspondence of person locations. In F , persons appear as intensity blobs, which are tracked only if they exhibit a sufficient amount of volume, as measured by their intensity sum in F . New persons are detected as new blobs that exhibit a temporal persistence over a few frames. The tracker is implemented in CPU and modified to track intensity blobs, rather than skin-colored blobs in color images for which it was originally formulated. It is robust to transient localization failures and, most importantly, designed to retain the tracking of blobs even if those appear merged for long temporal intervals. This way, tracking succeeds even if subjects are close to each other forming a single connected component in V (see Fig. 2).

Tracking robustness is supported by the high frame rate ($> 10\text{ Hz}$) of operation and, also, by fine granularity of volumetric representation (1cm^3). High framerate casts blob motion in F smooth and simpler to track. Fine granularity increases the precision of blob localization in F . To conserve communication cost, middleware events are created upon change of person location, or when persons enter and leave the scene.

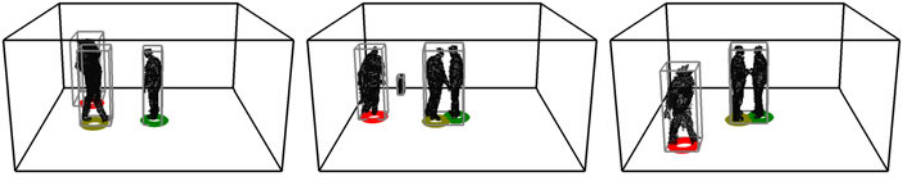


Fig. 2. Person localization and tracking. Estimated person location is marked with colored circles. Tracking is successful although in some frames visual hulls are merged. Occurring transiently, the spurious structure in frame 2 is disregarded by the tracker.

4.3 Textured Visual Hull

Collecting the output of volumetric occupancy from shared memory (Sec. 4.1), a GPU process estimates the visual hull of persons and, optionally, textures it. The hull is computed as the 0-isosurface in V , by a parallel implementation of the “Marching Cubes” algorithm [22]. To benefit from operations optimized on GPU hardware, V is treated as a 3D texture. A thread is assigned to each voxel and accesses V ’s values at the locations of mesh vertices by interpolating the 3D texture at the corresponding locations; further details can be found in [23]. The isosurface represents the visual hull and is encoded as a mesh M of triangles in shared memory.

Texture mapping on each triangle j of M also makes use of GPU-optimized operations. Initially, the views in which j is visible are identified, by employing a depth buffer Z_i for each view i . Each pixel in Z_i encodes the distance of κ_i to the surface that is imaged at that pixel. Buffer Z_i is computed by calculating the distance $\delta_{ij} = |\tau_j - \kappa_i|$ for each triangle, where τ_j is the triangle’s 3D centroid. Triangles are projected on Z_i and the minimum distance that is imaged in each pixel of Z_i is assigned to that pixel. Let Δ the length of a voxel’s side. Then,

$$|\delta_{ij} - Z_i(P_i \cdot [\tau_j; 1]^T)| < \Delta, \quad (2)$$

is a criterion that indicates if triangle j is indeed imaged at location of view i ; (2) is false, if triangle j is occluded in view i . Threshold Δ is sufficient as M ’s triangles are contained within voxel size. This criterion also facilitates parallelization since, otherwise, the sequential maintenance of the list of triangles imaged at $Z_i(P_i \cdot [\tau_j; 1]^T)$ would be required to cope with pixels imaging multiple triangles along the images of M ’s vertices. Aiming at efficiency, the number of considered triangles is reduced by disregarding those whose normal forms an angle greater than $\pi/2$ with the optical axis of view i . Texture coordinates of triangle nodes, $P_i \cdot [\tau_j; 1]^T$, have been already computed during the evaluation of (2) and are retrieved instead of recomputed.

To resolve multi-texturing conflicts in triangles visible in multiple views, textures are blended according to a weighting factor proportional to the size of the projected area, on a pixel shader of the GPU, so that distal and oblique views are weighted less. Further details can be found in [23]. Figure 1(right) visualizes the obtained results.

4.4 Head Pose Estimation

Head pose estimation provides information about the direction at which a person is facing at. The task is challenging in wide areas, because faces are imaged in poor resolution and are often occluded. The multiview head pose estimation method in [24] is parallelized and employed to provide an estimate of 3D head center location \mathbf{c} and the 3 rotational pose components (*pitch*, *yaw*, *roll*).

The method exhibits increased accuracy in the context of distant viewing, over other head pose estimation methods that fuse single-view pose estimates [25–27] and yield only 2 pose components. It method utilizes the textured visual hull M to collect all available facial texture fragments and resolve occlusions, which is received as input from the module of Sec. 4.3.

An instance of this method is applied independently on each person detected by the module in Sec. 4.2. For each person, head center \mathbf{c} is tracked by a variant of the Mean Shift [28] algorithm, using a 3D spherical kernel S matched to the part of M that reconstructs the persons head. The system broadcasts \mathbf{c} to host workstations which perform concurrently face detection [29] within the areas α_i of I_i where S projects. Per-view orientation estimates (*pitch*, *yaw*) are obtained as $\mathbf{o}_k = \mathbf{f}_k - \mathbf{c}_k$, where k enumerates the views where a face was detected in α_k ; \mathbf{f}_k is the intersection of M with the ray from camera k through the face’s detection in α_k . Estimates \mathbf{o}_k are fused at the central workstation into a median vector \mathbf{o}_c .

The texture of the visual hull is then projected on S , with \mathbf{c} being the center of projection to form a spherical image I_s , an operation optimized on the GPU as texture mapping. By construction, exactly one frontal face occurs in I_s and, thus, a generic CPU-based frontal face detector [29] suffices for its detection. To form I_s , only areas α_i of I_i are transmitted to the central workstation. Orientation \mathbf{o} is provided by the face center $\mathbf{p} \in I_s$ using a look up table to find its 3D correspondence on S . The orientation γ of the face in I_s provides the *roll*

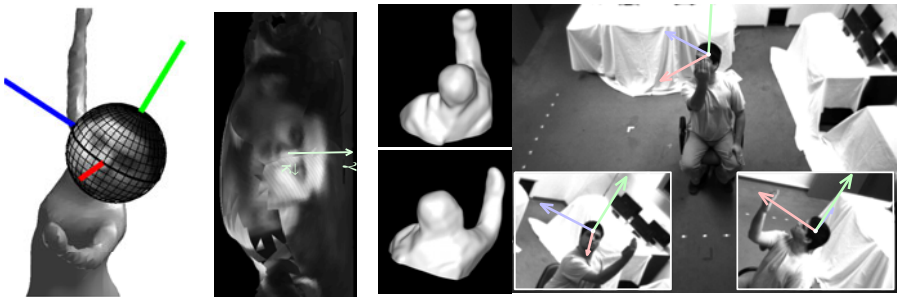


Fig. 3. 3D head pose estimation. On the left, shown is the visual hull with facial texture projected on S , for a benchmark dataset [13]. To its right the generated spherical image I_s is shown, with a the green vector indicating the face center and its 2D orientation in I_s . In the right panel, shown are two views of the visual hull of a subject during our experiments and the estimation result superimposed on some of the images I_i .

component of 3D pose and is optionally computed by optimizing a correlation-based symmetry operator in the $[0, \pi)$ range. The availability of the precomputed estimate \mathbf{o}_c simplifies the above process as follows. S is in-place rotated by R , $R \cdot [100]^T = \mathbf{o}_c$, so that the projected face occurs approximately (a) at the equator of S , where spherical distortions are minimized and (b) at the center of I_s thus reducing the area were a face is searched for.

The output, head center \mathbf{c} and 3 pose angles, is communicated through middleware and associated to the tracking id of the person as this has been derived in Sec. 4.2. Representative head pose estimation results are illustrated in Fig. 3.

5 Experiments

5.1 Computational Performance

The evaluated system consists of 8 cameras, using 1, 2 or 4 computers and full or lower resolution versions of I_i . Columns $a - f$ in Table 1 benchmark the performance of the proposed implementations and compare it with pertinent methods (left three columns), for *visual hull* computation, for the same amount of voxels. In columns $T1$ and $T2$ the performance of computing the *textured visual hull* is reported. The 1st row marks the achieved framerate, the 2nd the amount of computational power utilized, the 3rd the number of voxels in V , the 4th the resolution of I_i s, and the 5th the number of computers employed. Analytical performance measurements in more conditions and for intermediate operations (i.e. lens distortion compensation, background subtraction) can be found in [23]. The latency between a person’s motion and the reception of the corresponding event is $\approx 140\text{ ms}$ and localization accuracy is $\approx 4\text{ cm}$.

Table 1. Performance measurements and comparison (see text)

	[6]	[7]	[30]	a	b	c	d	e	f	T1	T2
<i>Hz</i>	25	14	30	23.8	34	40	98.1	71.4	103.2	25.3	24.1
<i>GFLOPS</i>	1614	933	836	437	437	894	894	1788	1788	1788	1788
<i>voxels</i>	2^{11}	2^{11}	2^{11}	2^{11}	2^{11}	2^{11}	2^{11}	2^{11}	2^{11}	2^{24}	2^{24}
<i>pixels</i>	5×2^7	5×2^7	5×2^7	5×2^7	5×2^6	5×2^7	5×2^6	5×2^7	5×2^6	5×2^7	5×2^8
<i>computers</i>	5	1	11	1	1	2	2	4	4	4	4

In [6], the computation of V is distributed in the GPUs of multiple computers. For each \mathbf{v} , a partial estimate of $V(\mathbf{v})$ is computed and transmitted centrally, to be fused with the rest of estimates. Albeit the RL compression, communication overhead is significant as it corresponds to N times the number of voxels in V . The increased performance of the proposed method stems both from the parallelization of the background subtraction stage and, most importantly, from the transmission of (RL encoded) images B_i that requires significantly less capacity. In [5], a minimal communication cost is obtained, by transmitting only the silhouettes in B_i s, but then only per-view parallelization is achieved, as opposed to

per-voxel. The system in [7] eliminates this communication cost by centralizing all computation in one computer. This solution does not scale well with the number of views (the online version is limited to four). In contrast to [6], increasing granularity of V does not increase communication cost, as transmission cost of images B_i is constant.

Regarding head pose estimation, coarse orientation estimation \mathbf{o}_c runs at $\approx 10\text{ Hz}$ and drops to $\approx 2\text{ Hz}$ when estimating 3D pose to the precision of 1° , for 8 views in 2 computers. Accuracy experiments replicate the $\approx 3^\circ$ accuracy reported in [24] for I_s of 1280×960 resolution, but accuracy drops to $\approx 5^\circ$ when I_i are 640×480 . The bottleneck of the whole process is face detection which is implemented in the CPU. To parallelize it, the task is distributed to the computers hosting the cameras.

Overall, results indicate improvement of state of the art in performance, efficient use of computational resources and linear scaling of computational demands with respect to reconstructed volume and number of views.

5.2 Pilot Applications

Pilot applications were developed to evaluate (a) the proposed system in interactive scenarios and (b) the development process based on the proposed platform. Interactivity is supported by audiovisual feedback, provided by wall mounted displays and a surround audio system. Applications are implemented in *Flash Action Script* and communicate with the platform through the middleware.

In a cultural heritage application, a fresco is projected on a wall of the room. Visitors observing the projection are localized and their position in front of the projection is tracked (Fig. 4, bottom). Depending on current and previous user locations, the display is augmented with visual and textual information in the language of each individual user. The application utilizes person localization events to implement the interaction scenario, which also incorporates contextual environmental constraints, i.e. further information about the fresco region is provided if a user re-visits a particular location. More details on the above scenario can be found in [31].

Using the middleware layer, the visual processes in Sec. 4 can be customized. For example, in a gaming application called *footprints*, as players move around, the contact area of their feet with the floor is used to virtually paint the floor. The method in Sec. 4.2 is employed to determine user location and then, the method in Sec. 4.1 is re-invoked at higher voxel tessellation only at the corresponding volumes to reconstruct the volume around the feet of the user. The increased reconstruction accuracy is also employed in *walkman*, where alternating black and white areas on the floor represent piano keys and players can play music by stepping on them. Similarly, the cultural application above has been extended to collaborate with the method in Sec. 4.4 to determine the fresco region that each user is facing at. Using the result of person localization, only the volume around the person's head is reconstructed at high resolution.

Pilot applications were evaluated and showcased to more than 100 persons, which exhibited diversity in age, gender, cultural and professional background



Fig. 4. Pilot applications. Top: The display is updated based on the location and walk-through trajectories of visitors. Bottom-left: footprint extraction; obtained footprints over 20 frames are shown superimposed. Bottom-right: the interactive game's interface is projected on the wall providing visual feedback to the users, as they play music by stepping in appropriate spots.

(see [31] for more details on this usability evaluation). The overall impression was that due to brisk system response the applications are considered as exciting and engaging and, thus, system response was adequate towards supporting the aforementioned interactive applications.

6 Conclusions

This paper presented a multiview platform that facilitates the development and integration of parallel and distributed visual processes. On top of this platform, basic visual processes have been efficiently implemented. The functionalities of the platform become accessible via an integrating middleware layer that communicates high-level visual computation results to the application layer. The efficacy of the proposed implementation and architecture is assessed by the development, in a simple prototyping language, of pilot applications that utilize the developed infrastructure. The proposed system is characterized by increased robustness in tracking persons at high framerate, and its reduced requirements in computational hardware. The requirements are linearly related to the volume of required computation, or otherwise, the spatial extent of the area to be

covered and the number of views. System architecture adapts to the availability of resources, few or abundant, and system performance scales linearly with respect to them.

Future work regards the adoption of GPU implementation of the face detection process (i.e. [32]), the adoption of occlusion maps so that room furniture can be accounted in visibility computation [7]. Additionally, we plan to expand the set of available visual processes (e.g., by implementing gesture recognition) in order to enrich the repertoire of interaction capabilities and facilitate the development of more elaborate applications over the proposed platform.

Acknowledgements

This work was partially supported by the FORTH-ICS internal RTD Programme “Ambient Intelligence and Smart Environments”.

References


1. Ramachandran, U., Nikhil, R., Rehg, J., Angelov, Y., Paul, A., Adhikari, S., Mackenzie, K., Harel, N., Knobe, K.: Stampede: a cluster programming middleware for interactive stream-oriented applications. *IEEE Trans. Parallel and Distributed Systems* 14, 1140–1154 (2003)
2. Gualdi, G., Prati, A., Cucchiara, R., Ardizzone, E., Cascia, M.L., Presti, L.L., Morana, M.: Enabling technologies on hybrid camera networks for behavioral analysis of unattended indoor environments and their surroundings. In: *ACM Multimedia Workshops*, pp. 101–108 (2008)
3. Chen, P., Ahammad, P., Boyer, C., Huang, S., Lin, L., Lobaton, E., Meingast, M., Oh, S., Wang, S., Yan, P., Yang, A., Yeo, C., Chang, L., Tygar, J., Sastry, S.: CITRIC: A low-bandwidth wireless camera network platform. In: *ACM/IEEE Int. Conference on Distributed Smart Cameras*, pp. 1–10 (2008)
4. Jung, S.H., Bajcsy, R.: A framework for constructing real-time immersive environments for training physical activities. *Journal of Multimedia* 1, 9–17 (2006)
5. Allard, J., Franco, J., Menier, C., Boyer, E., Raffin, B.: The Grimage platform: A mixed reality environment for interactions. In: *ICCVS* (2006)
6. Ladikos, A., Benhimane, S., Navab, N.: Efficient visual hull computation for real-time 3d reconstruction using CUDA. In: *CVPR Workshops*, pp. 1–8 (2008)
7. Schick, A., Stiefelhagen, R.: Real-time GPU-based voxel carving with systematic occlusion handling. In: Denzler, J., Notni, G., Süße, H. (eds.) *DAGM 2009*. LNCS, vol. 5748, pp. 372–381. Springer, Heidelberg (2009)
8. Laurentini, A.: The visual hull concept for silhouette-based image understanding. *PAMI* 16, 150–162 (1994)
9. Sarmis, T., Zabulis, X., Argyros, A.A.: A checkerboard detection utility for intrinsic and extrinsic camera cluster calibration. Technical Report TR-397 (2009)
10. Bouguet, J.Y.: Camera calibration toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc
11. Lourakis, M., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software* 36 (2009)
12. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: *International Conference on Pattern Recognition*, pp. 28–31 (2004)

13. INRIA Perception Group, <http://4drepository.inrialpes.fr/>
14. Tran, S., Lin, Z., Harwood, D., Davis, L.: UMD_VDT, an integration of detection and tracking methods for multiple human tracking. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 179–190. Springer, Heidelberg (2008)
15. Wu, B., Singh, V., Kuo, C., Zhang, L., Lee, S., Nevatia, R.: CLEAR 2007 evaluation of USC human tracking system for surveillance videos. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 191–196. Springer, Heidelberg (2008)
16. Khan, S.M., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 133–146. Springer, Heidelberg (2006)
17. Mittal, A., Davis, L.: M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. In: IJCV, pp. 189–203 (2003)
18. Reddy, D., Sankaranarayanan, A., Cevher, V., Chellappa, R.: Compressed sensing for multi-view tracking and 3-D voxel reconstruction. In: ICIP, pp. 221–224 (2008)
19. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. PAMI 30, 267–282 (2008)
20. Liem, M., Gavrila, D.: Multi-person tracking with overlapping cameras in complex, dynamic environments. In: BMVC (2009)
21. Argyros, A.A., Lourakis, M.I.A.: Real time tracking of multiple skin-colored objects with a possibly moving camera. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 368–379. Springer, Heidelberg (2004)
22. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. In: SIGGRAPH, pp. 163–169 (1987)
23. Tzevanidis, K., Zabulis, X., Sarmis, T., Koutlemanis, P., Kyriazis, N., Argyros, A.: From multiple views to textured 3d meshes: a gpu-powered approach. In: ECCV Workshops, pp. 5–11 (2010)
24. Zabulis, X., Sarmis, T., Argyros, A.A.: 3D head pose estimation from multiple distant views. In: BMVC (2009)
25. Voit, M., Nickel, K., Stiefelhagen, R.: Neural network-based head pose estimation and multi-view fusion. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 291–298. Springer, Heidelberg (2007)
26. Zhang, Z., Hu, Y., Liu, M., Huang, T.: Head pose estimation in seminar room using multi view face detectors. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 299–304. Springer, Heidelberg (2007)
27. Tian, Y., Brown, L., Conell, J., Pankanti, S., Hapapur, A., Senior, A., Bolle, R.: Absolute head pose estimation from overhead wide-angle cameras. In: AMFG, pp. 92–99 (2003)
28. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24, 603–619 (2002)
29. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–588 (2001)
30. Franco, J., Menier, C., Boyer, E., Raffin, B.: A distributed approach for real time 3D modeling. In: CVPR Workshops, p. 31 (2004)
31. Zabulis, X., Grammenos, D., Sarmis, T., Tzevanidis, K., Argyros, A.A.: Exploration of large-scale museum artifacts through non-instrumented, location-based, multi-user interaction. In: VAST (2010)
32. Naruniec, J.: Using GPU for face detection. In: SPIE, vol. 7502, pp. 204–206 (2009)

Egocentric Visual Event Classification with Location-Based Priors

Sudeep Sundaram and Walterio W. Mayol-Cuevas

Department of Computer Science, University of Bristol

Abstract. We present a method for visual classification of actions and events captured from an egocentric point of view. The method tackles the challenge of a moving camera by creating deformable graph models for classification of actions. Action models are learned from low resolution, roughly stabilized difference images acquired using a single monocular camera. In parallel, raw images from the camera are used to estimate the user's location using a visual Simultaneous Localization and Mapping (SLAM) system. Action-location priors, learned using a labeled set of locations, further aid action classification and bring events into context. We present results on a dataset collected within a cluttered environment, consisting of routine manipulations performed on objects without tags. 

1 Introduction and Related Work

Visual event and activity classification has been mostly studied for cases when the camera is static and/or where the action is well centered and localized in the image [1–4]. Less work has been concerned with the case of a moving camera, which is the situation in systems that are observing inside-out e.g. a wearable system.

Detecting events and activity on the move can lead to assistive devices and this is indeed one of the primary goals for work carried out in this area. Examples of this are applications ranging from monitoring systems for the elderly and disabled [5–7], to systems that “watch and learn” how to carry out complex tasks [8, 9].

When used for sequential activity recognition, knowledge of *where* the user is in each time step, can play a vital role in ensuring robustness of the system. Location also brings about the all-important element of context in terms of the user's interaction with the immediate environment. Benefits of using location for recognition of user activity have previously been demonstrated. In [10] and [6], *only* location is used, while in [11], location is combined with signals obtained from a microphone to recognize activity. To the best of our knowledge, location has yet to be combined with human actions to recognize events and activity on the move. Several recent systems have demonstrated significant interest in human action recognition using cameras, although again a majority of the methods deal with appropriately placed static cameras.

Feature descriptors used for representing actions, can be broadly classified based on sparse or dense sampling of feature points from space-time representations of actions.

¹ The authors are deeply grateful to the British Council for the PhD studentship granted to SS, and to the EUFP7 COGNITO project for partially funding WMC.

Space-time interest points for action recognition were used in [12–14] and yielded considerably good results. However, densely sampled features have been shown to generally perform better [15]. In particular, Histograms of Oriented Gradients [1] has been a popular choice of feature descriptor for actions [2–4].

Visual sensing for user location meanwhile, has recently seen important advances and of particular importance here, are those methods amenable to real-time performance. Specifically, some works related to localization and mapping [16, 17] have developed fast methods for re-location.

2 Motivation and Contributions

Any method for recognition of egocentric manipulations must address problems arising due to - (1) camera motion, (2) changes in camera vantage point, (3) variations in the way a manipulation is performed, and (4) computational efficiency, to enable real-time performance.

In order to address the first problem, we carry out coarse stabilization of the input sequence to compensate for camera motion, as described in Section 3.1. The presented method learns *translation-invariant, deformable* graph models (covered in Section 5) for each manipulation class, thus addressing the second and third issues. Computational efficiency is ensured by classifying actions using low resolution images.

In order to build on the use of location for activity recognition, we use the same wearable camera to estimate the user’s location within a labeled (not tagged) environment using a Simultaneous Localization and Mapping system. Learned prior distributions of manipulations performed in known locations are used to enhance the classifier’s performance.

The contributions of this paper are two-fold - (1) the use of deformable graph-based action models for classification of egocentric visual events, and (2) the estimation of user location from the same sensor to bring events into context, resulting in improved classification.

3 Action Cell

This section describes the steps leading up to feature-based representations of action sequences. Our approach for event classification aims to avoid the recognition of individual objects that are being manipulated, and concentrates instead on the more generic hand and arm motion and the general working scene detection. This demands careful extraction of manipulation data with reduced background noise.

3.1 2-D Affine Image Registration

Given our use of a wearable camera, we first attempt to roughly compensate for camera motion relative to the background. Let I_k be the current frame being processed, and I_{k+1} be the next incoming frame. Let $\phi(I, \mu)$ denote the result obtained when an image I is affine transformed by parameter vector μ . We approximate camera motion

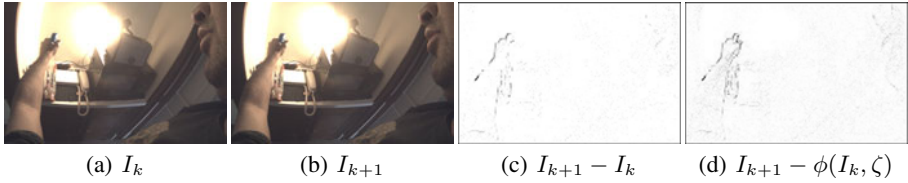


Fig. 1. Consecutive frames (*a*, *b*) from a “Spray” action sample, with contrast normalized difference images (*c*, *d*). Once camera motion is compensated, foreground motion of the hand and object is more clearly visible.

compensation by finding the affine transformation parameter vector ζ that minimizes the absolute intensity difference between $\phi(I_k, \zeta)$, and I_{k+1} . Estimation of ζ and the corresponding stabilized difference image D_k is given by:

$$\zeta = \operatorname{argmin}_{\mu} (I_{k+1} - \phi(I_k, \mu)) \quad (1)$$

$$D_k = I_{k+1} - \phi(I_k, \zeta) \quad (2)$$

3.2 Action Cell Extraction and Matching

The volume of stabilized difference images over the action sequence is then split into $16 \times 16 \times t$ spatio-temporal blocks, where t is the temporal length of the sequence. Histograms of oriented gradients are computed from the image contained within each time step in each spatio-temporal block, as described in [11]. The histograms are then concatenated over the entire temporal length to obtain a feature vector of size $n_b \times t$, where n_b is the number of histogram bins. We term this feature vector, along with information about its spatial location in the image, as an *action cell*.

In order to match any two given action cells a_1 and a_2 , respectively of lengths t_1 and t_2 , a distance matrix Δ of size $t_1 \times t_2$ is constructed such that cell Δ_{xy} contains the L_2 distance between histograms $x \in a_1$ and $y \in a_2$. Normalized dynamic time warping is used to compute a matching cost between a_1 and a_2 by finding the shortest path through Δ .

4 Action Fragment

A number of action cells belonging to a single action sequence may be modeled as vertices of a graph, which we term as an *action fragment*. This section deals with the extraction and matching of action fragments. Formally, any given action sequence A can be converted to a sequence of roughly stabilized difference images, which in turn is used to generate a set of action cells α i.e. $A := \{\alpha_i : i = 1, 2, \dots, |\alpha|\}$. An action fragment is located on the action sequence as a set of unique action cells, and is modeled as graph $\mathcal{Y}_A = \langle v_A, \varepsilon_A \rangle$, whose vertices $v_A \subseteq \alpha$ are the action cells, and edges ε_A may be found using one of various methods such as Delaunay Triangulation.

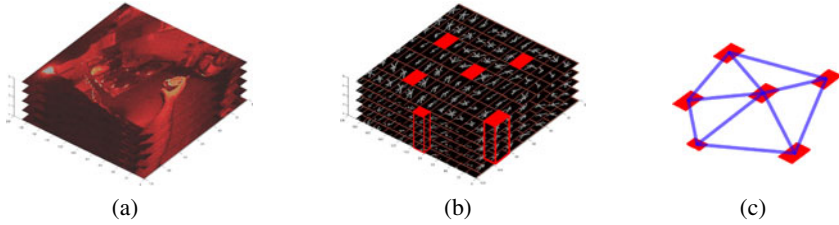


Fig. 2. Example of an action sequence (a) represented as a set of action cells (b). Randomly picked action cells form the vertices of a graph that represents a seed fragment (c).

4.1 Seed Fragment

The first step in every iteration of the learning process is to randomly pick an action fragment in some given action sequence. In the above example, this is done by specifying $|v_A| : 1 \leq |v_A| \leq |\alpha|$, and populating v_A with action cells randomly picked from α . ε_A is determined by performing a Delaunay Triangulation on the spatial centroids of v_A . An example of this process is shown in Figure 2.

4.2 MRF-Based Fragment Localization

Consider any two action sequences A and B . Let $\mathcal{Y}_A = \langle v_A, \varepsilon_A \rangle$ be an action fragment computed from A . This section deals with the localization of \mathcal{Y}_A in B . In other words, we attempt to find the action fragment $\mathcal{Y}_B = \langle v_B, \varepsilon_B \rangle$ such that the cost $C(\mathcal{Y}_A, \mathcal{Y}_B)$ of matching \mathcal{Y}_A to \mathcal{Y}_B is minimized:

$$\mathcal{Y}_B = \operatorname{argmin}_{\mathcal{Y}_b} (C(\mathcal{Y}_A, \mathcal{Y}_b)) \quad \forall \mathcal{Y}_b \in B \quad (3)$$

Finding an exact solution for \mathcal{Y}_B is clearly NP-hard. Instead, we use the MAX-SUM approach [18] to find an approximation, in a manner previously adopted in [19].

The matching cost can be measured as:

$$C(\mathcal{Y}_A, \mathcal{Y}_B) = C(v_A, v_B) + C(\varepsilon_A, \varepsilon_B) \quad (4)$$

where $C(v_A, v_B)$ measures the cost of matching corresponding action cells between the two fragments, and $C(\varepsilon_A, \varepsilon_B)$ measures the cost of matching structures of the two graphs. The dual of Equation 4 would be to maximize the *qualities* of v_B and ε_B . The overall quality of the localized fragment \mathcal{Y}_B , to be maximized, is given by:

$$Q(\mathcal{Y}_B) = \sum_{v \in v_B} Q(v) + \sum_{\varepsilon \in \varepsilon_B} Q(\varepsilon) \quad (5)$$

\mathcal{Y}_B is computed using a Markov Random Field, which represents a graph consisting of $M = |v_A|$ nodes. The adjacency of the nodes is maintained as in \mathcal{Y}_A . Each node, called an object, consists of N fields or labels, with associated qualities. The labels of two adjacent nodes are fully connected by N^2 edges. An example of such a graph is shown in

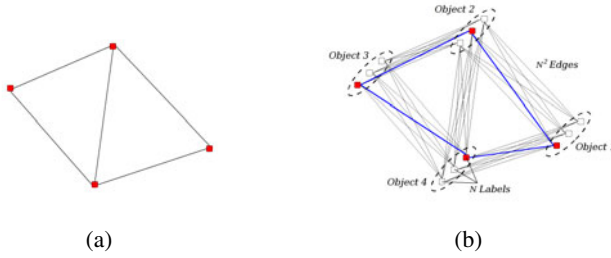


Fig. 3. Fragment Localization using a Markov Random Field. The model graph (action fragment) in (a) is localized on the MRF (b). The localized action fragment is highlighted by shaded (red) vertices and thick (blue) edges.

Figure 3. Maximizing Equation 5 is equivalent to finding a maximum posterior configuration of the MRF shown. In the current problem, labels corresponding to some Object i represent the N action cells in v_B , that are most similar to $v_A(i)$. Labels are found using an exhaustive search through the set v_B , using the matching technique described in Section 3.2. The quality of a label is inversely proportional to the cost for matching the label to its corresponding object. Dummy labels, with relatively low qualities, are added to each object, to facilitate localization where one or more objects remain unmatched. Label qualities for a single object are normalized to have a maximum value of 0 and a median of -1, and therefore lie in the range $[-\infty, 0]$. The quality of an edge is computed as the weighted sum of its length and orientation similarities to the corresponding edge in the model graph. Edge qualities are normalized to lie in the range $[-1, 0]$.

Let the $M \times N$ matrix L represent the label qualities for each of the objects, and the $|\varepsilon_A| \times N^2$ matrix E represent edge qualities between pairs of labels. The total quality of the labeling $S = \{n_1, \dots, n_M\}$ with $n_i \in \{1, \dots, N\}$ is given by

$$Q(S) = \sum_{m=1}^M L(m, S(m)) + \sum_{e=1}^{|\varepsilon_A|} E(e, \beta(E, S, \varepsilon_A)) \tag{6}$$

where $\beta(E, S, \varepsilon_A)$ denotes the column representing the edge between the labels chosen to represent the edge $\varepsilon_A(e)$. Rewriting Equations 3, 4 and 5 in terms of the MAX-SUM problem, Υ_B can be computed by finding the set $S^* = \text{argmax}_S(Q(S))$.

5 Learning Action Models

Consider a dataset of manipulations $A = \{A_i : i = 1, \dots, Z\}$ labeled class-wise, consisting of Z classes. Let any class i be represented by the set $A_i = \{A_{ij} : \text{class}(A_{ij}) = i ; j = 1, \dots, |A_i|\}$ of manipulations A_{ij} .

5.1 Fragment Models

Consider a seed fragment Υ_m chosen from a randomly picked manipulation sample A_{im} , as described in Section 4.1. In order to build a fragment model, we first localize

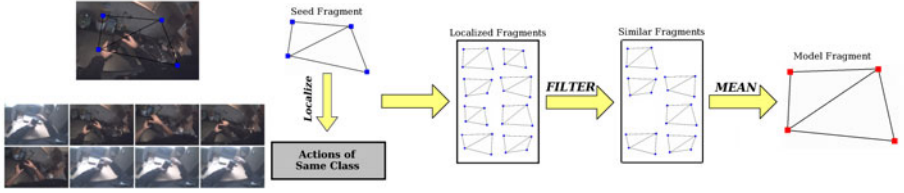


Fig. 4. From seed fragment Υ_m to similar fragments κ to model fragment Υ_f

Υ_m in all samples $A_{ij} \in A_i$, resulting in a set of similar action fragments κ .

$$\kappa = \{\Upsilon_j : j = 1, \dots, |A_i| ; \Upsilon_j = \underset{\Upsilon}{\operatorname{argmin}} (C(\Upsilon_m, \Upsilon)) \forall \Upsilon \in A_{ij}\} \quad (7)$$

Each Υ_j is evaluated using the quality of localizing Υ_m on A_{ij} , obtained using Equation 6. If this quality is found to be low, then Υ_j is discarded from κ . Once all fragments are validated, if the size of κ is too small, then a new seed Υ_m is found, and the process repeats. This filtering step ensures *consistency* across learned action fragments of the same class.

If the size of κ is large enough, a fragment model can be created. Using each action cell $v_m \in \Upsilon_m$ as a reference, each of the corresponding action cells $v_j \in \Upsilon_j$ are re-aligned using dynamic programming as described in Section 3.2 so that they are of the same length as v_m . Mean feature vectors are then computed at each time step, resulting in a learned set of action cells. These action cells form the vertices of the learned fragment model Υ_f .

We now run a filtering step to check whether or not to retain Υ_f , by attempting to localize Υ_f on all samples in A . If Υ_f has been generated using a seed from class i , and q_{jk} represents the quality obtained by localizing Υ_f on some sample A_{jk} , then the quality Q_f of Υ_f is computed as follows:

$$\eta_s = \frac{1}{|A_i|} \sum_{j=1}^Z \sum_{k=1}^{|A_j|} \delta(i, j) e^{q_{jk}} \quad (8)$$

$$\eta_d = \frac{1}{|A| - |A_i|} \sum_{j=1}^Z \sum_{k=1}^{|A_j|} (1 - \delta(i, j)) e^{q_{jk}} \quad (9)$$

$$Q_f = \frac{\eta_s}{\eta_s + \eta_d} \quad (10)$$

where $\delta(\cdot, \cdot)$ is the Dirac delta function.

If Q_f is high enough, then the component action cells are assigned weights equal to Q_f and added to the consensus set Ψ_i . If Q_f is not high enough, the fragment is discarded and the process repeats. This step ensures that the learned fragment is *discriminative* across classes.

Note that the graphs representing all $\Upsilon_j \in \kappa$ have the same adjacency matrix, since they were all obtained by localizing a single seed. While this structure remains consistent, the spatial location of each of the corresponding action cells is likely to vary. This

makes the fragment model *translation-invariant* and *deformable*. Changes in the camera mount and/or changes in the user’s pose will thus, have little effect on classification.

5.2 Fragments to Actions

Section 5.1 described the procedure to obtain a single fragment model, and assign weights to the component action cells. This procedure is repeated a number of times for each class, resulting in consensus sets $\{\Psi_i : i = 1, \dots, Z\}$. The next step is to convert each of these consensus sets into models that can be used for action classification. Figure 5(a) shows an example consensus set, that consists of a number of action cells, obtained from “high quality” action fragments. It is likely that highly discriminative action cells are present as part of more than one action fragment, while the less discriminative ones may occur only once. In order to retain the more important action cells, we perform Euclidian distance based K-Means clustering, as shown in Figure 5(b). Clusters with low populations are discarded. In each cluster that remains, the action cell with the highest weight is retained as a representative, and is assigned a weight equal to the sum of the weights of all elements in the cluster. The remaining action cells in the cluster are discarded.

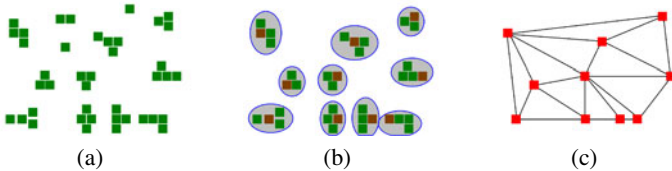


Fig. 5. Consensus set Ψ_i (a), clustered using KMeans (b) to form the action model ψ_i (c)

We now have a set of weighted highly discriminative action cells. We use these action cells as vertices of a graph, whose edges are given by performing a Delaunay Triangulation on the centroids of the action cells. An example is shown in Figure 5(c). The vertex weights in this graph are normalized to form a probability distribution, while the edges are assigned equal weights. This gives rise to an *action model* ψ_i for each class.

The model-building process described above takes place offline on a subset of the available dataset. Classification of the actions, on the other hand, is designed to be online, and happens immediately after the action is complete.

5.3 Action Classifier

Given a test action, it is matched with the learned models using the MRF-based method described in Section 4.2. This time, instead of finding action fragments, the matcher attempts to find the graphical action model on the test action. The recognized class ω for any input action Ω is given by $\omega = \operatorname{argmax}_i(Q(\psi_i, \Omega))$, where $Q(\psi_i, \Omega)$ is the quality of localizing action model ψ_i on action Ω .

6 User Location for Action Classification

Daily routine actions and activities are usually performed in the same environment(s) on a regular basis. We aim to take advantage of consistent information available in the user’s surroundings to improve event classification. Sparse maps of 3D features representing the user’s environment are built, from the camera that is used for action classification, using a Simultaneous Localization and Mapping system. We are more interested in the localization part, and thus the maps are built offline and stored. Probability distributions of actions performed in each map are learned, and used as priors to improve the accuracy of the action classifier.

Given some action sequence, the SLAM system is pushed into relocalization mode. In this manner, there is a resemblance with the work of [20] where disjoint locations are used for placing information. Here however, we use the method for relocalization described in [17] for its performance and reduced memory requirements. The localization method is robust to a degree of alterations in the mapped environment as produced by objects moving or being occluded. If the system manages to relocalize in one of the stored maps, the corresponding action prior is loaded and used to improve classification accuracy.

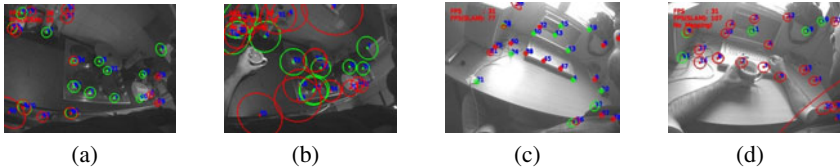


Fig. 6. User Location is provided by Relocalization in a 3D SLAM map. Figures (a,c) show examples of building a SLAM map offline, while (b,d) show relocalization in the classifier.

7 Experiments and Results

7.1 Dataset

Manipulation samples were collected by a single candidate over different days spanning 4 weeks - in order to capture natural variations in action. Samples were collected for 7 manipulation classes in 4 locations. Figure 7 shows the user setup and a map of the environment, followed by snapshots of the 4 locations where samples were collected. The manipulation classes (and the location(s) they were performed in) include *Answer Phone* (Niche), *Chop* (Kitchen), *Drink* (Kitchen, Desk), *Open Door* (Door), *Pour* (Kitchen), *Spray* (Kitchen, Niche), *Unscrew* (Kitchen, Desk). In all, the dataset consists of 277 manipulations, with a minimum of 34 per class.

7.2 Results

A subset of the dataset - 12 samples from each class - was randomly selected to train the classifier. The learned models provided a classification accuracy of 95.24% on the

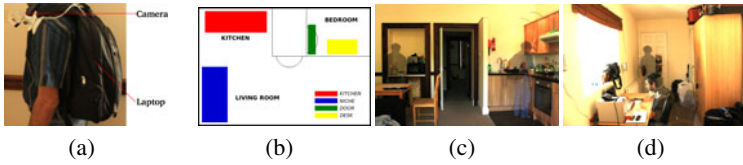


Fig. 7. User Setup and Environment. (a) shows the user with the wearable setup. (b) shows the environment used, consisting of (c) kitchen, niche, (d) desk and door locations.

training set alone. In order to analyze the performance of the classifier, statistics were generated over the entire dataset for varying values of N (number of labels in MRF), both with and without the use of location-based priors. Classification accuracies are measured for each individual action model as $\eta_i = \frac{TP(i)+TN(i)}{|A|}$, where $TP(i)$ and $TN(i)$ are respectively the number of “true positive” and “true negative” classifications for class i . The overall classification accuracy is computed as $\eta_{all} = \frac{\sum_i TP(i)}{|A|}$.

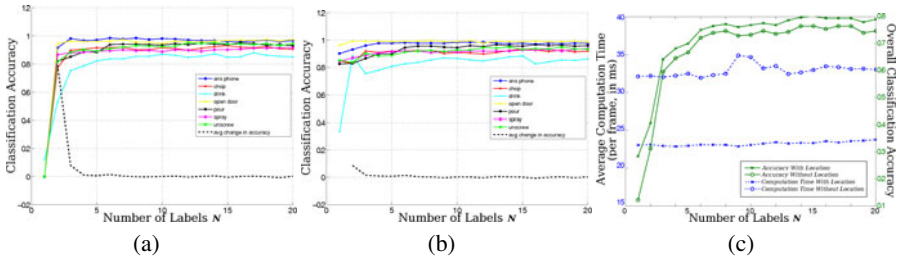


Fig. 8. Classification accuracies for each class (a) without location and (b) with location, with the dashed line indicating the change in average individual classification accuracy. (c) Overall classification accuracy, compared with computation times per frame.

Figure 8 contains classification accuracy plots for individual classes, and for the overall dataset analyzed against varying values of N . Classification accuracies for individual classes (see Figures 8(a) and 8(b)) remain consistently above 80% both with and without the use of location, for $N \geq 3$. With location, the overall classification improves by 4.3% on average, but more importantly reduces the computation time by 29.23%, due to the reduced number of models to be matched (Figure 8(c)).

8 Conclusion

We have presented a method that learns probabilistic graphical models to describe actions observed from a wearable camera. Further, we used the same sensor to estimate the user’s location, and combined this information with the action classifier to improve

its accuracy and performance. The results also validate the use of a monocular camera as a stand-alone sensor, capable of recognizing user manipulation activity without the need to recognize individual objects. Future work involves tests of our method for a number of candidates over longer periods of time. An extension to this work will involve automatic detection and classification of events from continuous video.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. Klaser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
3. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
4. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.: Action detection in complex scenes with spatial and temporal ambiguities. In: ICCV (2009)
5. Tapia, E.M., Intille, S.S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha, A., Mattern, F. (eds.) PERSVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
6. Liao, L., Patterson, D.J., Fox, D., Kautz, H.: Learning and inferring transportation routines. *Artificial Intelligence* 171, 311–331 (2007)
7. Van Laerhoven, K., Berlin, E.: When else did this happen? efficient subsequence representation and matching for wearable activity data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823. Springer, Heidelberg (2009)
8. Kuniyoshi, Y., Inaba, M., Inoue, H.: Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation* 10, 799–822 (1994)
9. Hovland, G.E., Sikka, P., McCarragher, B.J.: Skill acquisition from human demonstration using a hidden markov model. In: IROS (1996)
10. Aoki, H., Schiele, B., Pentland, A.: Realtime personal positioning system for wearable computers. In: ISWC (1999)
11. Clarkson, B., Mase, K., Pentland, A.: Recognizing user context via wearable sensors. In: ISWC (2000)
12. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV (2003)
13. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
14. Patron Perez, A., Reid, I., Patron, A., Reid, I.: A probabilistic framework for recognizing similar actions using spatio-temporal features. In: BMVC (2007)
15. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
16. Williams, B., Klein, G., Reid, I.: Real-time SLAM relocalisation. In: ICCV (2007)
17. Chekhlov, D., Mayol Cuevas, W., Calway, A.: Appearance based indexing for relocalisation in real-time visual slam. In: BMVC (2008)
18. Werner, T.: A linear programming approach to max-sum problem: A review. *PAMI* 29, 1165–1179 (2007)
19. Donner, R., Micusik, B., Langs, G., Bischof, H.: Sparse mrf appearance models for fast anatomical structure localisation. In: BMVC (2007)
20. Castle, R.O., Klein, G., Murray, D.W.: Video-rate localization in multiple maps for wearable augmented reality. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318. Springer, Heidelberg (2008)

View Invariant Activity Recognition with Manifold Learning

Sherif Azary and Andreas Savakis

Computing and Information Sciences and Computer Engineering
Rochester Institute of Technology, Rochester NY 14623

Abstract. Activity recognition in complex scenes can be very challenging because human actions are unconstrained and may be observed from multiple views. While progress has been made in recognizing activities from fixed views, more research is needed in developing view invariant recognition methods. Furthermore, the recognition and classification of activities involves processing data in the space and time domains, which involves large amounts of data and can be computationally expensive to process. To accommodate for view invariance and high dimensional data we propose the use of Manifold Learning using Locality Preserving Projections (LPP). We develop an efficient set of features based on radial distance and present a Manifold Learning framework for learning low dimensional representations of action primitives that can be used to recognize activities at multiple views. Using our approach we present high recognition rates on the Inria IXMAS dataset.

1 Introduction

The automatic recognition of human actions is a fundamental but challenging task in computer vision research for applications such as surveillance and scene understanding. Human activities are unconstrained and can drastically vary among individuals in many ways depending on shape, size, timing and view angle. To account for view invariance, [2] used image based rendering to reconstruct optimal views that would be ideal for classifiers. Silhouettes from multiple cameras were captured and projected into a 3D space so that the 3D motion path could be determined. These motion paths were then used to determine and create the optimal orthogonal views needed for classifiers. However, this system assumes linear motion paths so activities such as turning around and punching are not ideal. The work in [5] used five camera views of the same action and manifold learning was used to represent the data in a lower-dimensional space resulting in a clear identification of a single action view in a single manifold. This means that estimations can be made about untrained views of the same action which would result in a view invariant human activity recognition system.

Two general approaches to support time invariance of an activity are image analysis on individual frames and analysis on an entire sequence of images that constitutes a single action. The authors of [4] used optical flow to extract spatial gradient

descriptors on a per frame basis to define sequences of images as a calculated action using a voting scheme over time. In [5], the 2D Radon transform was applied to each frame of an action. The 2D Radon transform was converted to a 1D signal called the R-Transform and a surface was created as a sequence of these signals called the RXS surface. These surfaces could then be scaled down to a constant time interval while preserving action information supporting the concept of time invariance. However, the issue with such an approach is that a sliding time window is required in the testing phase which is not ideal since various activities cannot be recognized by a fixed length of time.

To account for the fact that the same activity can appear very different, researchers have also been looking into ways of finding commonalities that can allow for a general description of such activities. In [3], the contours of scaled silhouettes were used to identify structural and dynamic features including width, centroid displacements over time, and standard deviations. By extracting such information their system was capable of finding common relationships between various individuals conducting the same activities.

To accommodate for localization invariance, [4] used spatiotemporal segmentation to localize activities and a voting approach for activity classification. On a per frame basis, the Mean Shift Mode Estimation method was used to localize the spatial boundaries of a subject in a scene. Meanwhile, [5] determined the largest mass or silhouette in a scene and used bounding boxes to determine the subject's location. Although localization would not be a factor the system would be limited to processing one individual per scene.

In this paper, we present a methodology for recognizing view invariant human activities using manifold learning, specifically Locality Preserving Projections (LPP) [6]. Manifold learning is a technique for nonlinear dimensionality reduction. In essence, manifold learning attempts to identify low dimensional representations embedding high dimensional data. The work in [5] created one manifold for each action using Isomap which means a manifold would be needed to classify a single activity. In our approach we create just one manifold defining all actions of all views and requiring only one transformation to reduce our large input data set using LPP. This technique is similar to Principal Component Analysis (PCA). The main difference between PCA and LPP is that PCA is a linear technique that projects data along the directions of maximal variance while LPP is a methodology of obtaining linear projective mappings which preserve the neighborhood structure of a data set [1]. The details of LPP will be discussed in the following section.

2 Representing Activities and Training

2.1 Radial Distance

For each frame of an entire activity video, the silhouette can be represented by binary images after background subtraction. To efficiently describe a silhouette in some detail while maintaining robustness to noise, we define the radial distances from the silhouette centroid to the farthest contour at various angles and capture the entire

signature over 360 degrees. We identify connected components and their corresponding areas, bounding box regions, and centroid. For the training phase, we crop the largest detected object and process it, because it is assumed during the controlled training phase that there is only one individual conducting an activity at a time and the largest connected component in a frame is that individual. In the testing phase, the system does not make such assumptions and can process multiple individuals in a single scene. By cropping the detected connected region, the region $I(x,y)$ can be processed while preserving the characteristics of scale and localization invariance since the size and location of the silhouette becomes ignored.

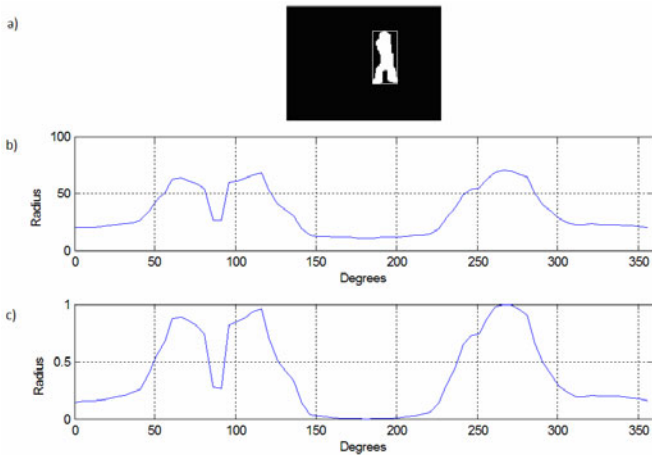


Fig. 1. An example of (a) a bounding box around a silhouette, (b) the corresponding radial measure plot over 72 evenly distributed angles, and (c) the normalized signal. The two peaks between 50 and 150 degrees represent the outline of the legs of the individuals and the peak at 265 degrees represents the detection of the individuals head.

Once a bounding box has been established along with the centroid of the silhouette, we convert the binary silhouette image to a contour plot and measure the Euclidean distance from the centroid to the bounds of the contour over 360 degrees in increments of 5 degrees using (1). This results in 72 radial measures that can then be used to form a 2D signal describing the radial distance measures of a silhouettes’ contour between 0 and 360 degrees as shown in Fig. 1. To further preserve scale invariance the radial magnitude is normalized using (2) and is shown in Fig. 1c.

$$I(x, y) \rightarrow r(\theta)$$

$$r(\theta) = \sqrt{(x_{centroid} - x(\theta)_{contour})^2 + (y_{centroid} - y(\theta)_{contour})^2} \tag{1}$$

$$r'(\theta) = \frac{r(\theta)}{\max_{\theta}(r(\theta))} \tag{2}$$

2.2 Radial Distance Surface

Time is added as an extra parameter, since an action is conducted over multiple frames. The radial distance approach is applied on a single instance of time. Therefore, an instance of a cropped region defined by $I(x,y)$ must be defined over time as $I(x,y,t)$. In [5], the R-Transform of each frame of an action was combined to form the RXS surface that described the entire activity over time. In our process we follow a similar approach by creating a radial distance surface that can also describe an activity over time. Equations (1) and (2) are modified to include time as a parameter resulting in equations (3) and (4).

By incorporating time, the 2D signals defining an instance in time become a 3D surface defined by radial magnitude, angle, and time as shown in Fig. 2. As previously mentioned an action can not be executed in an allotted amount of time. The same individual bending down in one scene might take six seconds in one trial and take ten seconds in another trial. The system must support time invariance and this can be done by normalizing the time axis of the surface.

$$I(x, y, t) \rightarrow r_t(\theta)$$

$$r_t(\theta) = \sqrt{(x_{t,centroid} - x(\theta)_{t,contour})^2 + (y_{t,centroid} - y(\theta)_{t,contour})^2} \quad (3)$$

$$r'_t(\theta) = \frac{r_t(\theta)}{\max_{\theta,t}(r_t(\theta))} \quad (4)$$

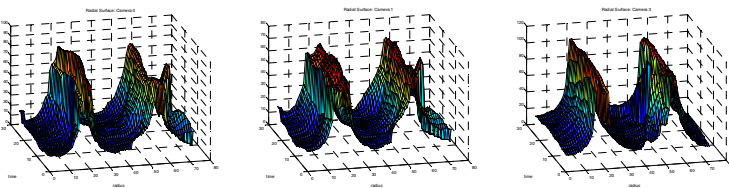


Fig. 2. An example of a 3-D surface plot defining the punching action from three different camera views

2.3 Locality Preserving Projections (LPP)

LPP is a linear dimensionality reduction algorithm that computes a lower dimensional representation of data from a high dimensional space. It is a linear approximation of the nonlinear Laplacian Eigenmap [6,7]. The transformation preserves local neighborhood information, which means that if an action can be represented in a lower dimensional space then the nearest neighbor search is indicative of a similar action.

Nonlinear methods such as LLE, Isomap, and Laplacian Eigenmaps can reveal the relationship of data along a manifold by learning the global structure of such

manifolds and finding mutual relationships among data points [12]. However, since these methods model data with nonlinear approaches, the lower dimensional representation of the data only represents the original high dimensional data. In other words, with these nonlinear approaches it is not clear how to map new test data. LPP is a linear algorithm and can easily map new test data which also makes this algorithm faster to execute than the other mentioned techniques [6,12].

Suppose we are given a set of data with m points such as $\{x_1, x_2, \dots, x_m\}$ in space R_n where R_n is the higher dimensional space of the original data set of n dimensions. The objective is to find a transformation matrix A that can map x_i to y_i with $\{y_1, y_2, \dots, y_m\}$ in space R_l for which $l < n$ as seen in (5). In other words, y_i is the representation of data in l -dimension space of the original data x_i in n dimensional space.

The first step of the LPP algorithm is to form the adjacency graph between nodes. Given G as a graph with m nodes, an edge is assigned between nodes i and j if x_i and x_j are close.

$$\begin{aligned}
 y_i &= A^T x_i \\
 x &\rightarrow n - \text{Dimensional Vector} \\
 y &\rightarrow l - \text{Dimensional Vector}
 \end{aligned}
 \tag{5}$$

Two variations of determining the closeness between nodes are the k -nearest neighbor and ϵ -ball [8]. The k -nearest neighbor approach is to select the k closest points to x_i . The ϵ -ball approach is to find points that satisfy (6) given a parameter of ϵ .

$$\|x_i - x_j\|^2 < \epsilon
 \tag{6}$$

Once the adjacency graph has been constructed, the next step is to assign weights to detected edges. A separate weight matrix W is formed of size $m \times m$. For unconnected nodes a weight of zeros is assigned while for connected nodes weights can be determined using two variations. The first is the Heat Kernel approach as shown in (7) for which a weight can be calculated given a parameter of t [6,7]. The second is the Simple-Minded approach for which a weight is automatically assigned a value of one if two nodes are connected as shown in (8).

$$w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}
 \tag{7}$$

$$w_{ij} = \begin{cases} 1 & \text{if edge} \\ 0 & \text{otherwise} \end{cases}
 \tag{8}$$

The final stage of the LPP algorithm is to form the Eigenmaps. First, calculate D which is the diagonal matrix whose elements are the sums of the columns of W as shown in (9). Second, calculate the Laplacian Matrix L as shown in (10) by subtracting the weight matrix from diagonal matrix. Finally, solve for the eigenvectors and eigenvalues in (11) for which a is a column of vectors which are the solutions of the

equation ordered according to their eigenvalues $\lambda_0 < \lambda_1 < \dots < \lambda_{l-1}$ [6]. Given the column vectors we now have A as shown in (12) which can be used in (5) to map between x and y .

$$D_{ii} = \sum_j W_{ji} \quad (9)$$

$$L = D - W \quad (10)$$

$$XLX^T a = \lambda XDX^T a \quad (11)$$

$$A = (a_0, a_1, \dots, a_{l-1}) \quad (12)$$

$A \rightarrow n \times l \text{ matrix}$

2.4 Training

At this point we have established a method of identifying an activity with a surface plot that is invariant to scale, localization, and time. This data, although normalized, is very large to represent any single activity of a specific view. Our system needs to be trained with a lot of input data of multiple activities trained at different views. A 50×100 radial distance surface contains 5,000 data points which means there are 5,000 dimensions to process for one activity of one view. In our training phase, we train with nine individuals performing three trials of ten actions at four views. Furthermore, the training set size was doubled because video sequences were mirrored horizontally to train additional views. This leads to 2,160 input data sets each defined by 5,000 dimensions, which means the system would be dealing with 10,800,000 data points. The system could become overwhelmed and too slow to operate in real world applications. The authors of [11] state that modern data sets representing a large number of features are a burden to algorithms that attempt to use them and, even worse, can be irrelevant and even misleading. Manifold learning can uncover underlying parameters and address the issue of view invariant activity recognition and multiple descriptors of the same activity by finding correlations of the high dimensional data in a lower dimensional space. To support view invariance, we use LPP to construct a single manifold and define each action as a single class regardless of the view from which the action was captured.

It is also necessary to describe the manifold that was created for our system. The authors of [5] created one manifold for each action using Isomap, which means that eleven manifolds would be needed to classify eleven different actions at multiple views. In our approach we create just one manifold defining all actions of all views and requiring only one transformation to reduce our large input data set using LPP. Fig. 3a shows that nine eigenvalues were calculated after training ten activities using only one manifold. This means that our original data set consisting of 5,000

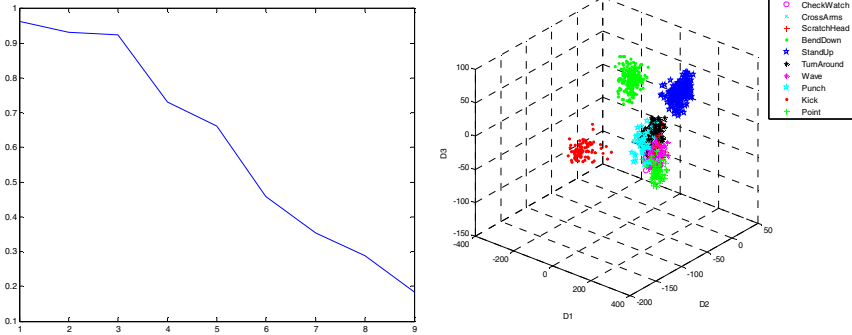


Fig. 3. The eigenvalues (*left*) for trained activities of Check Watch, Cross Arms, Scratch Head, Bend Down, Stand Up, Turn Around, Wave, Punch, Kick, and Point as well as their 3D lower dimensional representations (*right*)

dimensions can be represented in only nine dimensions. The corresponding 3D representation after applying LPP is shown in Fig. 3b. As can be observed in 3D space, there are clear separations between activities independently of the view.

3 Results

Our approach was applied to the Inria IXMAS data set which is available online and consists of twelve individuals executing eleven actions which include already generated silhouettes. The available actions are *check watch*, *cross arms*, *scratch head*, *bend down*, *stand up*, *turn around*, *wave*, *punch*, *kick*, *point*, and *throw ball*. For this research, all actions, except for *throw ball*, were chosen for training and testing because it was decided that human interactions with objects is outside the scope of this paper. The dataset was cleaned by excluding training video sequences with shadows, extraordinary noise, and un-descriptive views such as an individual pointing with their back to the camera. This ensures valid and descriptive training sets.

Each individual is represented by five different views as shown in Fig. 4. The fifth view is a top view which will be ignored for future processing since it does not provide any significance during training. The other views were mirrored horizontally to enlarge the training set and support training additional views. During the training phase, we were able to construct a manifold with clear separations in a lower dimensional space. This manifold was trained with nine individuals performing three trials of ten actions at eight different views. When testing new video scenes, the same approach is taken of obtaining the radial distance to the farthest contours from the centroid of all connected components in a scene. During the testing phase, the system can track multiple individuals in the same scene and process each individual separately. A radial distance surface is created and the data is then ready to be tested against the already constructed manifold. The data as in the training phase is

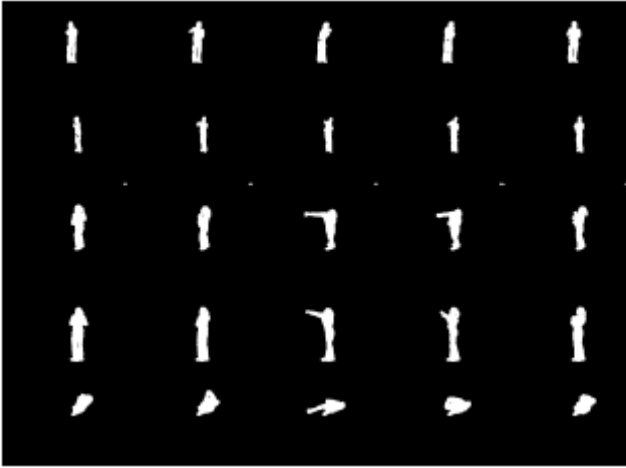


Fig. 4. Example silhouettes for a punching action for five views organized by row in the original Inria IXMAS dataset. The fifth view (fifth row) is a top view and is ignored.

vectorized resulting in a $l \times MN$ vector. This vector is multiplied by the eigenvectors determined from the training phase which is a $MN \times l$ data structure. The result after multiplication is a coordinate in the l dimensional space.

Since LPP applies a linear transformation as described by equation (5), we simply need to measure the Euclidean distance d of the lower dimensional representation of the data point to its nearest neighbors to classify an activity using (13).

$$d(p, q) = \sqrt{(p_{1,1} - q_{1,1})^2 + (p_{1,2} - q_{1,2})^2 + \dots + (p_{1,l} - q_{1,l})^2} \quad (13)$$

The following table shows the confusion matrix results after testing new data against the trained data using leave-one-out cross validation. Using the 1 nearest neighbor measure the overall accuracy was 92.48% with *turn around* being the most difficult action to classify. The accuracy with 3 nearest neighbor and 5 nearest neighbor is 93.23% and 93.98% respectively.

Overall the results look promising with the highest recognition rates using the 5 nearest neighbor measure. The biggest challenge is finding a clear separation between similar activities. For example, *scratch head* and *wave* can be confused because both actions require the act of raising an arm towards the head and, since the viewpoint the action is being captured from is not fixed, there is potential for confusion. The classification of the turning around action has a high error rate because the radial distance measure does not capture much useful information of this action over time. With actions such as punching and kicking the radial distance surface plot indicates a significant change while the turning around surface plot is not as descriptive.

Table 1. Confusion matrix for 1-NN with a 92.48% average accuracy

	CHECK WATCH	CROSS ARMS	SCRATCH HEAD	BEND DOWN	STAND UP	TURN	WAVE	PUNCH	KICK	POINT
CHECK WATCH	0.80		0.20							
CROSS ARMS		0.91				0.09				
SCRATCH HEAD			0.92				0.08			
BEND DOWN				1.00						
STAND UP					1.00					
TURN AROUND	0.13	0.04	0.04			0.79				
WAVE							1.0			
PUNCH							0.08	0.92		
KICK									1.0	
POINT							0.10			0.90

5 Conclusion

In this paper, we present a method for learning and classifying low dimensional representations of action primitives using manifold learning that can be used to recognize view invariant, scale invariant, localization invariant, and time invariant activities. We created a digital signature for each activity by processing individual frames using the radial distance approach and creating a radial distance surface. Then using manifold learning with LPP we were able to represent the entire activity in a lower dimensional space. New test video sequences were processed in the same manner and by using the nearest neighbor approach in the lower dimensional space we were able to classify activities with a high rate of accuracy.

References

- [1] Shlens, J.: A Tutorial on Principal Component Analysis. Salk Institute for Biological Studies, La Jolla (2005)
- [2] Bodor, R., Drenner, A., Fehr, D., Masoud, O., Papanikolopoulos, N.: View-independent human motion classification using image-based reconstruction. *Image and Vision Computing* 27(8), 1194–1206 (2009)
- [3] Cherla, S., Kulkarni, K., Kale, A., Ramasubramanian, V.: Towards fast, view-invariant human action recognition. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2008*, pp. 1–8 (2008)
- [4] Oikonomopoulos, A., Pantic, M., Patras, I.: An Implicit Spatiotemporal Shape Model for Human Activity Localization and Recognition. In: *IEEE Conf. Computer Vision and Pattern Recognition (Workshops)*, Miami, USA (June 2009)

- [5] Souvenir, R., Parrigan, K.: Viewpoint Manifolds for Action Recognition. *EURASIP Journal on Image and Video Processing* 2009, Article ID 738702, 13 (2009)
- [6] He, X., Niyogi, P.: Locality Preserving Projections. In: *Proc. Conf. Advances in Neural Information Processing Systems* (2003)
- [7] Belkin, M., Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, vol. 14 (2002)
- [8] Pless, R., Souvenir, R.: A Survey of Manifold Learning for Images. *IPJS Transactions on Computer Vision and Applications* 1, 83–94 (2009)
- [9] Ogale, A., Karapurkar, A., Aloimonos, Y.: View-Invariant Modeling and Recognition of Human Actions Using Grammars. In: *Int. Conf. on Computer Vision (ICCV), Workshop on Dynamical Vision*, Beijing, China (October 2005)
- [10] Oikonomopoulos, A., Patras, I., Pantic, M.: An implicit spatiotemporal shape model for human activity localization and recognition. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2009*, pp. 27–33 (2009)
- [11] Cayton, L.: Algorithms for Manifold Learning. Technical report, University of California, San Diego, California (2003)
- [12] Tang, Y., Rose, R.: A study of using locality preserving projections for feature extraction in speech recognition. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, March 31–April 4, pp. 1569–1572 (2008)
- [13] Cai, L., Du, S.: Rotation, scale and translation invariant image watermarking using Radon transform and Fourier transform. In: *Proceedings of the IEEE 6th Circuits and Systems Symposium Emerging Technologies: Frontiers of Mobile and Wireless Communication*, May 31–June 2, vol. 1, pp. 281–284 (2004)
- [14] Hejazi, M., Shevlyakov, G., Yo-Sung, H.: Modified Discrete Radon Transforms and Their Application to Rotation-Invariant Image Analysis. In: *IEEE 8th Workshop Multimedia Signal Processing* 2006, October 3–6, pp. 429–434 (2006)
- [15] Ye, M., Androutsos, D.: Robust affine invariant shape image retrieval using the ICA Zernike Moment Shape Descriptor. In: *16th IEEE International Conference Image Processing (ICIP)*, November 7–10, pp. 1065–1068 (2009)
- [16] Poppe, R., Poel, M.: Comparison of silhouette shape descriptors for example-based human pose recovery. In: *7th International Conference Automatic Face and Gesture Recognition, FGR 2006*, April 2–6, pp. 541–546 (2006)

Arm-Hand Behaviours Modelling: From Attention to Imitation

Sean R.F. Fanello, Ilaria Gori, and Fiora Pirri

Sapienza Università di Roma, Dipartimento di Informatica e Sistemistica,
Roma, RM, Italy
seanryan.fanello@gmail.com, ilary.gori@gmail.com,
fiora.pirri@dis.uniroma1.it

Abstract. We present a new and original method for modelling arm-hand actions, learning and recognition. We use an incremental approach to separate the arm-hand action recognition problem into three levels. The lower level exploits bottom-up attention to select the region of interest, and attention is specifically tuned towards human motion. The middle level serves to classify action primitives exploiting motion features as descriptors. Each of the primitives is modelled by a Mixture of Gaussian, and it is recognised by a complete, real time and robust recognition system. The higher level system combines sequences of primitives using deterministic finite automata. The contribution of the paper is a compositional based model for arm-hand behaviours allowing a robot to learn new actions in a one time shot demonstration of the action execution.

Keywords: gesture recognition, action segmentation, human motion analysis.

1 Introduction

We face the problem of modelling behaviours from a robot perspective. We provide an analysis of the role played by the primitive constituents of actions and show, for a number of simple primitives, how to make legal combinations of them in so enabling the robot to build and replicate the observed behaviour by its own.

Here we shall focus only on actions performed by hands and arms, although we extend the action class beyond the concept of gestures (as specified, e.g. in Mitra et al. survey [1]). In fact, potentially any general action, such as drinking or moving objects around, performable by hand and arm, can be included in our approach.

First of all we consider attention to and focus towards the human motion, as distinct from non-human motion, either natural or mechanical. This aspect, in particular, resorts to the theory of motion coherence and structured motion (see for example Wildes and Bergen [2]), for which oriented filters have been proven to be appropriate [3]. Indeed, we show how a bank of 3D Gabor filters can be tuned to respond selectively to some specific human motion. Thus, focus on regions of the scene interested by human motion provides the robot with a natural segmentation of where to look at for learning behaviours. In particular, attention to distinct human motion seems to be explicitly dependent on scale, frequency, direction, but not on shape. This fact has suggested us to

define descriptors based only on these features, from which we extract principally the directions of the arm-hand movements (see also [4]).

The very simple structure of the descriptors enables a straightforward classification that includes all direction dependent primitives, such as *up*, *tilt*, *release*, *grasp* and so on. The basic classification can be easily extended to any legal sequence of actions for which a deterministic accepting automata exists. In this sense, according to the classification on human motion analysis, as provided by Moeslund et al. [5], our approach seems to fall into the category of *action primitives and grammars*, as no explicit reference to human model is used in the behaviour modelling. For other taxonomies concerning action recognition we refer the reader to [6,7,8].

The purpose of our work encompasses the classification and regression problem (see [6,9]). The purpose is to enable robot action learning, by learning the primitives and their structured progression. This can be considered as a form of imitation learning (we refer the reader to the review of [10]) although an important generalisation inference is done in the construction of the accepting automaton.

Thus, at each step of the behaviour learning, the robot finds itself either modelling the behaviour via a new automaton, from the observed sequence, or accepting the observed behaviour via an already memorised one. This can be further extended by revising the learned automaton. Finally we have experimented the above model with the robot iCub (see Fig. 6), a humanoid robot designed by the RobotCub Consortium.

2 Focusing on Human Motion

Humans reveal a specific sensitivity to actions. It has been shown that action recognition is predominantly located in the left frontal lobe (see [11]) and that low level motion perception is biased towards stimuli complying with kinematics laws of human motion. Indeed, human visual sensitivity is greatest at roughly 5 cycle/degree and at 5 Hz.

We have used 3D Gabor filters to record responses to human motion and in particular to arm-hand motion in so as to learn attention towards these specific movements in a scene, as opposed to other kind of movements (e.g. a fan). We show that 3D Gabor filters can discriminate different motions by suitably selecting scale and frequency. The selected regions are used by the descriptors to identify primitives of actions.

The earliest studies on the Gabor transform [12] are due to Daugman [13] and to the experiments of Jones and Palmer [14], who tested the Daugman’s idea that simple receptive fields belong to a class of linear filters analogous to Gabor filters. Since then a wealth of literature has been produced on Gabor filters, to model several meaningful aspects of the visual process. Most of the works are, however, focused on the 2D analysis. A 3D Gabor, as the product of a 3D Gaussian and a complex 3D harmonic function, can be defined as follows:

$$\mathcal{G}(\mathbf{x}) = |\Sigma|^{-1/2} (2\pi)^{-3/2} \left[\exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\top \Sigma^{-1}(\mathbf{x} - \mathbf{x}_0)\right) \exp\left(2\pi i \mathbf{u}_0^\top (\mathbf{x} - \mathbf{x}_0)\right) \right] \quad (1)$$

Here $\mathbf{x} = (x, y, t)$, $\mathbf{x}_0 = (x_0, y_0, t_0)$ is the origin in the time-space domain, $\mathbf{u}_0 = (u_0, v_0, w_0)$ denote the central spatio-temporal frequency, finally $i = \sqrt{-1}$. Using Euler formula and simplifying, the harmonic term can be written as $\cos(-2\pi \mathbf{u}_0^\top \mathbf{x}_0 + \psi)$, with ψ the phase

parameters in Cartesian coordinates. From this, according to the ψ value, it is possible to obtain two terms in quadrature, the even and the odd Gabor filters (see, for example, [12] [15]), which we denote \mathcal{G}_O and \mathcal{G}_E . Clearly, with respect to Gabor’s representation of the information area [12] (see also [15]) these filters should be represented in a 6-dimensional space with coordinates x, y, t, u, v, w . However, following Daugman [15], we consider two representations, one in the space-time domain and one in the frequency domain. Here we shall mention only the space-time domain.

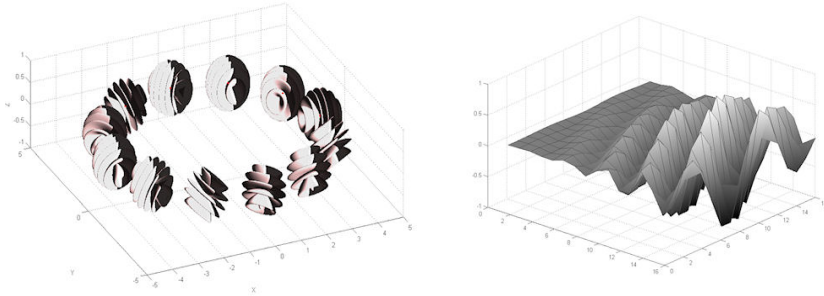


Fig. 1. On the left a bank of 3D Gabor filters with same scale and frequency and varying direction. On the right a slice along the $x - time$.

A Gabor can be specified by the parameters of scale (one per axis of the Gaussian support, which is an ellipsoid), of direction, by the angles θ and φ of the principal axes of the Gaussian support, and of central frequency. In fact, knowing the axes direction (eigenvectors) and the axes scale (eigenvalues) the Gaussian covariance Σ is determined. By varying these parameters, according to the spatial frequency contrast sensitivity and speed sensitivity in humans, providing limiting values, we have defined a bank of Gabor filters, in which the parameters range is specified as follows. Both the spatial and temporal frequencies are given as multiple of the Nyquist critical frequency $f_s = (1/2)$ cycle/pixel and $f_t = 12.5Hz$, given that the video sampling rate was $25Hz$. In particular, the frequency bandwidth is related to the Gaussian axes length as follows:

$$\Delta F_i = \frac{1}{2\sqrt{\lambda_i}}, \quad i = 1, \dots, 3 \tag{2}$$

With λ_i the i -th eigenvalue of Σ , and $\Delta F_i = F_i^{max} - F_i^{min}$ the maximal (resp. the minimal) frequency of the chosen channel in the i -th direction. We have chosen the central frequency to vary along 4 channels from $1/2$ to $1/8$ and, accordingly, the scale to vary from 0.1 to 0.8 for each of the axis of the Gaussian support. This amounts to 48 parameters. On the other hand the orientation is given for 6 directions, namely $\{0, 30, 60, 90, 120, 150\}$, for both the angles θ and φ . This amounts to 36 parameters. We have thus obtained a bank of 48×36 filters. Figure 1 shows, on the left, a bank of 3D

Gabor filters, with only the direction varying and the origin \mathbf{x}_0 varying on a circle, just for visualisation purposes.

To learn the human motion bias, we are given training videos V of about 800 frames taken at a sampling rate of about 25 Hz. The video resolution is reduced to 144×192 and a period of $\Delta T = 0.64s$ is considered to accumulate information, at the end of which the energy is computed. This amounts to volumes $V^{\Delta T}$ of 16 frames and, analogously the Gabor filter is defined by a volume of dimension $16 \times 16 \times 16$. The square of the motion energy, for the given interval, is defined as:

$$En_i^{\Delta T}(\mathbf{x})^2 = \sum_{\Delta T} \left(\int_{\mathbb{R}^3} \mathcal{G}_E^i(\mathbf{x}') V^{\Delta T}(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \right)^2 + \left(\int_{\mathbb{R}^3} \mathcal{G}_O^i(\mathbf{x}') V^{\Delta T}(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \right)^2 \quad (3)$$

Here the pair $(\mathcal{G}_E^i, \mathcal{G}_O^i)$ varies on the space of the filters bank, $\mathbf{x} = (x, y, t)$, $\mathbf{x}' = (x', y', t')$, and integration is triple on x', y' and t' . The energy is computed for each 3D Gabor in the filter bank, after smoothing the volume $V^{\Delta T}$, with $\Delta T \sim 0.64s$, with a binomial filter of size 3. Although the coverage of the bank is not complete, we look for the response that maximises the energy around a foveated region of at most 1 to 3 degrees, and it is minimal elsewhere. Intuitively, this means that the response of the receptive fields is sharp in the interesting regions. This fact, indeed, amounts to both maximise the energy and minimise the entropy of the information carried by the energy of the response. This is achieved by considering the energy voxels as i.i.d observations of a non-parametric kernel density. The non parametric kernel density of the energy is estimated using a uniform kernel (see next section, equation (7)), with bandwidth $H = 0.08 \cdot \mathbf{I}$, \mathbf{I} the identity matrix, along the 3 dimensions of the foveated region (for non parametric densities and the estimation of the bandwidth, we refer the reader to [16]).

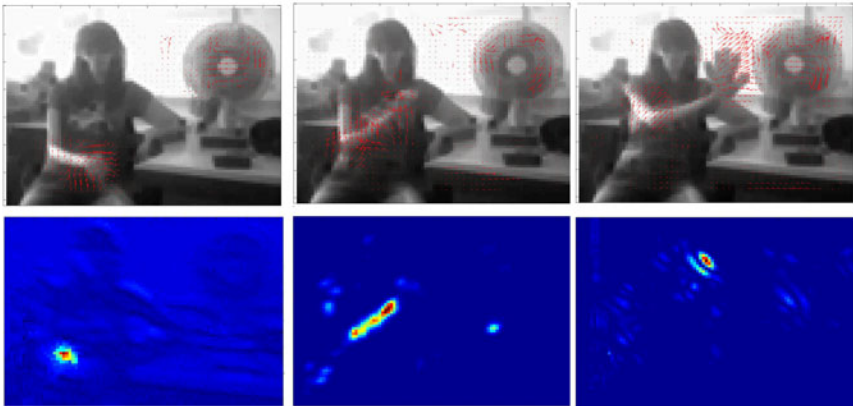


Fig. 2. The higher sequence illustrates the optical flow, detecting evenly the fan and the arm-hand motion. The lower sequence illustrates the energy of the quadrature pair Gabor filters, along a path of ΔT minimising the entropy. This path is constant over 1/6 cycle/pixel and varying direction. The scale is fixed with $a = 0.38, b = 0.46, c = 0.6$. here all images have been resized to 1/2.

On the other hand, the response is discriminative if the energy peaks are minimal in number, and hence the correlation is high on closest spatio-temporal regions. Therefore the optimisation criterion amounts to maximise the energy subject to both the minimisation of the entropy $E(p)$ of the non-parametric density p and the minimisation of an error function defined as the sum of the squared distances between any two energy peaks. Here a peak is any energy value x such that

$$x \geq \frac{4}{3N} \sum En. \quad (4)$$

Here N is the dimension of En obtained by vectorisation of En_i^{AT} . It is interesting to note that, under this optimisation criterion we have the following results:

1. Given ~ 800 frames at $25Hz$, with T about half minute, the maximisation of the energy, subject to the minimisation of the entropy, and subject to the minimisation of energy peaks distance, at each ΔT , ensures that the motion of a congruous source is tracked.
2. For attention towards human-motion only scale and central frequency influence energy response, while direction can be kept varying.
3. Human motion is located at the medium low frequencies of the filter bank.

It follows that, choosing a Gabor filter of any direction, with central frequency in space-time of about $(1/6)$ cycle/pixel, cycle/frame, with minimal scale, if the optimisation criteria are satisfied along the whole path T then at the peaked regions arm-hand motion is very likely to be included. Some experiments are shown in Figure 2 and compared with optical flow where we note that 3D Gabor filtering can discriminate between hand-arm motion and the fan motion, note that the fan had two different velocities.

3 Online Classification of Action Primitives

In the previous section we have specified how to obtain the region of interest (ROI), where arm-hand motion is identified, for each frame of a video. In this section we show how actions primitives can be classified online. As gathered in the previous section, the optimisation criteria for identifying human motion were not tuned to direction: as far as a movement of the hand or arm is displayed the motion direction varies continuously, therefore it is less relevant than scale and frequency. However once the motion

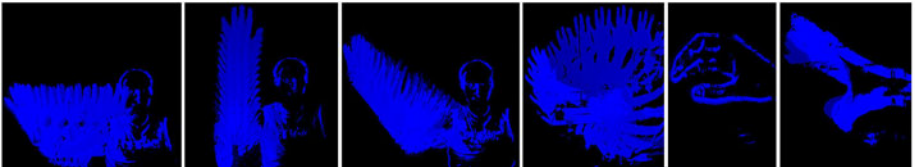


Fig. 3. The figure illustrates the grabbed directions of some of the gesture primitives in pairs: Right (0°)- Left (180°), Up (90°)- Down (270°), Tilt 45° - Tilt 215° , Tilt 135° - Tilt 315° , Rotate (Right and Left), Grab and Release

energy, as the squared sum of the responses of the two filters in quadrature, has been obtained, according to Heeger [17] (see also [18]), it is possible to recover the optical flow. However, once the scale and frequency have been selected for attention, any direction works well with the selected scale and frequency for online bottom-up attention. Therefore to ease performance in gesture tracking and online classification we have chosen to use a simple and well performing optical flow algorithm such as Horn and Shunck’s algorithm [19]. Other methods such as Lucas-Kanade’s algorithm [20], Variational Optical Flow [21] and Brox’s Optical Flow [22] are either too demanding, in terms of features requirements, or too computationally expensive. For example Brox’s algorithm slow down computation at 6 Hz, whereas for real-time tracking human motion 25 Hz are needed.

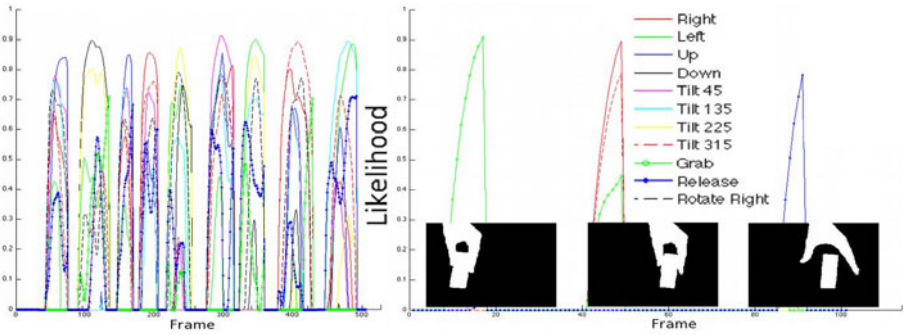


Fig. 4. The Figure on the left illustrates the likelihood computed at real-time at each time step t over a sequence of $T = 500$ frames. On the right the likelihood trend for the action “Grab-Right-Release”.

Let $\langle V(x, y, t), U(x, y, t), t \rangle_{t=1, \dots, T}$ be the optical flow vector, for each pixel in the ROI. The principal directions of the velocity vectors are defined as follows:

$$dir(x, y, t) = \frac{\pi}{2k} (\lceil \frac{k}{\pi} \arctan(\frac{V(x, y, t)}{U(x, y, t)}) \rceil + \lfloor \frac{k}{\pi} \arctan(\frac{V(x, y, t)}{U(x, y, t)}) \rfloor) \tag{5}$$

hence, at each (x, y) pixel in the ROI, at time t , the principal direction θ_j takes the following discrete values:

$$\theta_j = \pm \frac{(2j - 1)\pi}{2k}, \quad j = 1, \dots, 2k \tag{6}$$

Here k is half the number of required principal directions. We can note that the size of $dir(t)$ depends on the dimension of the ROI. In order to obtain a normalised features vector $\mathbf{X}(t) \in \mathbb{R}^{2k}$ we use a uniform kernel which, essentially, transforms $dir(t)$ into its histogram via a non parametric kernel density. More specifically, let $n = 2k$, let \mathbf{J} be the indicator function, let $\mathbf{Y}(t)$ be the vectorisation of $dir(t)$, with m its size, let x be an element of a vector of size n , scaled between $min(\mathbf{Y}(t))$ and $max(\mathbf{Y}(t))$:

$$K(u) = \frac{1}{2} \mathbf{J}(|u| \leq 1) \text{ hence for } u = \frac{x - Y_s(t)}{h}, \quad X(x, t) = \frac{1}{nh} \sum_{s=1}^m K\left(\frac{x - Y_s(t)}{h}\right) \tag{7}$$

Table 1. Confusion Matrix for 11 of the arm-hand primitives. Here *false* denotes a false positive gesture in the sequence.

	Right	Left	Up	Down	T45°	T135°	T225°	T315°	Grab	Rel	Rot	False
Right	0.7										0.15	0.15
Left		0.7										0.3
Up			1.0									
Down				1.0								
T45°	0.1				0.7						0.1	0.1
T135°						0.8						0.2
T225°				0.3			0.7					
T315°								1.0				
Grab	0.1	0.1							0.8			
Rel	0.1	0.1								0.8		
Rot	0.1				0.1						0.7	0.1
False												

here we have chosen $h = 1/2^8$. The obtained feature vector $\mathbf{X}(t) \in \mathbb{R}^{2k}$ is then used for any further classification of primitive actions.

Given a source of sequences of 100 arm-hand actions (gestures) we have defined 11 principal primitives. Figure 3 illustrates 5 pairs of primitives, *grab* and *release*. Once each frame is encoded into the above defined descriptor, we can obtain parametric descriptors by estimating for each primitive action a mixture of Gaussian. For the estimation, to suitably assess the number of components of each mixture, we have been using the Spectral Clustering algorithm ([23]). Therefore for each primitive action A_s , $s = 1, \dots, M$, with $M = 11$, the number of primitive actions considered, a mixture of Gaussian g_{A_s} , is estimated, with parameters $(\mu_1, \dots, \mu_m, \Sigma_1, \dots, \Sigma_m, \pi_1, \dots, \pi_m)$. The mixtures are directly used for classification.

Given a video sequence of length T , we need to attribute a class A_s to each feature descriptor \mathbf{X}_i , $i = 1, \dots, T$, $\mathbf{X}_i = \mathbf{X}(t_i)$, to establish the primitive actions appearing in the sequence. We note that because of the low frequency of arm-hand motion, it turns out that the same primitive holds for several frames, therefore it is possible to monitor the likelihood of a feature vector and searching the Gaussian space only at specific break points, indicating change direction.

Indeed, consider a buffer $\mathcal{B}_T = (\mathbf{X}_1, \dots, \mathbf{X}_T)$ of features vectors, as defined in equation (7), obtained by coding T frames, where $\mathbf{X}(t_i) = \mathbf{X}_i$ the i -th descriptor, at time t_i , in the buffer. The posterior distribution of each primitive action, given each feature in the buffer is estimated via the softmax function. Namely,

$$P(A_s|\mathbf{X}_i) = \frac{\exp(\lambda_k)}{\sum_j \exp(\lambda_j)} \quad \text{with } \lambda_i = \log g_{A_s}(\mathbf{X}_i|A_s)P(A_s) \tag{8}$$

Hence the observed primitive is classified to action A_s , if $P(A_s|\mathbf{X}_i) > P(A_q|\mathbf{X}_i)$ for any primitive A_q , $A_q \neq A_s$ with $g_{A_s} > \tau$, τ a threshold estimated in training, according to the

likelihood trend of each primitive. Now, given that, at time t_0 , A_s is chosen according to (8), the gradient of the likelihood is:

$$\Delta g_{A_s}(\mathbf{X}_i) = \sum_{h=1}^K p_h(\mathbf{X}_i) \pi_h \Sigma_h^{-1} (\mu_h - \mathbf{X}_i) \quad (9)$$

Here K is the number of components of g_{A_s} , π_h is the mixing parameter and Σ_h^{-1} the precision matrix. Now, as far as the likelihood goes in the direction of the gradient it follows that the action shown must be A_s and as soon as the likelihood decreases it follows that a change in direction is occurring. Therefore the next class has to be identified via (8) and again the gradient is monitored. At the end of the computation a sequence $\langle A_{s_1} : p_{s_1}, \dots, A_{s_k} : p_{s_k} \rangle$, of primitive actions, is returned. Each primitive in the sequence is labelled by the class posterior (according to (8)), computed at the maximum likelihood, reached by the primitive action in the computation window of the gradient.

In Figure 4 the likelihood trend of ten primitive gestures, over 500 frames, is shown. Table 1 illustrates the confusion matrix of the above defined primitives, for an online sequence of 100 gestures. From the confusion matrix it emerges that it is quite unlikely that the system mismatches a direction. However a weakness of the described online recognition algorithm is that it is possible to recognise false directions even if they are not in the performed sequence. In any case the accuracy of the whole system is around 80%, no matters if gestures are performed with varying speeds and by different actors.

4 Actions: Learning and Imitation

According to the steps described in the previous sections, an action can be specified by a sequence of primitive gestures (arm-hand primitive actions). For example the action *manipulation* can be specified by the sequence $\langle Grasp Rot Rel \rangle$. This sequence is recognised using the online estimation of the likelihood trend of each primitive action in the sequence, as gathered in the previous section. However, the same action *manipulation* can be described by $\langle Grasp Rel \rangle$ and by $\langle Grasp Up Rot Down Rel \rangle$, as well. Indeed, we consider a sequence of primitive gestures as a sample from an unknown *regular language*, specifying an action. We make the hypothesis that for each arm-hand action there is a regular language $\mathcal{L}(\mathcal{A})$ generating all the sequences (or words or strings) that specify the action. It follows, by the properties of regular languages, that further complex actions can be obtained by composition, likewise partial actions can be matched within more complex actions.

We face the problem of learning a deterministic finite automaton (DFA) that recognises such a regular language. A DFA is defined by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, with q_0 the initial state; Σ is a finite input alphabet; $\delta : Q \times \Sigma \rightarrow Q$ is the transition function, and $F \subseteq Q$ is the set of final states (see e.g. [24]).

The problem of regular language inference described by a *canonical* DFA, consistent with the given sample, has been widely studied and, in particular, [25] have proposed the Regular Positive and Negative Inference (RPNI) algorithm to infer deterministic finite automata, in polynomial time. Here a canonical representation of an automaton

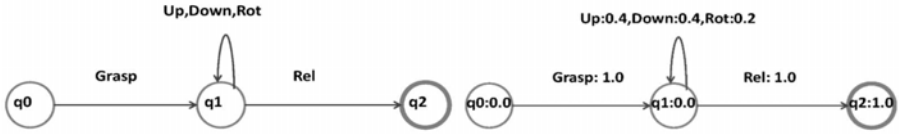


Fig. 5. On the left the DFA of the action *manipulation*, on the right its extended PDFA. Note that, because of the structure of S^+ , $I(q_0) = 1$ and $P_F(q_2) = 1$. The probabilities inside the states indicate the probability of the state to belong to Q_F , the set of final states.

\mathcal{A} is a minimal representation \mathcal{A}' such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$, where $\mathcal{L}(\mathcal{A})$ is the language accepted by the DFA \mathcal{A} .

In this section we briefly show how, using the classification steps of the previous section, it is possible to build a positive and negative sample (S^+ , S^-) of an unknown regular language, such that the sample is structurally complete, that is, the words in the sample make use of all edges, states and final states of the DFA. We also provide a probabilistic extension of the finite automaton, using the annotation of the sequences.

For each action \mathcal{A} , to be learned using the 11 primitives, we define an ordering on the sequences starting with the minimal sequence (e.g. for the manipulation action $\langle \textit{Grasp Rel} \rangle$), and increase the dimension with repeated primitives. Whenever a sequence fails to be recognised then the sequence, with the mismatched primitive, is added to the negative sample. It follows that, according to the recognition performance of the system, we should have 80 positive and 20 negative instances over 100 words of a specific action. Since the positive sample is provided by a benign advisor, it must be structurally complete.

Given (S^+, S^-) the RPNI algorithm starts by constructing an automaton hypothesis $PT(S^+)/\pi_r$, where $PT(S^+)$ is the *prefix tree acceptor* of S^+ . Here π_r is a partition of the prefixes $Pr(S^+)$ of S^+ , defined as $Pr(S^+) = \{u \in \Sigma^* | \exists v \in \Sigma^*, uv \in S^+\}$, where Σ is the alphabet of S^+ . An example of a prefix tree, together with a merging transformation leading to the canonical automaton is given in [25]. Figure 5 illustrates an automaton generated by a sample including the following sequences:

$$\begin{aligned}
 S^+ &= \{ \langle \textit{Grasp Rel} \rangle, \langle \textit{Grasp Down Rel} \rangle, \langle \textit{Grasp Down Rot Rel} \rangle, \langle \textit{Grasp Down Up Rot Rel} \rangle, \\
 &\quad \langle \textit{Grasp Rot Up Rel} \rangle, \langle \textit{Grasp Rot Up Down Rel} \rangle, \langle \textit{Grasp Up Down Rot Rel} \rangle, \} \\
 S^- &= \{ \langle \textit{Rel Rel} \rangle, \langle \textit{Grasp Grasp} \rangle, \langle \textit{Grasp Rel Rot} \rangle, \langle \textit{Grasp Rel Up} \rangle \}
 \end{aligned} \tag{10}$$

Probabilistic extensions of DFA have been treated both from the point of view of probabilistic acceptors [26] and the point of view of automata as generative models of stochastic languages [27][28].

Here, instead, we have assumed that the negative sample comes from the distribution of failures on single elements of the alphabet and we use the distribution on the primitive actions to compute the distribution induced by the identified automaton. Following [27][28], we define the extension $P\mathcal{A}$ of a DFA \mathcal{A} as \mathcal{A} together with the functions $I_A : Q_{init} \mapsto [0, 1]$, $P_A : \delta \mapsto [0, 1]$ and $F_A : Q_F \mapsto [0, 1]$ where $Q_{init} \subseteq Q$ is the set of initial states and $Q_F \subseteq Q$ is the set of final states. If w is a word accepted by \mathcal{A} then

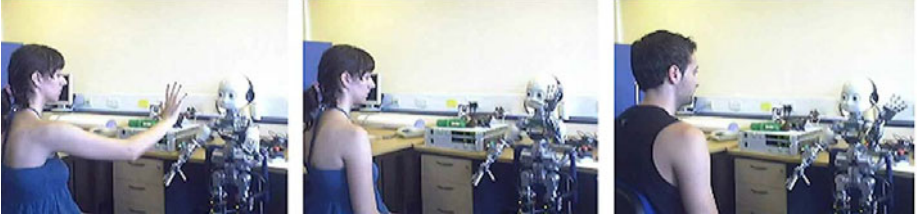


Fig. 6. iCub repeats actions performed by demonstrator

there exists at least a path $\theta \in \Theta_A$, Θ_A the set of paths to final states, from $\exists s_0 \in Q_{init}$ to some $s_k \in Q_F$ and the probability of θ is:

$$Pr_A(\theta) = I_A(s_0) \left(\prod_{i=1}^k P_A(\delta(s_{i-1}, A_i)) \right) F_A(s_k) \quad (s_i \text{ is a current state in the path}) \quad (11)$$

Thus, given a normalization constant α , the probability of generating a word w is

$$Pr_A(w) = \alpha \sum_{\theta \in \Theta_A} Pr_A(\theta) \quad (12)$$

It follows that if an action w is parsed by \mathcal{A} then there exists among the valid paths the most probable one, according to the probabilities estimated in classification, as in HMMs.

In order to add probabilities to states and transitions we proceed as follows. Let $S = (S^+, S^-)$ and let $PT(S^+)$ be the prefix tree acceptor of S^+ . Let $h_j^F = \#\{A_j \in \Sigma \mid A_j \text{ occurs as last symbol of } w, w \in S^+\}$ and $h_j^I = \#\{A_j \in \Sigma \mid A_j \text{ occurs as first symbol of } w, w \in S^+\}$. Now, let $f_{jk} = \#\{q_k \mid \delta(q, A_j) = q_k, A_j \in \Sigma\}$ and $l_{jk} = \#\{A_j \mid (q_k, A_j) \in \delta, q_k \in Q_{init}\}$. Then we define:

$$F_A^*(q_k) = \frac{\sum_j f_{jk}}{\sum_j h_j^F} \quad \text{and} \quad I_A^*(q_k) = \frac{\sum_j l_{jk}}{\sum_j h_j^I} \quad (13)$$

Here F_A^* and I_A^* are intermediate estimations. We recall that each word in S^+ is a labelled sequence according to the classification step. Namely if $w \in S^+$ then $w = A_{j_1} : p_{j_1}, \dots, A_{j_n} : p_{j_n}$. Now, for each branch in the $PT(S^+)$ we construct a transition matrix U_k such that the dimension of U_k is $|Q_A^*| \times |\Sigma|$, with $|Q_A^*| = m$ the number of states in $PT(S^+)$. Here an element u_{ij} of U_k indicates the transition to state q_i in the k -th branch, of the symbol A_j (i.e. of primitive action A_j), in other words it indicates the position of A_j in the sequence accepted by the k -th branch, since by construction q_i is labelled by the prefix of the sequence up to A_j . Thus $u_{ij} = 0$ if there is no transition of A_j to q_i and $u_{ij} = p_{ij}$ if A_j labels the transition to q_i in $PT(S^+)$. Then all these transition matrices are added and normalised. For the normalisation we build a matrix Z which is formed by repetition of a vector V , that is, $Z = V \otimes \mathbf{1}_n^T$, $n = |\Sigma|$. Thus, let $H = \sum_k U_k$, with U_k the matrix of the k -th branch of the $PT(S^+)$ tree. we define $V_U = \sum_j H^j$, that is, the sum of H over the columns. Then $U = H./Z$, with Z the normalisation matrix defined

above, and $\cdot /$ the element wise division between elements of H and elements of Z . Now, in order to obtain the transition probability matrix for \mathcal{A} , at each merge step of state i and state j of the RNPI, assuming $i < j$, we have to eliminate a row u_j . To this end we first obtain the new row $u_i^{new} = (u_i + u_j)/2$ and then we can cross out u_j . It follows that the new matrix U^{new} is $(m - 1) \times n$, n the cardinality of Σ , and it is still stochastic. As this process is repeated for all merging operations in the NRPI algorithm, in the end the last U^{new} obtained will be a stochastic matrix with the right number of transitions.

At this point we are left with the two diagonal matrices F_A^* , I_A^* and the matrix U^{new} . We define the three new vectors, which have all the same dimension:

$$\begin{aligned} V_F &= \text{diag}(F_A^*) \\ V_I &= \text{diag}(I_A^*) \\ V_\delta &= \sum_j U^{(new,j)} \end{aligned} \quad (14)$$

Let $Z = V_F + V_\delta$. We can finally define $F_A = F_A^* \cdot / Z$, $P_A = U^{new} \cdot / Z$ and $I_A = I_A^* / \sum V_I$.

It follows that the requirement for the DFA \mathcal{A} to be a PFA, namely:

$$\begin{aligned} \sum_{q \in Q_A} I_A(q) &= 1 \\ F_A(q) + \sum_{A \in \Sigma} P_A(\delta(q, A)) &= 1 \quad \forall q \in Q_A \end{aligned} \quad (15)$$

is satisfied, see Figure 5. We can note that, by the above construction, each transition δ is labelled according to the sample mean of each primitive action in the sequences mentioned in S^+ . Figure 6 shows sequences of learning and imitation.

5 Conclusions and Acknowledgements

We have described an original method, as far as we know, within the process of real time recognition and learning of direction-based arm-hand actions. Our main contribution is an incremental method that, from attention to human motion to the inference of a DFA, develops a model of some specific human behaviour that can be used to learn and recognise more complex actions of the same kind. For this early system we have chosen simple primitive gestures easily learnable and recognisable with low computational cost. In order to enable the robot to repeat the action, we have extended the DFA to a probabilistic DFA, which generates together with a language a distribution on it. Following the properties of regular languages it is possible to provide a real time learning system that can infer more complex actions. We have finally implemented the system, about which we have shown some specific performance results, on the iCub and tested that the set of actions demonstrated have been learned and replicated efficiently.

The research is supported by the EU project NIFTI, n. 247870.

References

1. Mitra, S., Acharya, T.: Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 37(3), 311–324 (2007)
2. Wildes, R.P., Bergen, J.R.: Qualitative spatiotemporal analysis using an oriented energy representation. In: Vernon, D. (ed.) *ECCV 2000*. LNCS, vol. 1843, pp. 768–784. Springer, Heidelberg (2000)
3. Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. *J. of the Optical Society of America A* 2(2), 284–299 (1985)

4. Braddick, O., O'Brien, J., Wattam-Bell, J., Atkinson, J., Turner, R.: Form and motion coherence activate independent, but not dorsal/ventral segregated, networks in the human brain. *Current Biology* 10, 731–734 (2000)
5. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding* 104(2-3), 90–126 (2006)
6. Poppe, R.: A survey on vision-based human action recognition. *Image and Vision Computing* 28, 976–990 (2010)
7. Aggarwal, J.K., Cai, Q.: Human motion analysis: A review. *Computer Vision and Image Understanding* 73, 428–440 (1999)
8. Bobick, A.F.: Movement, activity, and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London* 352, 1257–1265 (1997)
9. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J.F., Ramanan, D.: Computational studies of human motion: Part I, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision* 1(2/3) (2005)
10. Krüger, V., Kragic, D., Geib, C.: The meaning of action a review on action recognition and mapping. *Advanced Robotics* 21, 1473–1501 (2007)
11. Casile, A., Dayan, E., Caggiano, V., Hendler, T., Flash, T., Giese, M.A.: Neuronal enc. of human kinematic invariants during action obs. *Cereb Cortex* 20(7), 1647–1655 (2010)
12. Gabor, D.: Theory of communication. *J. IEE* 93(26, Part III), 429–460 (1946)
13. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America* 2(7), 1160–1169 (1985)
14. Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology* 58, 1233–1258 (1987)
15. Daugman, J.G.: Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. on ASSP* 36(7), 1169–1179 (1988)
16. Wasserman, L.: *All of Nonparametric Statistics*. Springer, Heidelberg (2005)
17. Heeger, D.J.: Optical flow using spatiotemporal filters. *International Journal of Computer Vision* 1(4), 279–302 (1988)
18. Watson, A.B., Ahumada, A.J.J.: Model of human visual-motion sensing. *Journal of the Optical Society of America A: Optics, Image Science, and Vision* 2(2), 322–342 (1985)
19. Horn, B.K.P., Shunk, B.G.: Determining optical flow. *Art. Intel.* 17, 185–203 (1981)
20. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. of DARPA Imaging Understanding Work*, pp. 121–130 (1981)
21. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Real-time optic flow computation with variational methods. In: Petkov, N., Westenberg, M.A. (eds.) *CAIP 2003*. LNCS, vol. 2756, pp. 222–229. Springer, Heidelberg (2003)
22. Brox, T., Bruhn, A., Papenberger, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
23. Luxburg, U.V.: A tutorial on spectral clustering. *Statistics and Comp.* 14, 395–416 (2007)
24. Hopcroft, J., Ullman, J.: *Introduction to Automata Theory Languages and Computation*. Addison-Wesley, Reading (1979)
25. Oncina, J., García, P.: *Identifying regular languages in polynomial time*. World Scientific Publishing, Singapore (1992)
26. Rabin, M.O.: Probabilistic automata. *Information and Control* 6(3), 230–245 (1963)
27. Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines-part i-ii. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(7), 1013–1039 (2005)
28. Dupont, P., Denis, F., Esposito, Y.: Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. *Pattern Recognition* 38(9), 1349–1371 (2005)

Hand Detection and Gesture Recognition Exploit Motion Times Image in Complicate Scenarios

Zhan Song^{1,2}, Hanxuan Yang^{1,2}, Yanguo Zhao^{1,2}, and Feng Zheng^{1,2}

¹ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

² The Chinese University of Hong Kong, Hong Kong, China

{zhan.song, hx.yang, yg.zhao, feng.zheng}@sub.siat.ac.cn

Abstract. Hand gesture recognition in complicate scenario is still a challenging problem in computer vision domain. In this paper, a novel hand gesture recognition system is presented. To detect the exact hand target from complicate scenarios, the color and motion clues are used to obtain potential hand regions. And then a method named Motion Times Image (MTI) is proposed to identify the optimal hand location. The R-transform descriptor is used to describe the hand shape features and an offline trained Support Vector Machine with Radial Basis Function kernels (RBF-SVM) is exploited to perform the hand gesture recognition task. Extensive experiments with different users under dynamic and complicate scenarios are conducted to show its high recognition accuracy and strong robustness.

1 Introduction

Hand gesture recognition has been an important research topic in Human-Computer Interaction (HCI) domain for its intuitiveness and convenience. Subject to the non-rigid property of hand palm, it usually shows with various shape and appearance. Moreover, in real application, subject to the dynamic and complicate background, robust and efficient hand gesture recognition is still a challenge task in computer vision community [1]. Current hand gesture recognition methods and systems can be classified into two categories. The first class can be called active approach. In such approaches, the user is required to wear some special designed gloves or put some markers on the fingers [2]. Such artificial markers can greatly relieve the difficulty for hand segmentation and its gesture recognition. However, these additional wearing obviously violates the original intention of hand gesture recognition. The other class can be called passive approach. Such approaches work with the naked hand directly. Performance of such system is determined by several issues, such as the hand segmentation, tracking and recognition algorithms etc.

In previous work, a lot of algorithms and systems have been proposed for the naked hand gesture recognition. In [3], a dynamical gesture recognition system is presented. The motion and shape features of human hand are extracted from a spatial and temporal segmentations procedure. But to scenarios with complicate and dynamic background, the precise segmentation of the hand cannot be promised and thus makes the system unstable. In [4], a multi-angle hand gesture recognition system for finger

guessing games is proposed. In the system implementation, three cameras are placed at front, left, and right directions to capture different view angle hand images. To each hand gesture, three SVM classifiers are trained with the images captured from all the three cameras. After the training process, the classifiers were fused to decide the gesture. However, only three hand gestures were tested in the experiment. In [5], the PCA and Gabor filters are used to recognize the American Sign Language (ASL) finger alphabets from hand gesture images. The classification is conducted with a method of fuzzy-c-mean clustering. Whereas, the recognition rate of similar alphabets is relatively low in their approach. In [6], the silhouette motion volume is used to extract the space-time saliency, space-time orientations and weighted moments. And the gesture classification is performed using nearest neighbor algorithm and Euclidean distance. In [7], a novel Human-TV interface is introduced. In this system, the user is required to put the hand on a predefined image region firstly. The recognition algorithm is conducted with the fusion of hand motion and skin color clues. In [8], the proposed method first separates the region of hand motion from complex background images by measuring entropy from adjacent images. Hand gesture is recognized by the approach of improved centroidal profile. However, similar spatial features usually cause false recognitions. Therefore, the number recognizable gesture is very limited. In [9], a hierarchical model-based approach is proposed to track and recognize hand gestures. The features of hand shape and orientation are used in tracking and recognition. Compared with appearance-based methods, model-based methods usually suffer from high computational cost due to the articulated hand motion and the large number of degrees of freedom and hence impractical to real-time applications.

In this paper, we present a robust appearance-based hand detection and gesture recognition system. In the hand detection method, by the using of color and motion clues, possible hand regions are detected firstly. And then, a novel algorithm named Motion Times Image (MTI) is proposed to indentify the exact hand target. The R-transform descriptor [10] is used to describe the shape feature of different hand gestures. An offline support vector machine with radial basis function kernels (RBF-SVM) is trained for the hand gesture recognition. Experiments including robustness and accuracy test are provided to demonstrate the system's high performance under dynamic and complicate background.

The paper is organized as follows. Section 2 presents the MTI based algorithm for hand detection in complicate scenario. The description and recognition algorithms for hand gestures are introduced in Section 3. The experimental results are given in Section 4. Conclusion and future work can be found in Section 5.

2 Hand Detection in Complicate Scenario

Hand detection is usually the first step in all hand gesture recognition systems. To complicate scenario, especially where several hands appear, how to determine the exact hand target is still an open problem. In our method, all the possible hand regions are first detected by using the skin color and motion clues. Consequently, an algorithm named MTI is proposed to identify the exact hand target and its position.

2.1 Skin-Motion Mask Image

Skin color has proven to be a useful clue for computer vision applications, such as face detection [11], localization and tracking of hand [12] etc. In our system, we adopt the same strategy used in [13]. Initially, the input image is transformed from RGB space to YC_bC_r space in terms of separation of luminance and chrominance as well as the compactness of the skin cluster. Then the elliptical model for skin tones in the transformed C_bC_r space which is modified by luminance is used to detect the skin area. The parameters of the skin model and transformation are estimated by training the samples of skin patches of the HHI images [13]. In our skin color detection method, we additionally set a new parameter to regulate the major and minor axes of skin elliptical model. Users of our system can choose the special skin model according to the luminance conditions by adjusting this parameter. The details of skin model and transformation of image are referred to [13]. The resulting mask of skin detection includes hands, faces and other skin-like regions.

The ambiguities introduced by the above skin detection can be solved by a coarse-to-fine strategy with the help of other complementary features. In our method, we propose to further locate the hand in the resulting mask of skin detection by means of motion cues. Specifically, we defined a hand waving motion for users to supplement the skin detection, which needs users to wave the hand naturally.

Generally, motion detection methods can be divided into optical flow and frame difference. Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. However, the computational load of optical flow is quite high. In order to achieve a real-time system, we choose frame difference method to detect motions in our system.

Assume S_i and S_j be the skin mask images of frame I_i and I_j obtained from the former skin detection process, where $i - j = \sigma > 0$. In our experiment, σ is set to 2 empirically. σ is used to control the frame rate. Larger σ corresponds to lower sampling rate. By subtracting the binary image S_j from S_i instead of the gray level image I_j from I_i , the skin-motion mask image can be formulated as:

$$SM_i(x, y) = (S_i(x, y) - S_j(x, y))_+ \quad (1)$$

where $()_+$ denotes the positive entry, (x, y) is the coordinate. The skin-motion mask image denotes the moving parts of image I_i . Therefore, we can conclude that the skin-motion mask image can exclude the static skin-like regions, and only keep hand region as well as some other dynamic skin-like areas.

2.2 Motion Times Image Method for Hand Identification

Motion Energy Image (MEI) [14] and Motion History Image (MHI) [15] are two view-based temporal template approaches for the representation of actions. A binary

MEI is initially computed to act as an index into the action library. It can coarsely describe the spatial distribution of motion energy for a given view of a given action. MHI is a static image template where pixel intensity is a function of the motion history at that location, where brighter values correspond to more recent motion. The inputs of these two representations are image-differencing or silhouette image.

Inspired by the concept of MEI and MHI, a novel representation of hand-waving motion is proposed in this work. Firstly, the skin-motion mask image is used as the input. Secondly, the number of times for the skin-motion appearing at certain locations is used as an index into the action library or a function of the motion history but not the image. We call it Motion Times Image (MTI), which can be formulated as:

$$T_i^\tau(x, y) = \sum_{k=0}^{\tau} SM_{i-k}(x, y) \quad (2)$$

Note that the duration τ is critical in defining the temporal content or frequency of a periodic motion like hand waving. The pixel intensities in motion area are equivalent to the times of waving hand appearing in those locations. τ is set to 10 empirically, which conforms to the natural frequency when people wave their hands. Therefore, the non-periodic and longer-periodic motions will not produce big intensity in MTI. In other words, the bright region in MTI is only the waving hand area. The ultimate hand detection result can be obtained by thresholding the MTI as:

$$W_i(x, y) = \begin{cases} 0 & \text{if } T_i^\tau(x, y) < \alpha \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

The value of α is set to 5 empirically. If $(x_0, y_0)_i$ denotes the centroid of a binary image $W_i(x, y)$, by introducing a window size of $(p, q)_i$, we have:

$$\sum_{y=y_0-q}^{y_0+q} \sum_{x=x_0-p}^{x_0+p} W_i(x, y) / \sum_y \sum_x W_i(x, y) > 0.9 \quad (4)$$

The hand detection result is therefore a sub-region of the skin mask image S_i , where $H_i(x, y) = S_i(x_0 - p + x, y_0 - q + y)$ and $0 < x < 2p, 0 < y < 2q$. Fig. 1 shows one group of experimental result. The first row denotes an image sequence of hand-waving motion. The second row denotes the skin mask images S_i , which include hand, face and other skin-like areas. The third row denotes the skin-motion mask image SM_i . The fourth row shows the constructed motion times image (MTI). The result of the final hand localization in original image is showed in last row.

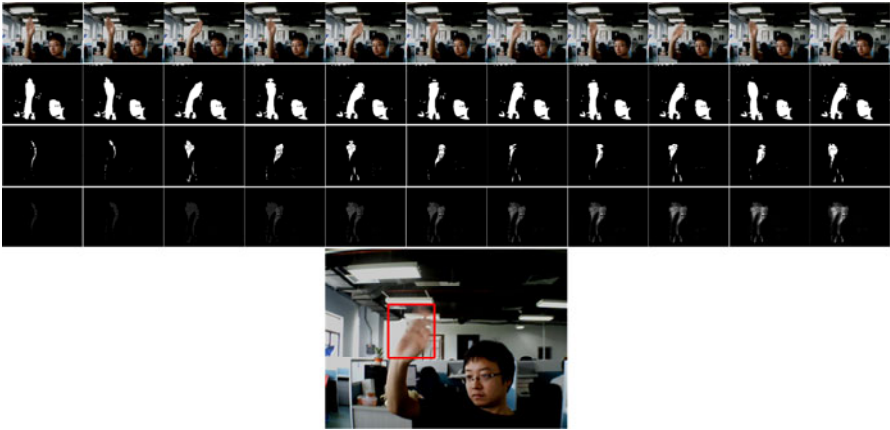


Fig. 1. The proposed Motion Times Image (MTI) algorithm is used to identify the exact hand target and its localization

3 Description and Recognition of Hand Gestures

Shape representation for object recognition includes contour-based descriptors and region-based descriptors. Since contour descriptors are based on the boundary of a shape, they cannot capture the internal structure of a shape. Furthermore, these methods are not suitable to disjoint shapes or shapes with holes because the boundary information is not available. In real applications, precisely contour extraction is a rather difficult task. In this work, a region-based method—R-transform descriptor is adopted to characterize the hand shape. This descriptor calculates on a shape as a whole taking into account all the pixels within the shape.

The R-transform descriptor is invariant to common geometrical transformations and can convert a binary image to a compact signal by the use of the two-dimensional Radon transformation. By definition, the Radon transform of an image is determined by a set of projections of the image along lines taken at different angles. For discrete binary image data, each non-zero image point is projected into a Radon matrix. Given an image H_i , its Radon transform can be defined as:

$$R_i(\rho, \theta) = \sum_x \sum_y H_i(x, y) \delta((x - p/2) \cos \theta + (y - q/2) \sin \theta - \rho) \quad (5)$$

where δ is the Dirac delta function, and (ρ, θ) is the polar coordinate. We discretize the polar coordinate and partition the panel into blocks B_j according to the resolution of image which as shown in Fig. 2 (a). The R-transform descriptor extends the Radon transform by calculating the sum of the Radon transform values in blocks of an image. The normalized representation of hand shape is given by:

$$V_i(j) = \sum_{(\rho, \theta) \in B_j} R_i(\rho, \theta) / \sum_{(\rho, \theta)} R_i(\rho, \theta) \quad (6)$$

The R-transform descriptor of the binary image is invariant to translation and scaling, which means such a descriptor is highly suitable for recognition of shapes. Considering its rotation invariance, we propose to solve this problem through collecting samples spanning various directions. From Fig. 2 (b), we can see that although the hand shape images are with various variations, the descriptors are similar among each other. This characteristic can guarantee our system be used in various situations. After extracting the shape feature, the RBF-SVM algorithm is used as the classifier for gesture recognition. The working flow of the propose system can be described as Fig. 3.

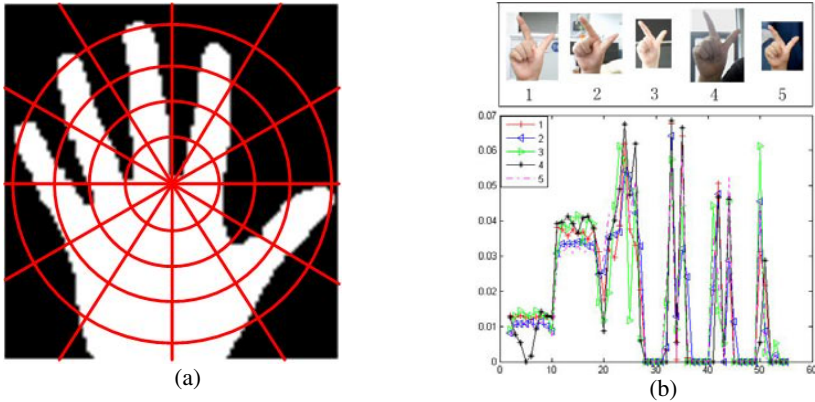


Fig. 2. (a) A sample of hand with partitioning the panel into blocks in the polar coordinate. (b) R-transform descriptors for one kind gesture, the five images are captured with various backgrounds, resolutions and illuminations.

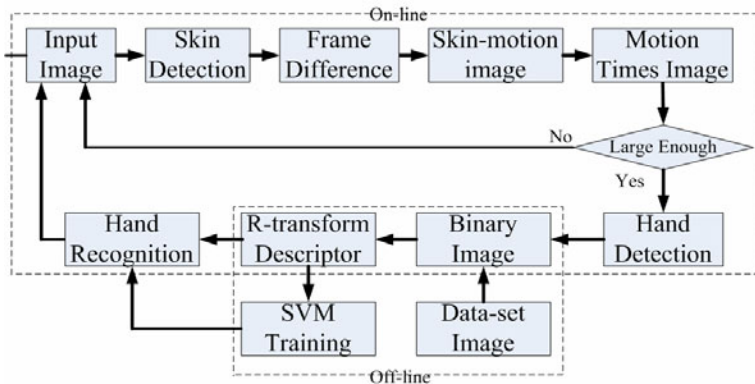


Fig. 3. Work flow of the proposed hand recognition system

4 Experimental Results

To evaluate the performance of the proposed system, a hand gesture database is constructed firstly. The database contains five distinct hand gestures as shown in Fig. 4 (from left to right referred to as A-E). The positive database contains 2316 images.

The gestures are collected from six persons with different size and skin color without particular constraints on illuminations and scales. Left-right reflections of these gestures were also added to the data set. Another negative database is also established, which also contains 2316 images which are the non-distinct gestures. In the algorithm, the RBF-SVM function from OpenCV library [16] with one-versus-rest scheme is adopted to train and classify on the database. All experiments were conducted on an AMD Athlon X2, which equipped with a 2.6G CPU, 1G RAM, and Windows XP OS. The system is running on C++ platform. When the camera resolution is set to 320×240 pixels, it takes about 14ms to process one frame.



Fig. 4. Hand gesture samples in the database. The first row shows five distinct gestures in the positive data set (referred as A-E). The second row shows some non-distinct gestures in the negative database.

4.1 Hand Detection in a Complicate Scenario

To demonstrate the performance of the proposed MTI hand detection method, this experiment is conducted in a normal office environment where three users are waving their hands with different motion patterns (from left to right: long-periodic waving motion, the proposed waving motion, motionless). As shown in Fig. 5, the right hand with defined motion pattern can be detected within several frames.



Fig. 5. Hand detection result via MTI method. The left person in the first image is doing a long-periodic hand waving. The right person is holding a static hand with similar shape to the target hand. The middle person is doing the proposed hand waving. The red square in the last image denotes that our system can identify the right hand without confusion.

4.2 Evaluation of the Hand Gesture Recognition Accuracy

The RBF-SVM classifier with one-versus-rest scheme is implemented on the database. In order to deal with the unbalanced data problem within one-versus-rest method, we randomly selected the same number of negative samples as that of the positive samples when training each of the five classifiers. After the parameters have been optimized via cross validation, our system has been tested by seven different people in an office environment with clutter background. Each one repeated the five distinct gestures several times at various locations. The final classification results are laid out in the confusion matrix as shown in Table 1 which gives a mean accuracy of 97.7%.

Table 1. Confusion matrix using R-transform descriptor and RBF-SVM classifier

		Predicted Gestures				
		A	B	C	D	E
Actual gestures	A	99.88	0.12	0.00	0.00	0.00
	B	0.00	100.0	0.00	0.00	0.00
	C	0.00	3.39	96.04	0.57	0.00
	D	1.74	0.18	0.09	97.99	0.00
	E	0.00	5.02	0.00	0.37	94.61

5 Conclusion and Future Work

In this paper, we have presented a robust hand detection and gesture recognition system. In the hand detection phase, skin color is first used to separate all skin-like regions. And then we proposed a novel method named Motion Times Image to identify the exact hand target. In the hand gesture recognition phase, the R-transform descriptor is used to describe the hand shape feature. We also have built a gesture data set which is used to train our gesture classifier. Consequently, the RBF-SVM classifier with one-versus-rest scheme is used for the recognition task. In the experiments, a database contain 5 distinct hand gestures are established. The database is consisted of a negative and positive dataset, and each one contains 2316 images from 6 different persons. Extensive experiments with different users under dynamic and complicate scenarios are conducted to show its high recognition accuracy and strong robustness. Future work can introduce more robust hand features to further improve the recognition accuracy.

Acknowledgments

The work described in this article was partially supported by NSFC (Project no. 61002040) and Knowledge Innovation Program of the Chinese Academy of Sciences (Grant no. KGXC2-YW-156).

References

1. Wu, Y., Lin, J., Huang, T.S.: Analyzing and Capturing Articulated Hand Motion in Image Sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27(12), 1910–1922 (2005)
2. Mistry, P., Maes, P., Chang, L.: WUW - Wear Ur World: A Wearable Gestural Interface. In: *Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 4111–4116. ACM, New York (2009)
3. Yang, M.H., Ahuja, N., Tabb, M.: Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(8), 1061–1074 (2002)
4. Chen, Y.T., Tseng, K.T.: Multiple-angle Hand Gesture Recognition by Fusing SVM Classifiers. In: *IEEE conference on Automation Science and Engineering*, Scottsdale, AZ, USA, pp. 527–530 (2007)
5. Amin, M.A., Yan, H.: Sign Language Finger Alphabet Recognition from Gabor-PCA Representation of Hand Gestures. In: *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, Hong Kong, pp. 2218–2223 (2007)
6. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-time Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(12), 2247–2253 (2007)
7. Stenger, B., Woodley, T., Kim, T.K., Hernández, C., Cipolla, R.: AIDIA - Adaptive Interface for Display Interaction. In: *Proceedings of British Machine Vision Conference*, Leeds (2008)
8. Shin, J.H., Lee, J.S., Kil, S.K., Shen, D.F., Ryu, J.G., Lee, E.H., Min, H.K., Hong, S.H.: Hand Region Extraction and Gesture Recognition Using Entropy Analysis. *Proceedings of International Journal of Computer Science and Network Security* 6(2), 216–222 (2006)
9. Stenger, B., Thayananthan, A., Torr, P., Cipolla, R.: Model-based Hand Tracking Using a Hierarchical Bayesian Filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28, 1372–1384 (2006)
10. Tabbone, S., Wendling, L., Salmon, J.-P.: A New Shape Descriptor Defined on the Radon Transform. *Computer Vision and Image Understanding* 102(1), 42–51 (2006)
11. Vassili, V., Sazonov, V., Andreeva, A.: A Survey on Pixel-Based Skin Color Detection Techniques. In: *Proc. Graphicon 2003*, pp. 85–92 (2003)
12. Xiong, Y., Fang, B., Quek, F.: Extraction of Hand Gestures with Adaptive Skin Color Models and Its Applications to Meeting Analysis. In: *Proceedings of the Eighth IEEE International Symposium on Multimedia*, pp. 647–651 (2006)
13. Hsu, R.-L., Mottleb, M.A., Jain, A.K.: Face Detection in Color Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(5), 696–706 (2002)
14. Rosin, P.: Thresholding for Change Detection. In: *Proceedings of IEEE International Conference on Computer Vision*, pp. 274–279 (1998)
15. Bobick, A.F., Davis, J.W.: The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(3), 257–267 (2001)
16. <http://sourceforge.net/projects/opencvlibrary> (2010)

Face Verification Using Indirect Neighbourhood Components Analysis

Hieu V. Nguyen and Li Bai

School of Computer Science, University of Nottingham,
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{vhn,bai}@cs.nott.ac.uk

<http://www.nottingham.ac.uk/cs/>

Abstract. Face verification is the task of deciding by analyzing face images, whether a person is who he/she claims to be. This is very challenging due to image variations in lighting, pose, facial expression, and age. The task boils down to computing the distance between two face vectors. As such, appropriate distance metrics are essential for face verification accuracy. In this paper we propose a new method, named the Indirect Neighbourhood Components Analysis (INCA) for learning a distance metric for facial verification. Specifically, INCA is the result of combining ideas from two recently introduced methods: One-shot Similarity learning (OSS) and Neighbourhood Components Analysis (NCA). Our method is tested on the state-of-the-art dataset, the Labeled Faces in the Wild (LFW), and has achieved promising results even in very low dimensions.

1 Introduction

Face verification has been extensively researched for decades. The reason for its popularity is the non-intrusiveness and wide range of practical applications, such as access control, video surveillance, and telecommunication. The biggest challenge in face verification comes from the numerous variations of a face image, due to changes in lighting, pose, facial expression, and age. It is a very difficult problem, especially using images captured in totally uncontrolled environment, for instance, images from surveillance cameras, or from the web. Over the years, many public face datasets have been created for researchers to advance state of the art and make their methods comparable. This practice has proved to be extremely useful.

FERET [1] is the first popular face dataset freely available to researchers. It was created in 1993 and since then research in face recognition has advanced considerably. Researchers have come very close to fully recognizing all the frontal images in FERET [23]. However, these methods are not robust to deal with non-frontal face images.

Recently a new face dataset named the Labeled Faces in the Wild (LFW) [4] was created. LFW is a full protocol for evaluating face verification algorithms.

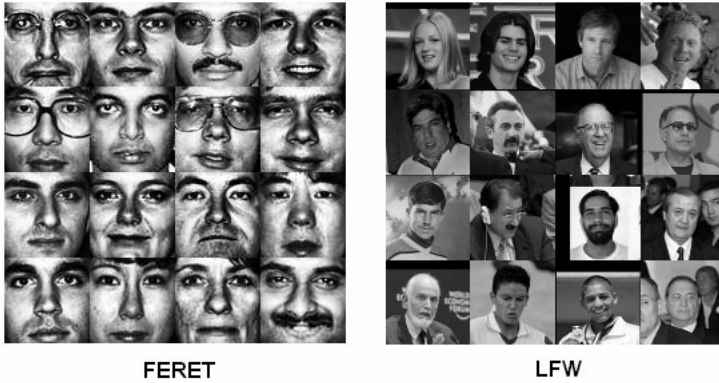


Fig. 1. From FERET to LFW

Unlike FERET, LFW is designed for unconstrained face verification and therefore more challenging. Faces in LFW can vary in all possible ways due to pose, lighting, expression, age, scale, and misalignment (Figure 1). Also, LFW is an open-set protocol in which subjects in testing phase are unknown in training phase [4]. This restriction prevents us from using the popular k-nearest neighbour (kNN) algorithm at the verification step. To overcome this problem, Wolf et al. proposed One-shot Similarity (OSS) learning method in which two testing faces are compared indirectly through faces in a negative set [5]. This method works very well and achieved state of the art results [6,7]. The limitation of OSS is that it only uses training data for selecting the negative set and hence does not take full advantage of the training data.

Goldberger et al. [8] proposed Neighbourhood Components Analysis (NCA), a distance metric learning algorithm especially designed to improve kNN classification. The algorithm is to learn a Mahalanobis distance by minimizing the leave-one-out cross validation error of the kNN classifier on a training set. NCA was applied for face recognition and achieved very good performance [9]. However, NCA is not applicable to the LFW dataset since it requires labeled training data (see section 3.1).

In this paper, we propose a new method named the Indirect Neighbourhood Components Analysis (INCA) which tries to overcome the limitations of OSS and NCA by taking advantages of both. There are two main contributions. The first contribution is that we extend NCA with an additional negative set of face images so that class labels are not required for training. The second contribution is that we propose a variant of the kNN algorithm particularly designed for verification.

The rest of this paper is structured as follows. Section 2 presents the INCA method in detail. Section 3 present how INCA can be applied to face verification. Experimental results are presented in section 4. Finally, conclusion is given in section 5.

2 Proposed Method

The main idea is that we want to apply NCA to face verification without the requirement of labeled training data. To achieve that goal, we propose to use an additional negative set of face images and a variant of kNN algorithm particularly designed for verification.

A negative set is a set of face images of subjects who do not appear in the training and testing sets. We denote this set Z (Figure 2) and its number of elements s . Then given two testing faces with feature vectors x and y , we can conclude they belong to the same person if x is closer to y than any z_j in Z and vice versa. In other words, x is y 's nearest neighbour and y is x 's nearest neighbour. This is similar to nearest-neighbour algorithm for classification.

To apply kNN to face verification, we have modified the classical kNN algorithm. First, we sort all distances from y and z_j ($j = 1 \rightarrow s$) to x in ascending order and define r_{xy} as the ranking of the distance from y to x in that order. As there are $s + 1$ distances, r_{xy} can range from 1 to $s + 1$. Then we can conclude that x and y belong to the same person if $r_{xy} + r_{yx} \leq k$ ($k = 2$ in case of nearest-neighbour). With everything set up, we are ready to present the INCA method in detail.

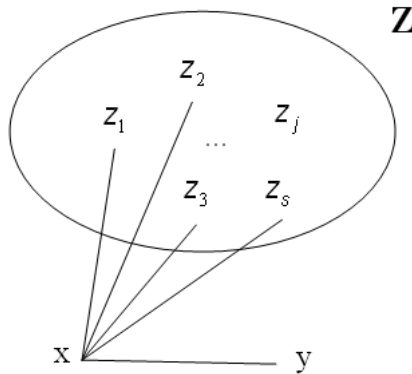


Fig. 2. Negative set

We begin with the problem formalization. Let $\{x_i, y_i, l_i\}$ denote a training set of samples with pairs of input vectors $x_i, y_i \in R^m$ and binary class labels $l_i \in \{1, 0\}$ which indicates whether x_i and y_i match or not. Note that class labels here are not the same as class labels NCA requires for training. In NCA, class labels provide information about subjects' identities. The goal is to learn a linear transformation $A : R^m \rightarrow R^d$ ($d \leq m$) which maximizes the performance of kNN verification in the reduced subspace.

Ideally, we would like to optimize performance on future test data, but since we do not know the true data distribution we attempt to minimize leave-one-out (LOO) performance on the training data instead. Given a finite set of linear

transformations, we can easily select the best one, namely the one that minimizes the number of verification errors. The kNN verification error, however, is a discontinuous function of the transformation matrix A , given that an infinitesimal change in A may change the neighbour graph and hence affect LOO verification performance by a finite amount. Therefore, we cannot use this optimization criteria in this case where there is a continuously parameterized family of linear transformations which must be searched. Instead, we adopt a more well-behaved measure of kNN performance, by using a differentiable cost function based on stochastic neighbour assignments in the transformed subspace. In particular, each input vector x_i selects y_i as its neighbour with some probability $p_{x_i y_i}$. Then we can compute the probability that x_i and y_i match as $\frac{1}{2}(p_{x_i y_i} + p_{y_i x_i})$. Similarly, the probability that x_i and y_i do not match can be computed as $1 - \frac{1}{2}(p_{x_i y_i} + p_{y_i x_i})$. We define $p_{x_i y_i}$ using a softmax over Euclidean distances in the transformed subspace:

$$p_{x_i y_i} = \frac{e^{-\|Ax_i - Ay_i\|^2}}{e^{-\|Ax_i - Ay_i\|^2} + \sum_{j=1}^s e^{-\|Ax_i - Az_j\|^2}} \tag{1}$$

Also, we denote the positive and negative sample index sets by Pos and Neg :

$$Pos = \{i | l_i = 1\}$$

$$Neg = \{i | l_i = 0\}$$

Under the stochastic selection rule (II), we can compute the probability p_i that sample i will be correctly verified:

$$p_i = \begin{cases} \frac{1}{2}(p_{x_i y_i} + p_{y_i x_i}) & \text{if } i \in Pos \\ 1 - \frac{1}{2}(p_{x_i y_i} + p_{y_i x_i}) & \text{if } i \in Neg \end{cases}$$

Therefore, the expected number of training samples correctly verified is:

$$\sum p_i = \sum_{i \in Pos} \frac{1}{2}(p_{x_i y_i} + p_{y_i x_i}) - \sum_{i \in Neg} \frac{1}{2}(p_{x_i y_i} + p_{y_i x_i}) + |Neg|$$

where $|Neg|$ is the number of negative samples. As $|Neg|$ and $\frac{1}{2}$ are constants, we can ignore them to have the following objective function:

$$f(A) = \sum_{i \in Pos} (p_{x_i y_i} + p_{y_i x_i}) - \sum_{i \in Neg} (p_{x_i y_i} + p_{y_i x_i}) \tag{2}$$

Differentiating $f(A)$ with respect to the transformation matrix A yields a gradient rule which we can use for learning. The gradient can be computed as follows:

$$\frac{\partial f}{\partial A} = \sum_{i \in Pos} \frac{\partial(p_{x_i y_i} + p_{y_i x_i})}{\partial A} - \sum_{i \in Neg} \frac{\partial(p_{x_i y_i} + p_{y_i x_i})}{\partial A}$$

$$\frac{\partial(p_{x_i y_i})}{\partial A} = \frac{1}{h_i(A)} \frac{\partial(g_i)}{\partial A} - \frac{g_i(A)}{h_i^2(A)} \frac{\partial(h_i)}{\partial A}$$

where $g_i(A)$ and $h_i(A)$ are the numerator and denominator of $p_{x_i y_i}$, that is:

$$g_i(A) = e^{-\|Ax_i - Ay_i\|^2}$$

$$h_i(A) = e^{-\|Ax_i - Ay_i\|^2} + \sum_{j=1}^s e^{-\|Ax_i - Az_j\|^2}$$

We can continue with:

$$\frac{\partial(g_i)}{\partial A} = -e^{-\|Ax_i - Ay_i\|^2} \times 2A(x_i - y_i)(x_i - y_i)^T$$

$$\frac{\partial(h_i)}{\partial A} = \frac{\partial(g_i)}{\partial A} + \sum_{j=1}^s -e^{-\|Ax_i - Az_j\|^2} \times 2A(x_i - z_j)(x_i - z_j)^T$$

As the roles of x_i and y_i are the same, $\frac{\partial(p_{y_i x_i})}{\partial A}$ can be computed similarly.

The learning algorithm is to maximize $f(A)$ using a gradient-based optimizer such as delta-bar-delta or conjugate gradients. We used the Conjugate Gradient method. Of course, as the objective function $f(A)$ is not convex, some care must be taken to avoid local maxima during training. We have experimentally observed that the linear transformation obtained by Principal Component Analysis (PCA) method can serve as a good starting point for the Conjugate Gradient algorithm.

Note that the norm of matrix A controls the softness of the neighbour assignments. By learning the overall scale of A as well as the relative directions of its rows we are also effectively learning a real-valued estimate of the optimal number of neighbours (k). For example, replacing A with αA , it can easily be shown that as α tends to infinity, the probabilistic assignment is reduced to deterministic nearest-neighbour assignment in the same transformed subspace. In practice, however, it is simpler to estimate the optimal value of k using cross validation. Algorithm 1 describes the proposed method with both training and testing phases. Next we will present how INCA can be applied to face verification.

3 Application to Face Verification

In this section, we show how INCA can be applied to face verification on the LFW dataset in detail.

3.1 LFW Dataset

The dataset contains more than 13,000 images of faces collected from the web. These images have a very large degree of variability in face pose, age, expression, race and illumination. There are two evaluation settings by the authors of the LFW: the restricted and the unrestricted setting. This paper considers restricted setting. Under this setting no identity information of the faces is given. The only information available to a face verification algorithm is a pair of input images

Algorithm 1. The INCA method for verification

*Training:***INPUT**

- $S = \{x_i, y_i, l_i\}$: a set of training samples ($x_i, y_i \in R^m, l_i \in \{0, 1\}$)
- $Z = \{z_j\}$: a negative set of face images
- d : reduced dimension

OUTPUT

- $A_{d \times m}$: output transformation matrix that maximizes the objective function (2)
- k : optimal value of k

1. set initial value for A (e.g. using the PCA method)
2. apply the Conjugate Gradient method to maximize the objective function
3. estimate k using cross validation
4. **return** A and k

*Testing:***INPUT** - (x, y) : two testing faces**OUTPUT** - match/unmatch decision

1. transform x and y to dimension-reduced subspace using the INCA transformation matrix
 2. verify using the proposed kNN variant
-

and the algorithm is expected to determine whether the pair of images come from the same person. The performance of an algorithm is measured by a 10-fold cross validation procedure. See [4] for details.

There are three versions of the LFW available: original, funneled and aligned. In [6], Wolf et al. showed that the aligned version is better than funneled version at dealing with misalignment. Therefore, we are going to use the aligned version in all of our experiments.

3.2 The Negative Set

The negative set is collected from Caltech 10000 Web Faces database [10]. The dataset has 10,524 human faces of various resolutions and in different settings, e.g. portrait images, groups of people, etc. We do not use face images whose sizes are smaller than 30×30 pixels. Hence, the number of faces is reduced to 3,407. These 3,407 faces are normalized to the size of 80×150 (as shown in Figure 3). In our experiments, a thousand faces from these normalized faces are selected randomly to form the negative set.

Face Verification Pipeline. The overview of our method is presented in Figure 4. First, two original images are cropped to smaller sizes. Next some



Fig. 3. Examples of normalized faces from Caltech 10000 Web Faces database

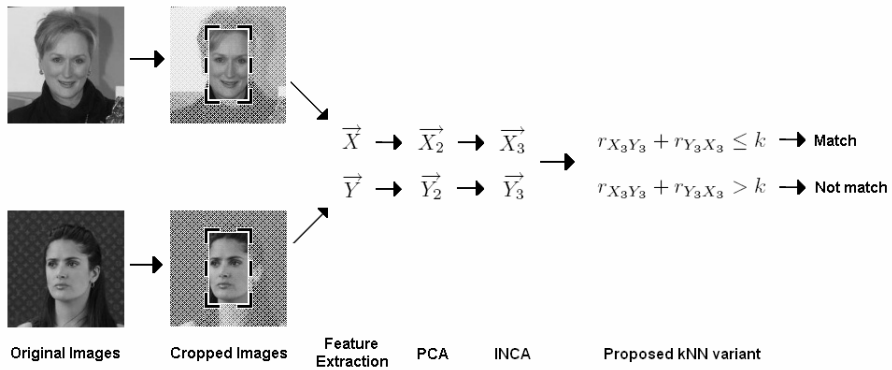


Fig. 4. Overview of Face verification process

feature extraction method is used to form feature vectors (\vec{X}, \vec{Y}) from the cropped images. These vectors are passed to PCA to get two dimension-reduced vectors (\vec{X}_2, \vec{Y}_2) . Then INCA is used to transform (\vec{X}_2, \vec{Y}_2) to (\vec{X}_3, \vec{Y}_3) in the final subspace. Finally, the proposed kNN variant is used to conclude whether the two faces match or not. Each step will be discussed in detail.

Preprocessing. The original size of each image is 250×250 pixels. At the preprocessing step, we simply crop the image to remove the background, leaving a 80×150 face image. The next step after preprocessing is to extract features from the image.

Feature Extraction. To test the robustness of our method to different types of features, we carry out experiments on three facial descriptors: Intensity, Local Binary Patterns and Gabor Wavelets.

Intensity is the simplest feature extraction method. The feature vector is formed by concatenating all the pixels. The length of the feature vector is 12,000 ($= 80 \times 150$).

Local Binary Patterns (LBP) was first applied for Face Recognition in [11] with very promising results. In our experiments, the face is divided into non-overlapping 10×10 blocks and LBP histograms are extracted in all blocks to form the feature vector whose length is 7,080 ($= 8 \times 15 \times 59$).

Gabor Wavelets [12,13] with 5 scales and 8 orientations are convoluted at different pixels selected uniformly with the downsampling rate of 10×10 . The length of the feature vector is 4,800 ($= 5 \times 8 \times 8 \times 15$).

Dimension Reduction. Before applying any learning method, we use PCA to reduce the dimension of the original feature vector to a more tractable number. A thousand normalized faces from the Caltech 10000 Web Faces database are randomly selected to create the covariance matrix in PCA. We notice in our experiments that the specific value of the reduced dimension after applying PCA doesn't affect the accuracy very much as long as it is not too small.

4 Experimental Results

In this section, we present the results of two experiments.

The goal of the first experiment is to test the performance of four methods on three types of features. Four tested methods are Euclidean distance in the original space (Euclidean), Principal Component Analysis (PCA), One-shot Similarity Learning using LDA (OSS-LDA) [5] and our method (INCA). Three types of features are Intensity (IN), Local Binary Patterns (LBP), and Gabor Wavelets (GABOR). As shown in table 1, INCA improves about 5 – 8% over OSS-LDA and about 10 – 15% over Euclidean distance. LBP seems to perform better than Intensity and Gabor Wavelets. Using square root of the feature vector improves the accuracy about 2 – 3% in most cases. The highest accuracy we can get from a single type of feature is 0.8217 ± 0.0046 using INCA with the square root of the LBP feature.

The goal of the second experiment is to test how our method performs in different reduced dimensions. In this experiment, we tested PCA and INCA with the square root of the LBP feature. The reduced dimensions range from 5

Table 1. Mean (\pm standard error) scores on the LFW using different methods

		Euclidean	PCA	OSS-LDA	INCA
IN	original	0.655 ± 0.0067	0.6587 ± 0.007	0.6867 ± 0.0059	0.7423 ± 0.0032
	sqrt	0.6535 ± 0.0061	0.6593 ± 0.0062	0.6718 ± 0.0057	0.7576 ± 0.0057
LBP	original	0.6527 ± 0.0098	0.6767 ± 0.0071	0.7335 ± 0.0051	0.8005 ± 0.0054
	sqrt	0.6977 ± 0.0047	0.7005 ± 0.0062	0.7617 ± 0.0035	0.8217 ± 0.0046
GABOR	original	0.5887 ± 0.0095	0.6083 ± 0.0103	0.7011 ± 0.0057	0.7848 ± 0.0035
	sqrt	0.6335 ± 0.0097	0.6552 ± 0.0075	0.7225 ± 0.0042	0.7916 ± 0.0049

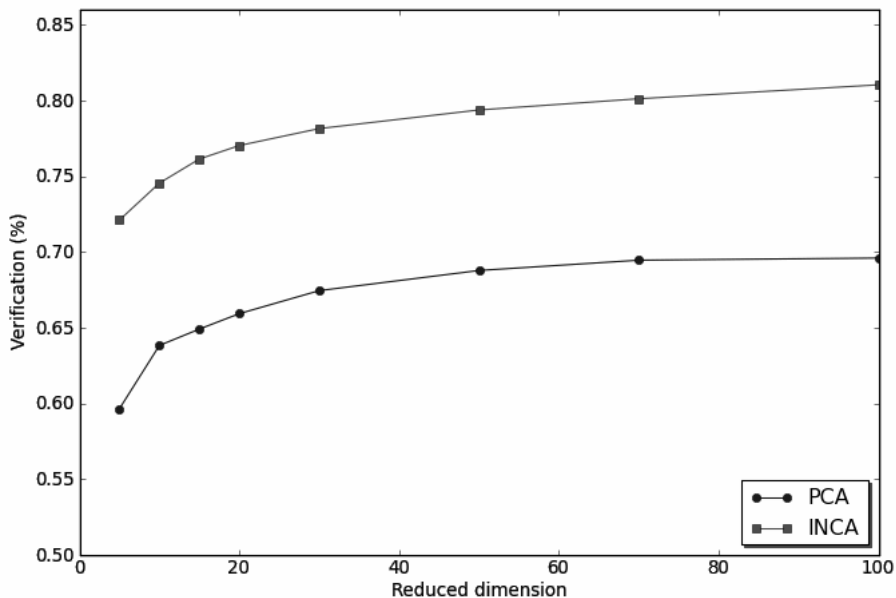


Fig. 5. Mean scores on the LFW using PCA and INCA with different reduced dimensions

to 100. As shown in Figure 5, our method performs well even in 5-dimensional subspace (about 72%) compared to PCA (about 60%).

5 Conclusion

We have introduced a novel method named the Indirect Neighbourhood Components Analysis for learning a distance metric based on the ideas of One-shot Similarity learning and Neighbourhood Components Analysis. Our method uses an addition set of faces images to remove the requirement of labeled training data in NCA and a kNN variant for verification. We tested our method on the LFW dataset and achieved good results, even in very low-dimensional representations.

References

1. Phillips, P., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing* 16, 295–306 (1998)
2. Shan, S., Zhang, W., Su, Y., Chen, X., Gao, W., FRJDL, I., CAS, B.: Ensemble of Piecewise FDA Based on Spatial Histograms of Local (Gabor) Binary Patterns for Face Recognition. In: *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 606–609 (2006)

3. Hieu, N., Bai, L., Shen, L.: Local gabor binary pattern whitened pca: A novel approach for face recognition from single image per person. In: Proceedings of the 3rd IAPR/IEEE International Conference on Biometrics (2009)
4. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst (2007)
5. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305. Springer, Heidelberg (2008)
6. Wolf, L., Hassner, T., Taigman, Y.: Similarity scores based on background samples. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) Computer Vision – ACCV 2009. LNCS, vol. 5995, pp. 88–97. Springer, Heidelberg (2010)
7. Taigman, Y., Wolf, L., Hassner, T.: Multiple one-shots for utilizing class label information. In: The British Machine Vision Conference, BMVC (2009)
8. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighborhood component analysis. In: NIPS
9. Butman, M., Goldberger, J.: Face recognition using classification-based linear projections. EURASIP J. Adv. Signal Process, 1–7 (2008)
10. Angelova, A., Abu-Mostafa, Y., Perona, P.: Pruning training sets for learning of object categories. In: CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 1, pp. 494–501. IEEE Computer Society, Los Alamitos (2005)
11. Ahonen, T., Hadid, A., Pietikainen, M.: Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
12. Daugman, J.: Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression. IEEE Trans. Acoust. Speech Signal Process 36 (1988)
13. Shan, S., Gao, W., Chang, Y., Cao, B., Yang, P.: Review the strength of Gabor features for face recognition from the angle of its robustness to mis-alignment. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 1 (2004)

Efficient Algorithms for Image and High Dimensional Data Processing Using Eikonal Equation on Graphs^{*}

Xavier Desquesnes¹, Abderrahim Elmoataz¹,
Olivier Lézoray¹, and Vinh-Thong Ta²

¹ Université de Caen Basse-Normandie, ENSICAEN, CNRS, GREYC Image Team

² LaBRI (Université de Bordeaux – CNRS) – IPB

Abstract. In this paper we propose an adaptation of the static eikonal equation over weighted graphs of arbitrary structure using a framework of discrete operators. Based on this formulation, we provide explicit solutions for the \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_∞ norms. Efficient algorithms to compute the explicit solution of the eikonal equation on graphs are also described. We then present several applications of our methodology for image processing such as superpixels decomposition, region based segmentation or patch-based segmentation using non-local configurations. By working on graphs, our formulation provides an unified approach for the processing of any data that can be represented by a graph such as high-dimensional data.

1 Introduction

Initially designed for geometric optics, the eikonal equation has become a very popular approach for computer graphics and computer vision with numerous applications. For example, solution of the eikonal equation is used to compute geodesic distances on discrete and parametric surfaces [1,2]. In computer vision, one can quote the shape-from-shading problem [3,4], median axis or skeleton extraction [5], noise removal, feature detection or segmentation [6,7], which can be solved using the eikonal equation.

The eikonal equation is a special case of nonlinear Hamilton-Jacobi partial differential equations and is given by:

$$\begin{cases} H(x, f, \nabla f) = 0 & x \in \Omega \subset \mathbb{R}^m \\ f(x) = \phi(x) & x \in \Gamma \subset \Omega \end{cases} \quad (1)$$

where ϕ is a positive function defined on Ω and $f(x)$ is the traveling time or distance from the source Γ . Then, the eikonal equation can be expressed by using the following Hamiltonian:

* This work was supported under a doctoral grant of the Conseil Régional de Basse-Normandie and of the Coeur et Cancer association in collaboration with the Department of Anatomical and Cytological Pathology from Cotentin Hospital Center.

$$H(x, f, \nabla f) = \|(\nabla f)(x)\| - P(x) \tag{2}$$

where P is a given potential function.

Most of resolutions rely on two main ingredients: a numerical scheme for discretization which leads to nonlinear system and an efficient approach to solve this system. Hamiltonian discretization on Cartesian grids is a well-known problem. Many numerical Hamiltonian schemes can be found in literature, such as Godunov or Lax-Friedrich one [8,6]. In the same way, numerical schemes have been developed in non-Cartesian domain and can be used on both structured and unstructured meshes [1,6]. Other schemes exist for some particular case of two and three dimensional manifolds [2].

The static approach to solve the eikonal equation is usually based on a discretization of the Hamiltonian (2). Numerous methods have been proposed: Rouy and Tourin iterative schemes [3] based on a fixed-point method that solves a quadratic equation (but the complexity behaves as $O(N^2)$ in the worst case), Zhao *fast sweeping* method [9] using a Gauss-Seidel update scheme, Tsitsiklis optimal algorithm with sorted lists of active nodes [10], or the *fast marching* algorithm proposed by Sethian which is the most widely used, to name a few. Another approach to solve (2) is to consider a time-dependent version:

$$\begin{cases} \frac{\partial f(x,t)}{\partial t} = -\|(\nabla f)(x,t)\| + P(x), & x \in \Omega \subset \mathbb{R}^m \\ f(x,t) = \phi(x), & x \in \Gamma \subset \Omega \\ f(x,0) = \phi_0(x). \end{cases} \tag{3}$$

Recently, approaches have been proposed to transcript partial differential equations in a discrete setting by using partial difference equations on graphs [11]. These methods have been applied for image and data processing [12]. With these approaches, the authors in [13] have proposed an adaptation of (3) on weighted graphs of arbitrary topology. Given a graph $G = (V, E, w)$ their discrete analogous of (3) on graphs is

$$\begin{cases} \frac{\partial f(u,t)}{\partial t} = -\|(\nabla_w^- f)(u)\|_p + P(u) & u \in V \\ f(u,t) = \phi(u) & u \in V_0 \subset V \\ f(u,0) = \phi_0(u) & u \in V \end{cases} \tag{4}$$

with V a finite set of vertices, E a set of edges and w is a similarity function defined on edges. $\|(\nabla_w^- f)(u)\|_p$ denotes the \mathcal{L}_p norm of a discrete weighted directional gradient operator defined on graphs (see Section 2 for detailed weighted graphs notations and definitions). One can see that formulation (4) needs numerous iterations due to finite propagation speed and CFL conditions to converge to the solution of the eikonal equation.

A fastest approach to solve the eikonal equation on weighted graphs is to consider the static version of the equation, which can be solved without expensive iterations. Then (4) can be rewritten as

$$\begin{cases} \|(\nabla_w^- f)(u)\|_p = P(u). & u \in V \\ f(u) = 0 & u \in V_0 \subset V. \end{cases} \tag{5}$$

Main contributions

In this work, we propose a static version of the eikonal equation over arbitrary weighted graphs, based on a framework of discrete operators. Explicit solutions of that equation are provided for particular values of $p \in \{1, 2, \infty\}$, and efficient algorithms to obtain such solutions are given. Our general arbitrary weighted graphs formulation provides several advantages. Such a formulation enables the processing of a huge variety of discrete data that can be represented by a weighted graph; i.e. data with any structures or topologies, and embed in spaces of arbitrary dimensions (these dimensions could in fact be very high). Considering image processing, one can find a good approximation of euclidean distance with an appropriate weight function and a large neighborhood [14]. Moreover, the given formulation enables to deal with hierarchical segmentation or textured images segmentation within a unique formulation. This paper also proposes an application to high dimensional data clustering.

Paper organization

The rest of this paper is organized as follows. In Section 2, definitions and notations used in this work are provided. In Section 3, we present our adaptation of the static formulation of the eikonal equation on weighted graphs and provide explicit solutions for $p = \{1, 2, \infty\}$. Finally, efficient algorithms that consider any graphs are presented to obtain these solutions. Section 4 presents experiments that show the potentialities of the proposed methodology. Section 5 concludes.

2 Graph Definitions and Operators on Weighted Graphs

We begin by briefly reviewing some basic definitions and operators on weighted graphs which are the main components of our adaptation of the static eikonal equation.

Notations and Definitions. We assume that any discrete domain can be modeled by a weighted graph. Let $G = (V, E, w)$ be a weighted graph composed of two finite sets: $V = \{u_1, \dots, u_n\}$ of n vertices and $E \subset V \times V$ a set of weighted edges. An edge $(u, v) \in E$ connects two adjacent vertices u and v . The weight w_{uv} of an edge (u, v) can be defined by a function $w : V \times V \rightarrow \mathbb{R}^+$ if $(u, v) \in E$, and $w_{uv} = 0$ otherwise. We denote by $N(u)$ the neighborhood of a vertex u , i.e. the subset of vertices that share an edge with u . In this paper, graphs are assumed to be connected, undirected and with no self loops.

Let $f : V \rightarrow \mathbb{R}$ be a discrete real-valued function that assigns a real value $f(u)$ to each vertex $u \in V$. We denote by $\mathcal{H}(V)$ the Hilbert space of such functions defined on V .

Operators on Weighted Graphs. For better comprehension of the next Section, we now quickly recall some operators on weighted graphs as they are defined in [13].

Considering a weighted graph $G = (V, E, w)$ and a function $f \in \mathcal{H}(V)$, the *weighted discrete partial derivative operator* of f is

$$(\partial_v f)(u) = \sqrt{w_{uv}}(f(v) - f(u)). \tag{6}$$

Two directional partial derivative operators are defined in [13] but, in this work, we restrict ourself to the use of the *internal* one which is defined as

$$(\partial_v^- f)(u) = -\sqrt{w_{uv}} \min(0, f(v) - f(u)). \tag{7}$$

The *weighted directional discrete gradient* $(\nabla_w^- f)(u)$ defined at a vertex $u \in V$ is the vector of all internal partial derivatives:

$$(\nabla_w^- f)(u) = \left((\partial_v^- f)(u) \right)_{(u,v) \in E}^T \tag{8}$$

the corresponding \mathcal{L}_p norm is

$$\|(\nabla_w^- f)(u)\|_p = \left(\sum_{v \sim v} w_{uv}^{p/2} \max(0, f(u) - f(v))^p \right)^{1/p}, \tag{9}$$

and for the \mathcal{L}_∞ norm we have

$$\|(\nabla_w^- f)(u)\|_\infty = \max_{v \sim u} (\sqrt{w_{uv}} \max(0, f(u) - f(v))). \tag{10}$$

One can note that these previous definitions are defined on graphs of arbitrary topology, and can be used to design a general method for solving the eikonal equation on any discrete data sets.

3 Proposed Formulation and Algorithms

In this Section, we present our adaptation of the static eikonal equation over weighted graphs and provide explicit solutions of the equation for particular values of $p \in \{1, 2, \infty\}$. Efficient algorithms are provided to obtain such solutions.

3.1 Eikonal Equation on Weighted Graphs

Starting from the continuous formulation (2) and inspired by its time dependent approach over weighted graphs (4), we obtain a discrete adaptation of the static version of the eikonal equation. Given a graph $G = (V, E, w)$ and a function $f \in \mathcal{H}(V)$:

$$\begin{cases} \|(\nabla_w^- f)(u)\|_p = P(u). & u \in V \\ f(u) = 0 & u \in V_0, \end{cases} \tag{11}$$

where $V_0 \subset V$ corresponds to the initial set of seed vertices. Using norms defined in (9) and (10), we obtain the following equations for the \mathcal{L}_p and \mathcal{L}_∞ norms.

$$\left(\sum_{v \sim v} w_{uv}^{p/2} \max(0, (f(u) - f(v)))^p \right)^{1/p} = P(u), \quad p \in \{1, 2\}. \tag{12}$$

$$\max_{v \sim u} (\sqrt{w_{uv}} \max(0, f(u) - f(v))) = P(u), \quad p = \infty. \tag{13}$$

In next Sections, we present numerical schemes and algorithms to approximate the solution of these equations.

3.2 Numericals Schemes and Algorithms

As the main contribution of this paper, we propose numerical schemes to solve the static eikonal equation on arbitrary graphs ((I2) and (I3)). We emphasize that these schemes can be directly obtained without any spatial discretization since all the operators and functions involved in these equations are discrete.

Now, we study the case where $p \in \{1, 2\}$. With a simple transformation of variables, from (I2) we have

$$\sum_{v \sim u} \left[\frac{(x - f(v))^+}{h_{uv}} \right]^p = P(u)^p \tag{14}$$

where $x = f(u)$, $h_{uv} = \sqrt{1/w_{uv}}$ and $\max(0, x)$ is denoted $(x)^+$. Then (I4) can be rewritten as

$$\sum_{i=1}^n \left[\frac{(x - a_i)^+}{h_i} \right]^p = C^p \tag{15}$$

with $n = \text{card}(N(u))$, $a_i = \{f(v_i) \mid v_i \in N(u) \text{ with } i = 1, \dots, n\}$ and $C = P(u)$. One can remark that (I5) is independent of the graph formulation. Let \bar{x} be the unique solution of (I5). This solution is obtained with an iterative algorithm which uses a sorted list of neighbors $\{a_i\}$. Algorithm uses a temporary variable \hat{x}_i which is computed at the iteration i with the following equation. In the case where $p = 1$:

$$\hat{x}_i = \frac{\left[\sum_{j=1}^{i+1} \left(\prod_{l \neq j, l=1}^{i+1} h_l \right) a_j \right] + \left(\prod_{l=1}^{i+1} h_l \right) C}{\sum_{j=1}^{i+1} \left(\prod_{l \neq j, l=1}^{i+1} h_l \right)}. \tag{16}$$

For the sake of clarity, solution \hat{x}_i for the case where $p = 2$ is not provided but can be obtained similarly.

Finally, the unique solution \bar{x} is equal to \hat{x}_i when $\hat{x}_i \leq a_{i+1}$. The iterative algorithm to compute \bar{x} (for $p = \{1, 2\}$) is summarized in Algo. I.

For the \mathcal{L}_∞ norm formulation (I3), the unique solution \bar{x} can be simply computed by the following equation:

$$\bar{x} = \min_{j=1}^n (a_j + h_j C) \tag{17}$$

where n corresponds to the number of neighbors. One can remark that this equation is a shortest path algorithm (Dijkstra like).

Algorithm 1

```

We know  $\exists k, 1 \leq k \leq n$  such that  $\bar{x}$  is the unique solution of the equation and
 $a_k \leq \bar{x} \leq a_{k+1}$ 
Sort the  $a_i, i = 1, \dots, n$  from the lowest to the greatest values.
 $a_{n+1} \leftarrow \infty$ 
 $m \leftarrow 1$ 
 $\hat{x} \leftarrow \infty$ 
while  $\hat{x} \geq a_{m+1}$  and  $m \leq n - 1$  do
     $\hat{x} \leftarrow$  solution of  $\sum_{i=1}^m \left[ \frac{(x-a_i)^+}{h_i} \right]^p = C^p$  with  $p = 1, 2$ 
     $m \leftarrow m + 1$ 
end while
 $\bar{x} \leftarrow \hat{x}$ 

```

With this formulation, we need a fast and efficient algorithm to compute the solution at each vertex of an arbitrary graph. Many Hamilton-Jacobi solvers can be used to solve (15). The Fast Marching’s updating scheme can be used, but in this paper, we prefer using Jeong and Whitaker Fast Iterative Method (FIM) [15]. The main advantage of this method is to solve the Hamilton-Jacobi equation without expensive data structures.

On an arbitrary graph, FIM consists in an active list of vertices to be updated and initialized with source vertices. Initial solutions are set to 0 for source vertices, ∞ otherwise. At each iteration t , all vertices in the list are updated, i.e. we compute the new solution by solving (11), until convergence: $|\bar{x}_{t+1} - \bar{x}_t| \leq \epsilon$ with $\epsilon \rightarrow 0$. Converged points are removed and their neighbors are added if further updates are needed.

Label propagation

Additionally to the previous algorithm, we propose a simple way to propagate an initial set of labels (from a set of source vertices V_0) through the graph, following the evolution of the propagating fronts. Because our approach allows to compute a distance map with many sources, our distance map becomes a nearest-source distance map on each vertex. Then, the propagating front which arrives at a vertex is necessarily the front coming from the nearest source of the vertex (according to the weight function sense). So, each time a distance is updated on a vertex u , we find the neighbor v of u which is the closest to both u and a seed of V_0 and extend the label of v to the current vertex u . The labeling process can be summarized by the following formula: Each time $f(u)$ is updated, the label $L(u)$ is given by

$$L(u) = L(v) \mid v \in N(u), f(v) < f(u) \text{ and } \frac{f(v)}{w_{uv}} = \min_{z \sim u} \left(\frac{f(z)}{w_{uz}} \right) \quad (18)$$

Complexity

If the graph is totally connected, the complexity of the proposed method is $O(N^3)$ (worst case), where N is the number of nodes in the graph. In practice, we use a sparse k -nearest neighbors graph with $k \ll N$, and the worst case complexity decreases to $O(Nk^2)$. For comparison, the iterative method [13] depends

on two additional parameters: the number of iterations I needed to reach the steady state and the number of initial seeds S . This yield to a complexity of $O(NkIS)$ that is much more higher than ours in practice (since $IS \gg k$).

Relation with other Schemes

As proposed above, our formulation is independent of the graph structure. One can remark that with adapted graph topology and weight function, the proposed formulation is linked to well-known schemes that have been proposed in literature to solve the eikonal equation such as Osher-Sethian or Dijkstra like schemes. In fact, with $p = 2$ and an m -dimensional grid graph, (12) corresponds to the Osher-Sethian discretization scheme. With an unweighted graph and the \mathcal{L}_∞ norm, the Dijkstra like shortest path formulation on graphs can be recovered. Interested readers can refer to [13] for a similar discussion which provide a detailed demonstration.

4 Experiments and Applications

As previously mentioned, our general graph-based formulation allows to deal with any discrete data once they can be represented by graphs. In this Section, we propose some experiments to illustrate that genericity as well as the behavior and the potentiality of such formulation and derived algorithms. All these experiments are processed with a constant potential function $P(u) = 1$. Other potential functions could obviously be used for particular applications.

4.1 Weighted Distances Computation on Graphs

Figure 1 presents the behavior of our formulation for weighted distance computation on arbitrary graphs. For the sake of clarity, the graph used in these experiments was obtained from a grayscale image (Fig 1(a)), with different neighborhoods and weighted functions. Except for the last column, all results are computed with a given weighted function that does not depend on the original image and show the propagating front from the node corresponding to the central pixel of the image (white line are superimposed level-sets). Results (b), (f) and (g) are computed with a 4-adjacency graph, with a constant weight function $f_1 = 1$ and $p = 1, 2, \infty$, respectively. The third and fourth columns show the same distance computation, with respectively 8-adjacency and 16-adjacency for $p = 1$ on the first line and $p = \infty$ on the second. The weight function is provided from [14] and weights each edge in order to reduce the regular-grid metrication errors. One can remark that such graph construction with large neighborhood allows to better approximate the euclidean distance on regular grid. The last column illustrates the weighted distance computation on a 4-adjacency graph using a weight function $f_2 = e^{-\frac{d_{uv}^2}{\sigma^2}}$ which holds the similarity between two nodes u and v . The distance is computed from the node corresponding to the top left pixel of the image and the resulting propagating front for $p = 2$ and $p = \infty$ are shown at Fig 1(e) respectively Fig 1(j). Because the weight function is designed to catch

the topology of the image, the propagating front on the associated graph evolves in a constant way on regular areas (as background or the interior of the apple) and slows down on boundaries. Now, using appropriate graph construction and weight function, we will illustrate the interest of computing such propagating fronts on graphs for image segmentation.

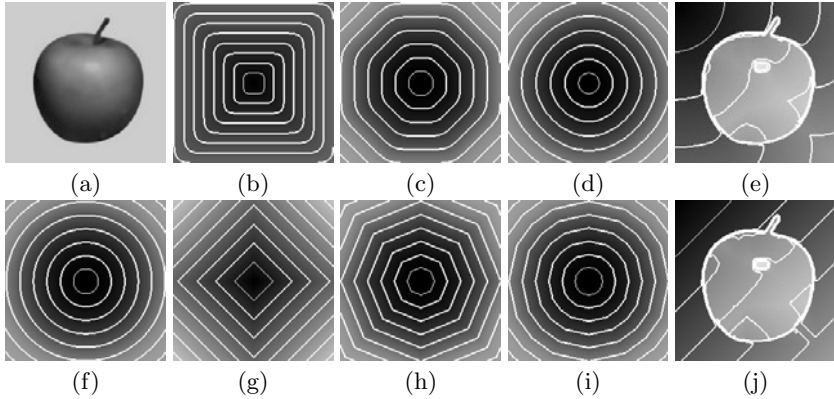


Fig. 1. Illustration of front propagation and weighted distances computation on graph with different configurations (neighborhood), weight functions and p values. Each image represents a weighted distance map from a pixel seed at the center of the image or at the top left corner. See text for details.

4.2 Image Processing

Image Segmentation using Graphs

The objective of the following examples is to illustrate both the genericity of our formulation and the potential of graphs for image processing. Figure 2 presents two images to segment and the associated initial sets of labels. One can remark that each image owns sub-regions of a same class which are not spatially connected (the sky in the first image, or the herd of elephants in the second one). The segmentation is performed on a reduced version of images, a RAG, obtained from a superpixel decomposition. The decomposition and the segmentation are both produced using graphs and our proposed algorithms. Initially developed by Ren and Malik [16], superpixels are an efficient way to reduce image complexity by grouping pixels in a region map while preserving contours. With TurboPixels [17], Levinshtein et al. have proposed an implementation of superpixels in which image decomposition is obtained by dilating a regular grid of seeds so as to adapt to local image structure. As TurboPixels, our implementation uses a regular-grid of seeds but seeds dilation is controlled by our label propagation method (18) instead of iterative evolving equations. In these examples, we use a 4-adjacency grid graph weighted from the similarity between each connected pixels and we compute the propagating front with $p = \infty$. Resulting partitions are shown in Fig. 2. From the obtained partition we associate a weighted RAG coupled with a

2-nearest-symmetric-neighbors graph (2-NSNG) to allow labels to grow beyond local neighborhood. In other words, each node of the RAG is also connected to its two most similar nodes in the whole RAG. Second column of Fig 2 shows these supplementary edges for few nodes (obviously all nodes have additional edges). Then, the same labeling method is performed on these new graphs in order to obtain the final segmentation (third column of Fig 2).

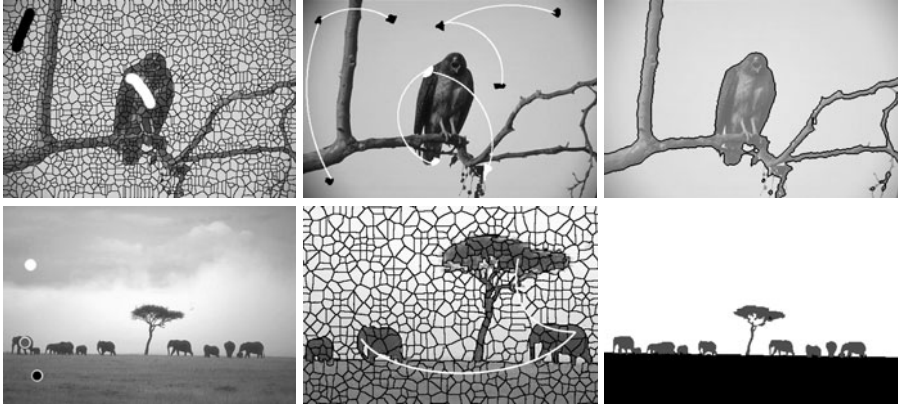


Fig. 2. Image segmentation using graphs (Images are extracted from the Berkeley Segmentation DataSet). The first column presents initial sets of label. The second one shows the RAGs obtained from superpixel decomposition and their 2-NSNG edges (for the sake of clarity additional edges are shown for only few nodes). Third column gives the final segmentations.

One can remark the particularly segmented herd of elephants from a unique label on the left elephant. Finally, we have performed these segmentations only using two successive graph representations of the images and the eikonal equation's based label propagation method proposed in this paper.

Readers interested in semi-supervised segmentation algorithms based on graphs should refer to other recent approaches, as those implementing random walk [18] or graph cuts [19].

Textured Image Segmentation using High Dimensional Pixel Characterization and Large Neighborhood

The following experiments now show advantages of our method to segment images with texture using high-dimensional pixel characterization and large neighborhood i.e. non-local patch based configurations (interested readers can refer to [13] for more details). Figure 3 shows semi-supervised segmentation using two different graphs. These graphs are computed from the initial image (Fig 3(a)) where initial labels are superimposed. The first graph is a weighted 4-adjacency grid graph where each pixel is characterized by its single intensity which only holds a limited local structure of the image. Figure 3(b) shows the segmentation

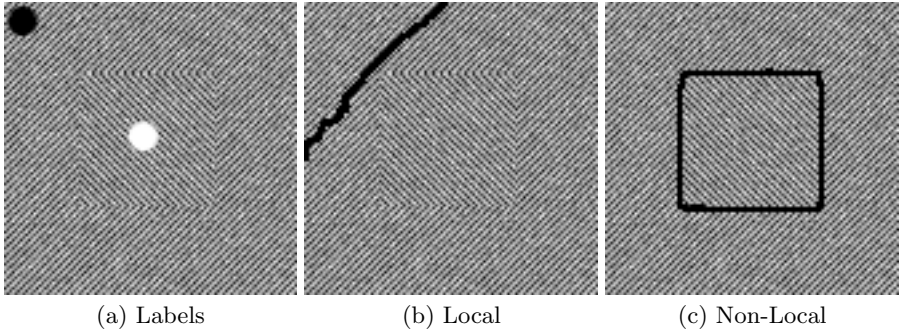


Fig. 3. Segmentation using patches. This figure shows the boundaries between each resulting partition with only local configuration using a 4-adjacency grid graph (b), or local and non-local configuration using a large 24-neighborhood and a patch-based pixel representation in a 25-dimensional space (c).

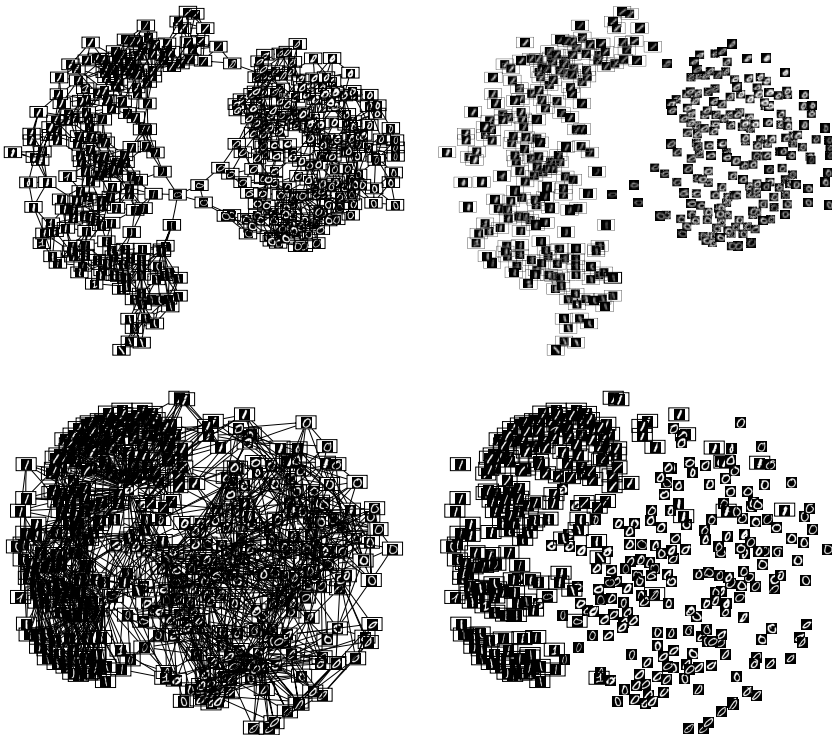


Fig. 4. Handwritten digits database clustering. Each digit is embedded in $\mathbb{R}^{28 \times 28}$. The left column shows initial graphs. Second column shows the final classification obtained by using label propagation on graphs. In the resulting graphs, nodes marked with the first label are surrounded with a square.

result with this graph. To avoid local structure restriction we build a second graph as a 24-neighbors graph, where each pixel is connected to every pixel in a 5×5 window centered on the pixel (excepting itself). In order to characterize texture, each pixel is represented by a vector of \mathbb{R}^{25} , filled with intensity of every pixels in a 5×5 patch centered on the pixel. Figure 3c shows the segmentation result with this graph which incorporates non-local interactions. One can remark advantages of non-local configurations in order to extract the desired object as compared to the local ones.

4.3 High Dimensional Unorganized Data Processing

We now provide experiments of our method with semi-supervised real-world data clustering. Database used in Fig 4 is a sample of 400 images from MNIST database. This database consist in handwritten digit images of size 28×28 . In order to cluster these data, a weighted 3-KNN graph is constructed where edges are weighted with a Gaussian kernel (each vertex is represented by a vector of $\mathbb{R}^{28 \times 28}$, filled with intensity of all image pixels.) A few nodes of each class (1 & 0) are marked with initial labels. First line of Fig 4 shows the graph and the obtained clustering using our label propagation method. In order to introduce some difficulties and show the accuracy of our method, each node of the previous graph is also linked to its most dissimilar node. The resulting graph and the new clusters are shown on the second line of Fig 4. One can remark the good clustering results in both cases, and the interest of the given methodology in order to process high-dimensional unorganized data.

5 Conclusion

In this paper, we proposed a solution of the static eikonal equation over weighted graphs using a framework of discrete operators. We showed that the proposed formulation leads to explicit solutions of the equation for different \mathcal{L}_p norms. Efficient algorithms to compute solutions and a label propagation method using the resolution of the eikonal equation on graphs were also provided. The given experiments have shown the behavior and the potentialities of such methodology applied to image processing and high-dimensional data clustering. Good results for high-dimensional unorganized data clustering could suggest interesting outgoing works as hierarchical graph coarse-gaining in order to simplify databases.

References

1. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. USA, 8431–8435 (1998)
2. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Weighted distance maps computation on parametric three-dimensional manifolds. J. Comput. Phys. 225, 771–784 (2007)

3. Rouy, E., Tourin, A.: A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.* 29, 867–884 (1992)
4. Bruss, A.R.: The eikonal equation: some results applicable to computer vision, pp. 69–87 (1989)
5. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W.: The hamilton-jacobi skeleton. In: *ICCV 1999: Proceedings of the International Conference on Computer Vision*, Washington, DC, USA, vol. 2, p. 828. IEEE Computer Society, Los Alamitos (1999)
6. Sethian, J.A.: Level set methods and fast marching methods - evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. In: *Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, Cambridge (1998)
7. Malladi, R., Sethian, J.A.: A unified approach to noise removal, image enhancement, and shape recovery. *IEEE Trans. On Image Processing* 5, 1554–1568 (1996)
8. Zhang, Y.T., Shu, C.W.: High-order weno schemes for hamilton-jacobi equations on triangular meshes. *SIAM J. Sci. Comput.* 24, 1005–1030 (2002)
9. Zhao, H.: A fast sweeping method for eikonal equations. *Mathematics of Computation* 74, 603–627 (1999)
10. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 40, 1528–1538 (1995)
11. Elmoataz, A., Lézoray, O., Bougleux, S., Ta, V.T.: Unifying local and nonlocal processing with partial difference operators on weighted graphs. In: *Proc. of LNLA*, vol. 44, pp. 11–26 (2008)
12. Bougleux, S., Elmoataz, A., Melkemi, M.: Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *Int. J. Comput. Vision* 84, 220–236 (2009)
13. Ta, V.T., Elmoataz, A., Lézoray, O.: Adaptation of eikonal equation over weighted graph. In: *Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) Scale Space and Variational Methods in Computer Vision*. LNCS, vol. 5567, pp. 187–199. Springer, Heidelberg (2009)
14. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: *ICCV 2003: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, p. 26. IEEE Computer Society, Los Alamitos (2003)
15. Jeong, W.K., Whitaker, R.T.: A fast iterative method for eikonal equations. *SIAM J. Sci. Comput.* 30, 2512–2534 (2008)
16. Ren, X., Malik, J.: Learning a classification model for segmentation. In: *ICCV 2003: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, p. 10. IEEE Computer Society, Los Alamitos (2003)
17. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 2290–2297 (2009)
18. Grady, L.: Minimal surfaces extend shortest path segmentation methods to 3D. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32, 321–334 (2010)
19. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision* 70, 109–131 (2006)

3D DCT Based Compression Method for Integral Images

Ju-Il Jeon and Hyun-Soo Kang*

College of ECE, ChungBuk National University, Chungju, Korea
seventhday@cbnu.ac.kr, hskang@cbnu.ac.kr

Abstract. This paper presents an efficient compression method for the integral images where an adaptive 3D block construction is employed for three dimensional discrete cosine transform (3D-DCT). The existing 3D-DCT based techniques take disadvantages in 3D block construction as applying a fixed block size and a fixed scanning regardless of the image characteristics. In this paper, therefore, we propose more flexible construction, i.e. four different block size modes and two scanning modes which are adaptively selected for coding of elemental images in an integral image. Experimental results show that the proposed method gives significant improvement in coding efficiency. In particular, at the high bit-rates, the proposed method is more efficient since the overhead bits for signalling of a block size and a scanning mode take less part of the total bits.

1 Introduction

Stereoscopy, integral imaging, and holography are well-known techniques in 3D image processing. Integral imaging has been being highlighted as an interesting alternative of stereoscopy and holography. It is a good reason for being highlighted that integral imaging may be more practical than holography and more comprehensive than stereoscopy, considering the amount of information processed and the number of views retrieved.

In integral imaging, image fidelity is crucial since 3D reconstruction quality is influenced by disparity information from elemental images, and hence high fidelity is required to find precise disparities. The standard image coding systems such as JPEG [1] and MPEG [2, 7] can be applied to integral images. However, they may not work efficiently, as they have been optimized to natural images which are so different from integral images in image characteristics. Accordingly, the compression algorithms customized to integral images are required.

There have been some works related to compression of integral images. In [3] and [4], MPEG-2 based compression methods were proposed where elemental images in an integral image were considered as an image sequence. E. Elharar et al. proposed a compression method based on discrete wavelet transform (DWT) followed by discrete cosine transform (DCT) [5]. In addition, 3D-DCT based

* Corresponding author.

compression methods have been studied [6]. To construct 3D blocks, elemental images were placed along to the third dimension, namely z-domain, with the predefined number of elemental images and scanning order. Specifically, 3D blocks of size $8 \times 8 \times 8$ were constructed by placing eight 8×8 blocks located at the same position in eight adjacent elemental images in the predefined scanning order. Then, 3D-DCT was applied to the 3D blocks and the resultant coefficients were quantized and entropy-encoded. Here, it should be noted that this method contains some limitations in terms of block size and scanning order. As a fixed block size and a fixed scanning mode were employed, further performance improvement might be restricted. Hence, we propose more flexible approach for integral images.

The remainder is organized as follows. In the section 2, we briefly describe the conventional 3D-DCT based compression methods. In the section 3, we give the details of the proposed method where the variable block size approach, the adaptive scanning approach, and 3D block modes to combine the two approaches are introduced. In the section 4, we evaluate our method by experiments, and then, in the section 5, we give conclusion remarks.

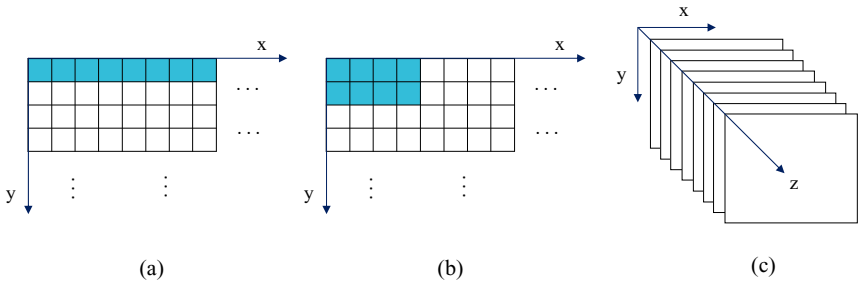


Fig. 1. Two possible assembly of elemental images for 3D-DCT. (a) Horizontal grouping. (b) Horizontal-vertical grouping. (c) A 3D block constructed by grouping.

2 Conventional 3D-DCT Based Method

The standard image compression systems such as JPEG and MPEG have employed the transform coding scheme using DCT which is very effective to achieve energy compaction in the frequency domain. As pixel correlation increases, DCT theoretically approaches the optimal transform, Karhunen-Loeve transform (KLT). Since the integral images have high correlation between elemental images (inter-correlation) as well as between neighboring pixels within a single elemental image (intra-correlation), 3D-DCT may be a good candidate to exploit the inter-correlation and the intra-correlation.

A 3D-DCT based compression for the integral images has been introduced in [6]. To effectively remove the inter-correlation and the intra-correlation, neighboring elemental images were placed in the third dimension. Two methods that place elemental images were introduced, as seen in Fig. 1, which are called horizontal grouping and horizontal-vertical grouping, respectively. It was shown in

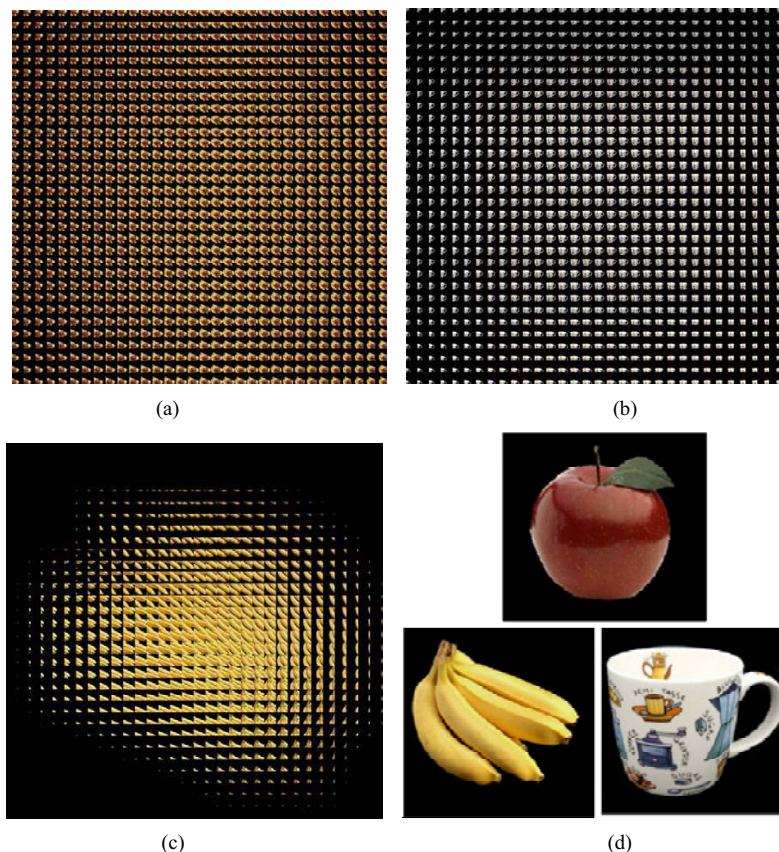


Fig. 2. Integral images (a) Image 1 (b) Image 2 (c) Image 3 (d) Objects used for rendering three images

[6] that 3D-DCT coding using two grouping methods outperforms JPEG. It is a reasonable result since the coding method is specialized for the integral images that have brick structure unlike natural images.

3 Proposed Method

Integral images have a variety of characteristics according to objects in the images as well as image acquisition parameters, while natural images are mostly characterized by objects in the images. Consider the images in Fig. 2, which will be used for performance evaluation in the experimental result section. Fig. 2(a) shows the integral image where an apple object is captured. It has the feature that all of elemental images include the object so that they may be very similar to one another. Fig. 2(b) is similar to Fig. 2(a) in characteristics except that

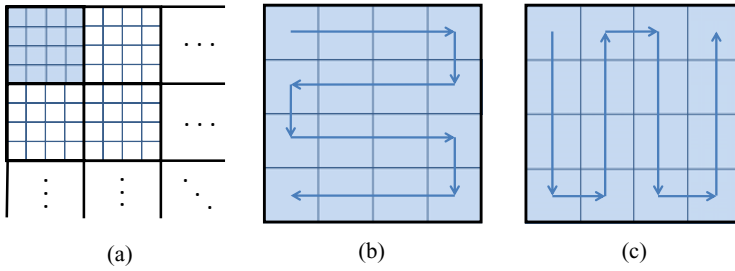


Fig. 3. Basic units and for scanning. (a) basic units: a bold solid box. (b) Horizontal scanning of basic units. (c) Vertical scanning of basic units.

a cup object is used. In contrast, in Fig. 2(c), the integral image where a banana object is captured is composed of elemental images with different features. Some of them contain only a partial part of the object. Extremely, the elemental images in the image boundaries have no object. The differences between these images are induced by different distances to the objects in acquisition.

As seen in these examples, the images have so different characteristics according to acquisition parameters. In addition, they have different features according to objects acquired. Taking into account these characteristics of the images, we should employ an adaptive strategy suitable for each of images to improve the coding efficiency. However, such image characteristics were not considered in the conventional methods based on 3D-DCT where a fixed block size and a predetermined scanning method were applied for construction of 3D blocks. Therefore, we propose a 3D-DCT based compression approach where adaptive block modes and adaptive scanning order are used.

3.1 Adaptive Scanning for 3D Block Construction

The standard image coding systems encode images with regular and repeated behaviors based on a coding unit. In our method, a basic unit for compression is also defined as a set of 4×4 neighboring elemental images, as a box shaded in Fig. 3(a), and then the compression process is performed basic unit by basic unit. As a simple example, assuming elemental images of size $N \times N$, sixteen elemental images can be placed to make a 3D block of $N \times N \times 16$. Then, we perform $N \times N \times 16$ 3D-DCT followed by quantization process and entropy coding. To provide a regularity of the compression process, we fixed the basic unit to 4×4 elemental images such as the macroblock of MPEG where an entire image is regularly partitioned into macroblocks of 16×16 pixels and compression is performed macroblock by macroblock. As a result, 3D blocks in our method are generated in accordance with the basic unit.

Fig. 3(b) and (c) show two scanning options for placing elemental images to form a 3D block. Though we considered other scanning options, Hilbert, perpendicular, and spiral scanning, we removed them from scanning options,

empirically considering trade-off between image quality improvement by adding more options and overhead bits increase required for more scanning options. The coding efficiency can be considerably dependent on the scanning modes since the elemental images are placed in the order of a selected scanning mode to the third dimension. If a scanning mode leads that neighboring elemental images are similar, a smooth change is experienced along to the z domain, which results in efficient energy compaction in the z domain. With the other scanning method, an abrupt change can be developed in the z domain, which causes inefficient energy compaction. Therefore, we should choose more efficient scanning order for efficient compression.

Considering above, we propose an adaptive scanning method where the scanning modes are evaluated and the best mode is selected. The method is applied to all 3D blocks in an image and the best modes are signaled to the decoder so that the decoder can identify the scanning mode of each 3D block.

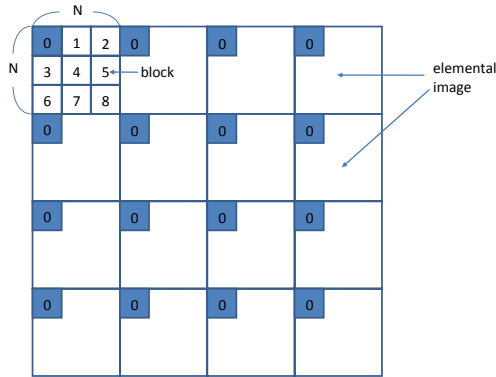


Fig. 4. Structure of a basic unit that contains multiple blocks in a single elemental image ($N=24$)

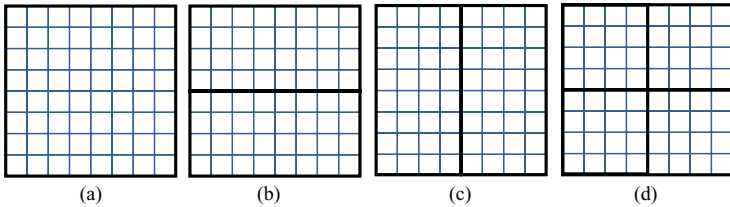


Fig. 5. Four block modes in the x-y image plane. (a) 8×8 block mode. (b) 8×4 block mode. (c) 4×8 block mode. (d) 4×4 block mode4.

3.2 Variable Size 3D Block Construction

We have introduced construction of 3D blocks of size $N \times N \times 16$ where $N \times N$ is the size of elemental images. The construction leads to a single 3D block per

basic unit. However, we do not have to restrict a single 3D block per basic unit. It is well-known that the performance of DCT based compression depends on block sizes and there exists the most efficient block size in terms of coding efficiency and computational complexity. Based on the fact, the standard image coding systems provide various options of block sizes.

In general, a small block is better option than a large block in case of the image areas that contains small objects or regions with high activity like texture. Since integral images have various characteristics according to acquisition circumstances, a variable block size approach may be efficient. Therefore, we employ the approach in 3D block construction.

With the variable block size approach, we may have a number of 3D blocks in a basic unit. An example of construction of multiple 3D blocks is shown in Fig. 4 where the size of each elemental image is 24×24 , and the size of 3D blocks is $8 \times 8 \times 16$. Each elemental image can be partitioned into nine 8×8 blocks which are indexed by 0 to 8 in Fig. 4. Each 3D block is constructed by placing 16 blocks with the same index in the scanning manner described in the previous subsection. Consequently, nine 3D blocks of size $8 \times 8 \times 16$ are created for 3D-DCT. In case that a different block size is applied, the different number of 3D blocks is given. For instance, in case that the elemental image is partitioned into 4×8 blocks, we have eighteenth 3D blocks of $4 \times 8 \times 16$. As explained in this example, we have many different options in block partitioning of the elemental image.

Taking into consideration the practical elemental image sizes and the efficient block sizes employed by the standard image coding systems, we selected four block modes, 8×8 , 8×4 , 4×8 , and 4×4 , shown in Fig. 5, fixing the z domain to be 16.

3.3 The Proposed Method Based on Adaptive 3D Block Construction

Combining the adaptive scanning and the variable size 3D block construction, eight 3D block modes are resulted by the product of two scanning modes and four block size modes. At first, the scanning modes and the variable block size modes are denoted by $S_i, i = 0, 1$, where S_0 = horizontal scan mode, S_1 = vertical scan mode, and $B_j, j = 0, 1, 2, 3$, where $B_0 = 8 \times 8$ mode, $B_1 = 8 \times 4$ mode, $B_2 = 4 \times 8$ mode, $B_3 = 4 \times 4$ mode, respectively. With the notation, we have 3D block mode $M_k = (S_i, B_j)$, where $k = 4i + j$.

Now that we have eight modes, $M_k, k = 0, 1, \dots, 7$, we have to find the best one of eight modes. The optimal solution to the problem is given by the rate-distortion optimization technique where the bit-rates and the distortions resulted by encoding with all allowed coding parameters are used to determine the best one. The technique is very common in the standard image coding systems since it gives the optimal solution in considerations of the rates and the distortions,

though it is computationally intensive. In our method, we also adopt the technique where the cost function for each mode is given by

$$J_k(Q) = D_k(Q) + \lambda(Q)R_k(Q), \quad k \in \{0, 1, \dots, 7\} \quad (1)$$

where Q is a quantization step size, D_k and R_k are the distortion, which is usually the sum of squared difference (SSD), and the number of bits generated by encoding with M_k and Q , respectively, and λ is a Lagrange multiplier which can be conceptually considered to play a role in balancing two terms. Then the optimal mode $M_{k_{opt}}$ is obtained by

$$k_{opt} = \arg \min_k J_k(Q) \quad (2)$$

In our method, according to change of block modes, R_k has considerable variations but D_k has only slight variations as long as Q is fixed. The cost function in H.264/AVC is concerned with motion estimation which causes significant variations for both of D_k and R_k according to macroblock types. However, our method is not as complicated as H.264/AVC. Accordingly, $J_k(Q)$ may be roughly concluded to a function of only rate $R_k(Q)$. That is, we can modify the cost function for simplicity as follows,

$$J'_k(Q) = R_k(Q), \quad k \in \{0, 1, \dots, 7\} \quad (3)$$

Finally, applying $J'_k(Q)$, instead of $J_k(Q)$, to Eq. (2), we can mostly find the best mode. In practical, Eq. (3) provides an important convenience that we do not have to find the optimal values of the Lagrange multipliers for various Q s which are generally given by many experiments for a variety of images.

Regarding the signaling bits, 3 bits are enough to indicate one of eight block modes. The signaling bits are 3 bit fixed-length coded for simple implementation, though they can be signaled more efficiently by variable-length codes since the block modes may be occurred with different probabilities. Conclusively, the bitstream of the proposed method consists of a 3 bit block mode field and a 3D-DCT coefficients' field, where the 3D-DCT coefficients are quantized and entropy coded by the exponential Golomb code for $k = 1$ used in the H.264/MPEG-4 AVC [7].

4 Experimental Results

For performance evaluation, we used the three images in Fig. 2 which were artificially rendered with objects of three objects, a banana, an apple, and a cup shown in Fig. 2 (d). Fig. 2 (a) is an image where the apple object is placed in front of the banana object. Fig. 2 (b) and (c) are the images containing the cup and the banana, respectively. In rendering, we set up the lens array of 32×32 and the image size of 1024×1024 , which allows the elemental image of 32×32 , and the banana object in Fig. 2 (c) was set to be much smaller in size than in Fig. 2 (a), being placed close to a virtual camera.

Table 1. Coding results according to a few fixed block sizes

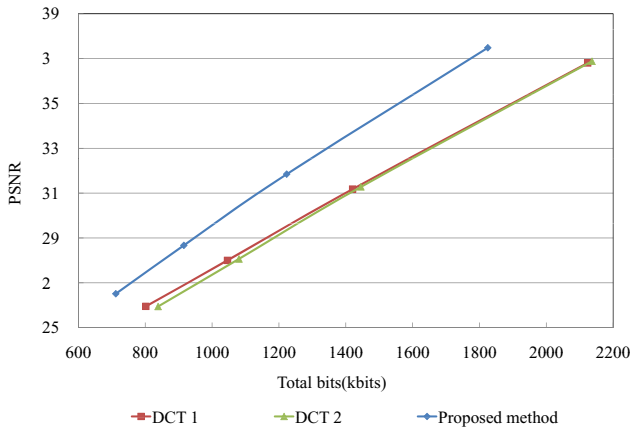
#Test images	Quant step	4x4x16		8x4x16		4x8x16		8x8x16	
		Bits	PSNR	Bits	PSNR	Bits	PSNR	Bits	PSNR
1	16	1869048	37.48	1968070	37.32	2010192	37.25	2093690	37.10
	32	1260616	31.87	1323486	31.58	1349622	31.53	1399784	31.33
	48	949122	28.69	984220	28.37	1004894	28.33	1032910	28.09
	64	745036	26.57	760782	26.25	778834	26.21	788932	25.99
2	16	1712524	38.96	1815346	38.62	1852926	38.56	1960910	38.24
	32	1205146	33.21	1258240	32.87	1280728	32.79	1332014	32.49
	48	944786	29.88	965402	29.59	1004894	29.55	1011376	29.30
	64	769900	27.63	777714	27.40	796538	27.37	806088	27.15
3	16	1297642	39.36	1305466	39.15	1303576	39.16	1309936	38.97
	32	867980	33.69	854928	33.54	854784	33.55	836604	33.43
	48	651376	30.64	625304	30.57	628398	30.59	603800	30.57
	64	515542	28.67	487264	28.68	493120	28.71	465522	28.79

Table 2. BDPSNRs (dB) and BDBRs (%) of the proposed method (PM) against the conventional methods

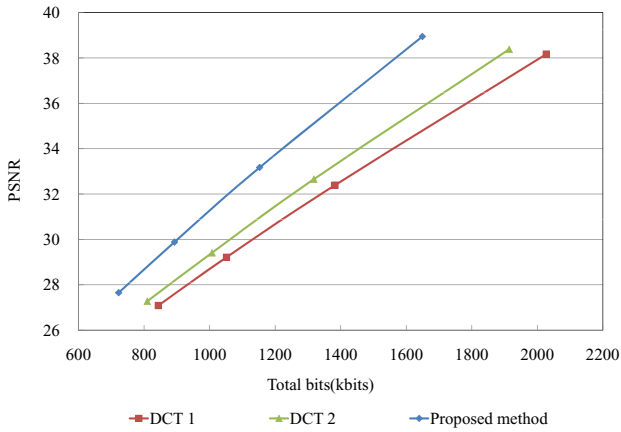
#Test images	PM vs 3D-DCT1		PM vs 3D-DCT2	
	BDPSNR	BDBR	BDPSNR	BDBR
1	2.29	-18.22	2.51	-19.20
2	3.07	-21.20	2.25	-15.72
3	1.50	-13.81	0.24	-2.46

First of all, the coding results for a few different block sizes are shown in Table 1, where four quantization step sizes are applied to obtain evaluation over the wide range of bit-rates. It is seen for the test image 1 and the test image 2 that the small block mode is mostly superior to the large block mode regardless of the quantization step sizes. As the test images contain significant detail, the small block option may be better. In contrast, it is seen for the test image 3 that the large block mode is mostly superior to the small block mode. As a result, we can conclude that adaptive selection of block sizes can improve the coding efficiency rather than the conventional fixed block size approach.

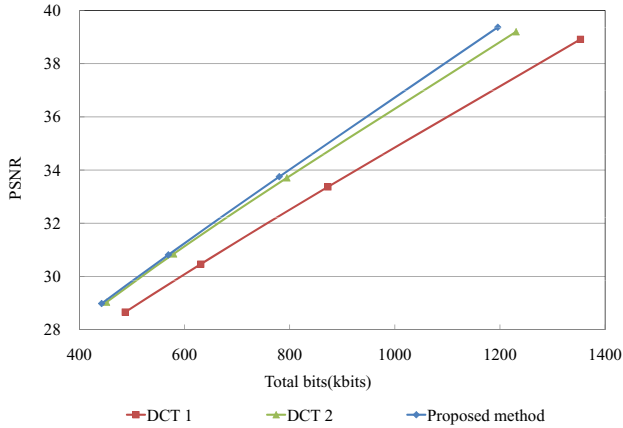
Table 2 shows the results of the proposed method which adaptively applies one of eight block modes. In the table, the 3D-DCT1 and 3D-DCT2 denotes the conventional methods based on the horizontal grouping and the horizontal-vertical grouping, respectively. The BD-PSNR and the BD-BR in the table denote the average PSNR gain over a few fixed bit-rates against the referring method (3D-DCT1 and 3D-DCT2 methods) and the average bit-rate gain over a few fixed PSNRs against the referring method, respectively. As seen in Table 2, the proposed method provides the significant BD-PSNR gains, i.e. 2.29dB and 2.51dB for the test image 1, 3.07dB and 2.25dB for the test image 2, and 1.5dB and 0.24dB for the test image 3.



(a) Test image 1



(b) Test image 2



(c) Test image 3

Fig. 6. PSNR versus the number of bits

The details of Table 2 were plotted in Fig. 6 where it is more obvious that the proposed method considerably outperforms the conventional methods. It should be noted that the proposed method yields more gain at the high bit-rates than at the low bit-rates. It is because the overhead incurred by the proposed method, the signaling bits for the block mode indication, relatively decreases to the total bits as the bit-rate increases.

5 Conclusion

We proposed an efficient compression method for integral images. Combining the adaptive scanning modes and the variable size 3D block modes, we introduced eight 3D block modes. According to the experimental results, the proposed method provides significant gains in spite of the overhead for signaling the block modes. Especially, at the high bit-rates, the proposed method is more efficient since the signaling bits take less part of the total bits. On the other hand, the proposed method has disadvantage in computational complexity as considering various block modes. Consequently, we have further works to develop fast algorithms to determine the optimal block mode.

Acknowledgement

This work was supported by the grant of the Korean Ministry of Education, Science and Technology (The Regional Core Research Program / Chungbuk BIT Research-Oriented University Consortium).

References

1. ITU-T and ISO/IEC JTC 1, Digital Compression and Coding of Continuous-Tone Still Images, Recommendation T.81 and ISO/IEC 10918-1 (1992)
2. ITU-T and ISO/IEC JTC 1, Generic coding of moving pictures and associated audio information-Part 2: Video, Recommendation H.262 and ISO/IEC 13818-2 (1994)
3. Yeom, S., Stern, A., Javidi, B.: Compression of 3D color integral images. *Opt. Exp.* 12, 1632–1642 (2004)
4. Sgouros, N., Kontaxakis, I., Sangriotis, M.: Effect of different traversal schemes in integral image coding. *Appl. Opt.* 47, 28–37 (2008)
5. Elharar, E., Stern, A., Hadar, O.: A Hybrid Compression Method for Integral Images Using Discrete Wavelet Transform and Discrete Cosine Transform. *J. Display Tech.* 3(3), 321–325 (2007)
6. Aggoun, A., Tabit, M.: Data compression of integral images for 3D TV. In: *Proceedings of 3DTV Conference*, pp. 1–4 (2007)
7. ITU-T and ISO/IEC JTC 1, Advanced video coding for generic audiovisual services, Recommendation H.264 and ISO/IEC 14496-10 MPEG-4 AVC (2003)

Plant Texture Classification Using Gabor Co-occurrences

James S. Cope, Paolo Remagnino, Sarah Barman, and Paul Wilkin

Digital Imaging Research Centre, Kingston University, London, UK
{j.cope,p.remagnino,s.barman}@kingston.ac.uk
Royal Botanical Gardens, Kew, London, UK
p.wilkin@kew.org

Abstract. Leaves provide an important source of data for research in comparative plant biology. This paper presents a method for comparing and classifying plants based on leaf texture. Joint distributions for the responses from applying different scales of the Gabor filter are calculated. The difference between leaf textures is calculated by the Jeffrey-divergence measure of corresponding distributions. This technique is also applied to the Brodatz texture database, to demonstrate its more general application, and comparison to the results from traditional texture analysis methods is given.

1 Introduction

In the field of comparative biology, novel sources of data are continuously being sought to enable or enhance research varying from studies of evolution to generating tools for taxon identification. Leaves are especially important in this regard, because in many applied fields, such as studies of ecology or palaeontology, reproductive organs, which may often provide an easier form of identification, are unavailable or present for only a limited season. Leaves are present during all seasons when plants are in growth. There are also millions of dried specimens available in herbaria around the world, many of which have already been imaged. While these specimens may possess reproductive organs, the main character features are often concealed in images through being internal or due to poor preparation. However, almost all specimens possess well-preserved and relatively easily imaged leaf material.

Traditional methods employed by botanists for describing leaves rely on terminology and are largely qualitative and open to some level of interpretation [8]. In recent decades plant science has begun to use a range of quantitative morphometric methods in comparative studies [20,13]. However, such data currently exist for a small minority of plant taxa, largely due to the limitations imposed by manual data capture.

In recent years there has been an increased interest in applying computer vision techniques to the problem of plant classification. Most of these studies have involved the analysis of leaf shape [7,22,11] or venation patterns [9,16,18], with

leaf texture having been largely ignored. Backes et al. have applied multi-scale fractal dimensions [1] and deterministic tourist walks [2] to plant classification by leaf texture, although their experiments involved very limited datasets (just five species in the latter case) and so may not work as well for a wider range of plant species. Casanova et al. [4] used Gabor filters on a larger dataset and achieved reasonable results, whilst Liu et al. have presented a method based on wavelet transforms and support vector machines [17]. Generalized Fourier descriptors were applied to leaf images acquired using a scanning electron microscope [14], although this data capture method is impractical for most purposes due to the specialist equipment required.

This paper presents a method for plant texture classification based on the joint distributions of Gabor filter responses. Section 2 describes a simple method for extracting consistent texture samples from leaves. Our method of texture analysis and classification is given in section 3. In section 4 details of experiments using our method and a number of traditional methods on both leaf texture datasets and the popular Brodatz dataset [3] are given, with the results presented and discussed in section 5.

2 Plant Texture Extraction

Much of the texture present on a leaf is due to the venation, with other sources of texture including hairs and glands. This venation can be separated into two main groups: the low order (primary and secondary) vein framework, and the higher order vein fabric. If texture samples (windows) are extracted randomly from a leaf, the level and quality of the vein framework present in a sample may vary greatly, and the sample may contain leaf damage, depending on the precise position of the sample on the leaf. For these reasons, we suggest a simple method of extracting samples which as far as possible contain only the vein fabric, as the contents of these samples should be more consistent. (figure 1).

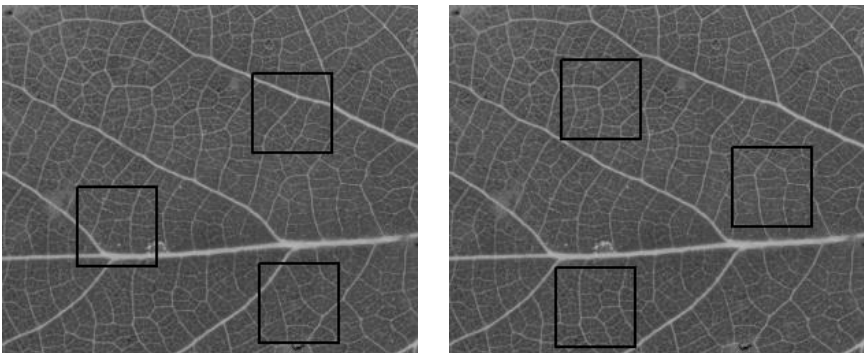


Fig. 1. Random sampling (left) compared with desired sampling (right)

The first stage is to reduce the scale of the image by convolving it with a Gaussian kernel and then sub-sampling. This has the effect of smoothing out much of the detail in the vein fabric, whilst retaining the main venation. Next, the image background, the paper on which the leaf is mounted, is removed. This can be done using Otsu's thresholding method [19]. An edge detection operator is then applied to the foreground of the image to provide a rough measure of the areas with strong edges in this scale space. A large number of potential windows are sampled at random from the foreground (containing only the leaf) and are sorted according to the sum of the squared edge magnitude for all the pixels within the window. The desired number of non-overlapping sample windows with the lowest sum can then be selected for use. A number of examples of windows selected by this method are given in figure 2.

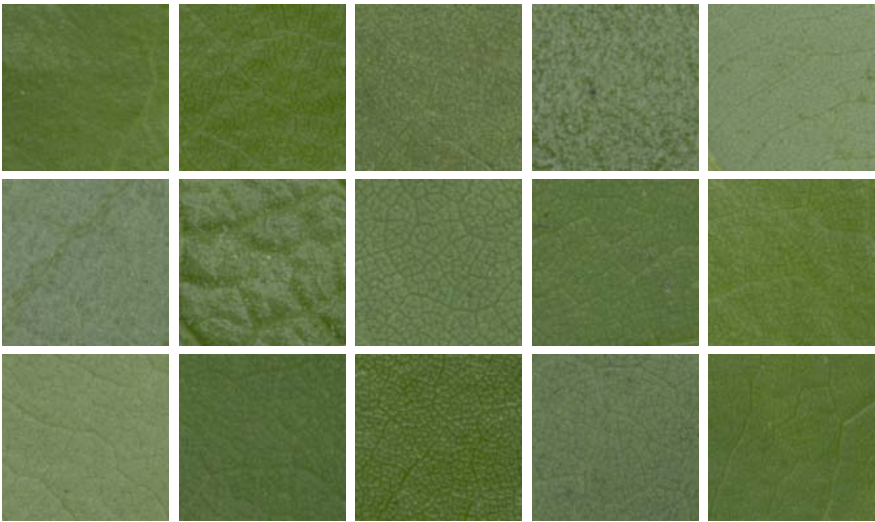


Fig. 2. Extracted texture samples from 15 species of *Quercus* (Oak)

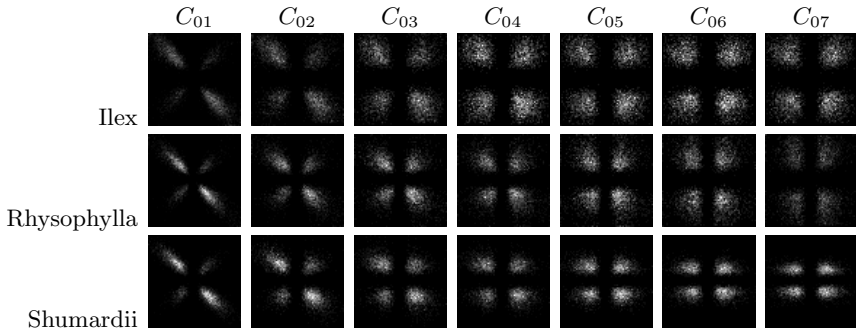
3 Texture Analysis and Classification

3.1 Gabor Filters

The texture analysis method presented in this paper is based around the joint distributions of Gabor filters. A Gabor filter [6] is essentially a sinusoid modulated by a Gaussian function. It can be expressed as follows:

$$G(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(\frac{2\pi x'}{\lambda} + \psi\right) \quad (1)$$

Table 1. Examples of Gabor co-occurrence matrices for 3 species of *Quercus*



Where:

- $x' = x \cos \theta + y \sin \theta$
- $y' = y \cos \theta - x \sin \theta$
- θ is the orientation of the filter.
- γ is the filter aspect ratio.
- σ is the standard deviation of the Gaussian.
- λ is the wavelength of the sinusoid.
- ψ is phase offset.

Gabor filters have been applied to a large range of computer vision problems including image segmentation [21] and face detection [12]. Of particular interest are the links found between Gabor filters and the human visual system [5].

3.2 Texture Analysis from Gabor Co-occurrences

A bank of 128 Gabor filters is created, where for filter G_{mn} , $\sigma = 1.5 * 1.2^{m-1}$, $\lambda = \frac{\sigma\pi}{2}$ and $\theta = \frac{n\pi}{16}$, with $m = 0..7$ and $n = 0..15$ referring to the filter scale and angle respectively. For all filters, $\gamma = 1$ and $\psi = 0$. The full set of filters is applied to each texture, but for each scale only the value corresponding to the highest absolute value for all the orientations is recorded for each pixel. This ensures that the method is rotation invariant. The results of the filtering for an image are combined into a series of co-occurrence matrices [10], whereby for each pair of scales, the resulting matrix describes the probability of a pixel producing one response value for the first scale, and another for the second.

$$C_{kl}(i, j) = \sum_x \sum_y \begin{cases} 1, & \text{if } g_k(x, y) = i \text{ and } g_l(x, y) = j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Where $g_m(x, y) = \max_{n=0..15} (G_{mn}(x, y) * I(x, y))$ is the maximum response from convolving the filters for scale m with the image I at point (x, y) , and (i, j) is a pair of response values. Examples of these matrices are given in table 1.

3.3 Classifying Textures

To classify textures, the corresponding co-occurrence matrices for different textures are directly compared. This is done by treating the co-occurrence matrices as probability distribution functions (pdfs), by simply dividing each value by the sum of all values, and using the Jeffery-divergence distance measure. For two pdfs, f_a and f_b , the distance between them, $JD(f_a, f_b)$, is calculated as follows:

$$JD(f_a, f_b) = \sum_i \sum_j f_a(i, j) \log \frac{2f_a(i, j)}{f_a(i, j) + f_b(i, j)} + f_b(i, j) \log \frac{2f_b(i, j)}{f_a(i, j) + f_b(i, j)} \quad (3)$$

The distance between two images A and B is then:

$$D(A, B) = \sum_k \sum_{l, l \neq k} JD(C_{kl}^A, C_{kl}^B) \quad (4)$$

Where C_{kl}^A and C_{kl}^B are respectively the co-occurrence matrices at scale k, l for images A and B . The final classification is performed using the the k -nearest neighbour method, with $k = 3$. The modal class of the 3 closest texture samples to the one being classified is chosen. In the case that all 3 classes are different, the class of the single closest texture sample is used instead. This strategy was chosen as it reduces the risk of classification errors due to outliers.

4 Experiments

4.1 Datasets

The method was evaluated using four texture datasets. The first dataset was constructed using the method described in section 2. For each of 8 leaves from 32 different species, 8 64×64 windows were selected. This window size was chosen to allow the windows from leaves with dense vein frameworks to fit between the main veins. Eight windows were then used to provide an adequate overall sample size, whilst more would require more computation and may not be possible for particularly small leaves. Each of the 8 samples for a leaf was filtered before they were combined into a single set of co-occurrence matrices. The second dataset used 8 windows sampled at random from the same leaves, to illustrate the value of our texture extraction method.

The remaining two datasets came from the Brodatz texture database [3], with the first of these using a 40-class subset of the full 111 texture classes used in the second on these sets. From each of the classes, 9 200×200 non-overlapping samples were selected. The second of these sets is particularly difficult, due to the weak intra-class homogeneity present in some classes [15].



Fig. 3. Randomly extracted texture samples from 15 species of Quercus

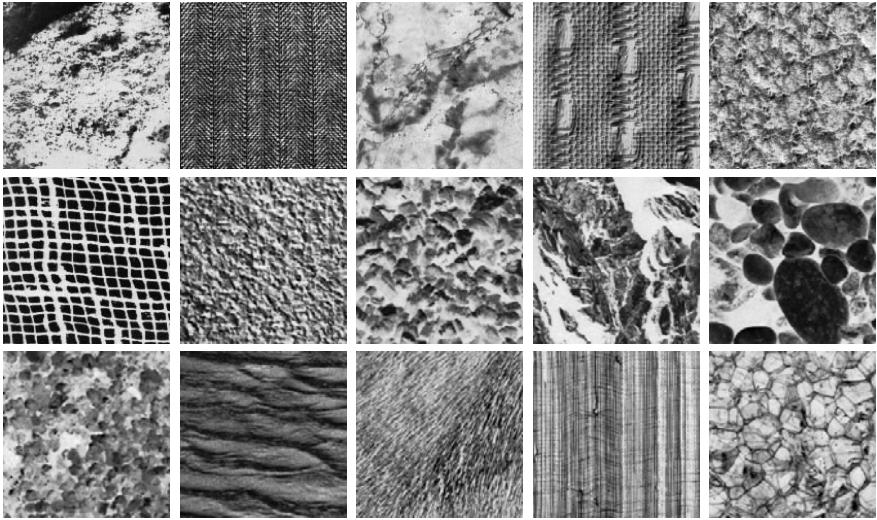


Fig. 4. Fifteen texture samples from the Brodatz dataset

4.2 Comparison Methods

For comparison, the above datasets were also used with a number of traditional texture analysis methods:

– Fourier Descriptors:

The Fourier Transform of each image was calculated. From this, a vector of 64 features was found, whereby the i^{th} feature $f_i = \sum_{\theta=0}^{\pi} F(i\frac{w}{64}, \theta)$, where $F(r, \theta)$ is the Fourier Transform in polar form, and w is half of the image width.

– Gabor Filters:

The set of Gabor filters used in section 3.2 is applied to each image. The energy in each resulting image is then calculated as $e_{\sigma\theta} = \sum_x \sum_y (G_{mn}(x, y) * I(x, y))^2$. The set of energies for each scale are then averaged resulting in 8 rotationally invariant features. This is similar to the approach used in 4

– Co-occurrence Matrices:

The traditional co-occurrence matrices were produced, using angles of $0rad$, $\frac{\pi}{4}rad$, $\frac{\pi}{2}rad$ and $\frac{3\pi}{4}rad$ and distances of 1,2 and 3. For each distance, a set of 14 textural features is calculated, as described by Haralick 10.

5 Results

The results for the experiments are given in tables 2 and 3. For the two leaf datasets, all the algorithms performed better on the dataset created as described in section 2, showing the value of our method of leaf texture extraction. For all datasets our method performed best, with the basic Gabor method performing worse. For the leaf datasets, the Fourier descriptors outperformed the co-occurrence matrices, whilst for the Brodatz datasets, co-occurrence matrices did better. It seems likely that the Fourier method is better at capturing finer detail, whilst the co-occurrence matrices perform best in images with higher contrasts between nearby pixels. This is supported by the greater improvement in quality for the Fourier method between the two leaf datasets.

Table 2. Results for the two leaf datasets

	Vein Fabric	Random
Our Method	85.16	79.69
Gabor	50.78	45.70
Fourier	82.42	62.89
Co-occurrence Matrices	69.14	61.72

Table 3. Results for the two Brodatz datasets

	40 Class Subset	Entire Brodatz
Our Method	97.50	95.50
Gabor	63.61	52.55
Fourier	77.50	74.47
Co-occurrence Matrices	87.50	81.18

6 Conclusions

This paper has presented a method for texture classification that outperforms a number of traditional methods. It was found to be effective in the difficult task of classifying plants based on leaf texture, for which extracting texture samples from the vein fabric was shown to produce better results. The method also achieved high classification rates on the Brodatz texture database, performing only slightly worse on the entire 111 classes than on a 40 class subset.

References

1. Backes, A.R., Bruno, O.M.: Plant leaf identification using multi-scale fractal dimension. In: Foggia, P., Sansone, C., Vento, M. (eds.) *Image Analysis and Processing – ICIAP 2009*. LNCS, vol. 5716, pp. 143–150. Springer, Heidelberg (2009)
2. Backes, A.R., Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Texture analysis and classification using deterministic tourist walk. *Pattern Recognition* 43, 685–694 (2010)
3. Brodatz, P.: *Textures: A Photographic Album For Artists And Designers*. Dover Publications, New York (1966)
4. Casanova, D., de Mesquita Sá Jr., J.J., Bruno, O.M.: Plant leaf identification using gabor wavelets. *International Journal Of Imaging Systems And Technology* 19, 236–243 (2009)
5. Daugman, J.G.: Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research* 20, 847–856 (1980)
6. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A* 2, 1160–1169 (1985)
7. Du, J.-X., Huang, D.-S., Wang, X.-F., Gu, X.: Shape recognition based on radial basis probabilistic neural network and application to plant species identification. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) *ISNN 2005*. LNCS, vol. 3497, pp. 281–285. Springer, Heidelberg (2005)
8. Ellis, B., Daly, D.C., Hickey, L.J., Johnson, K.R., Mitchell, J.D., Wilf, P., Wing, S.L.: *Manual Of Leaf Architecture*. Cornell University Press (2009)
9. Fu, H., Chi, Z.: Combined thresholding and neural network approach for vein pattern extraction from leaf images. In: *IEE Proceedings of Vision Image And Signal Processing*, vol. 153, pp. 881–892. Institution of Electrical Engineers (2006)
10. Haralick, R.M., Dinstein, I., Shanmugam, K.: Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics SMC-3*, 610–621 (1973)
11. Hearn, D.J.: Shape analysis for the automated identification of plants from images of leaves. *Taxon* 58, 934–954 (2009)
12. Huang, L.L., Shimizu, A., Kobatake, H.: Robust face detection using gabor filter features. *Pattern Recognition Letters* 26, 1641–1649 (2005)
13. Jensen, R.J., Ciofani, K.M., Miramontes, L.C.: Lines, outlines, and landmarks: Morphometric analyses of leaves of *acer rubrum*, *acer saccharinum* (aceraceae) and their hybrid. *Taxon* 51(3), 475–492 (2002)
14. Journaux, L., Destain, M.F., Miteran, J., Piron, A., Cointault, F.: Texture classification with generalized fourier descriptors in dimensionality reduction context: An overview exploration. In: Prevost, L., Marinai, S., Schwenker, F. (eds.) *ANNPR 2008*. LNCS (LNAI), vol. 5064, pp. 280–291. Springer, Heidelberg (2008)

15. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. *Transactions on Pattern Analysis And Machine Intelligence* 27, 1265–1278 (2005)
16. Li, Y., Chi, Z., Feng, D.D.: Leaf vein extraction using independent component analysis. In: *IEEE International Conference On Systems, Man, And Cybernetics*, pp. 3890–3984. IEEE, Los Alamitos (2006)
17. Liu, J., Zhang, S., Deng, S.: A method of plant classification based on wavelet transforms and support vector machines. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) *ICIC 2009*. LNCS, vol. 5754, pp. 253–260. Springer, Heidelberg (2009)
18. Mullen, R., Monekosso, D., Barman, S., Remagnino, P., Wilkin, P.: Artificial ants to extract leaf outlines and primary venation patterns. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) *ANTS 2008*. LNCS, vol. 5217, pp. 251–258. Springer, Heidelberg (2008)
19. Otsu, N.: A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics* 9, 62–66 (1979)
20. Plotze, R.d.O., Falvo, M., Padua, J.G., Bernacci, L.C., Vieira, M.L.C., Oliveira, G.C.X., Martinez, O.: Leaf shape analysis using the multiscale minkowski fractal dimension, a new morphometric method: A study with *passiflora* (passifloraceae). *Canadian Journal of Botany* 83(3), 287–301 (2005)
21. Sandler, R., Lindenbaum, M.: Gabor filter analysis for texture segmentation. In: *Computer Vision And Pattern Recognition Workshop*, p. 178. IEEE, Los Alamitos (2006)
22. Shen, Y., Zhou, C., Lin, K.: Leaf image retrieval using a shape based method. *Artificial Intelligence Applications And Innovations* 187, 711–719 (2009)

A Compressive Sensing Algorithm for Many-Core Architectures

A. Borghi¹, J. Darbon², S. Peyronnet¹, T.F. Chan³, and S. Osher⁴

¹ LRI, INRIA, Université Paris Sud, Orsay, F-91405

² CMLA, ENS Cachan, CNRS, PRES UniverSud, France

³ Hong Kong University of Science and Technology, Hong Kong

⁴ UCLA Mathematics Department, USA

Abstract. This paper describes a parallel algorithm for solving the l^1 -compressive sensing problem. Its design takes advantage of shared memory, vectorized, parallel and many-core microprocessors such as Graphics Processing Units (GPUs) and standard vectorized multi-core processors (e.g. quad-core CPUs). Experiments are conducted on these architectures, showing evidence of the efficiency of our approach.

1 Introduction

Compressive Sensing [1,2] has received a lot of interest both from a theoretical and algorithmical point of view due to its capacity of reconstructing a sparse signal from a very limited number of measures. Although the primary purpose of such an approach is signal processing oriented, it has also been successfully applied for image processing purposes such as face recognition [3], image filtering [4] and medical imaging [5]. More precisely, we are interested in solving the following l^1 constraint minimization problem:

$$\begin{cases} \min_u \|u\|_1 \\ s.t. Au = f \end{cases} \quad (1)$$

where $u \in \mathbb{R}^n$ is the signal to reconstruct while $f \in \mathbb{R}^m$ is the measurement and $A \in \mathbb{R}^{m \times n}$ is a compressive sensing matrix [1,2] with $m \ll n$. The purpose of this paper is to describe a simple and efficient algorithm designed for taking advantage of parallel capabilities of standard processors.

There is a large literature for solving this problem such as [2,6,7]. Many of these algorithms rely on an iterative thresholding/shrinkage procedure that is iterated until convergence. Our approach also falls into this class but we focus on the design of the approach such that it yields an efficient implementation on parallel many-core architectures. More precisely, our algorithm relies on the proximal point that is derived from a Moreau-Yosida regularization of the energy [8]. This approach guaranties the convergence of the algorithm toward the optimal solution. Our main idea consists in designing a specific Moreau-Yosida regularization that yields to a simple and efficient algorithm that allows a simple

and efficient parallel implementation. The remainder of the paper is as follows. We briefly describe multi-core CPU and GPU architectures in Section 2. Our algorithm is described in Section 3 and experiments are presented in Section 4.

2 Parallel Architectures

Our goal is to design an algorithm that benefits from parallel architectures such as vectorized multi-core CPU and modern GPU. We present here general concepts about parallel computing and implementation issues.

Coarse parallelism. Using several computing units to deploy tasks allow to improve performance. However, developers often prevent memory conflicts thanks to locks. It is well known that this method deteriorate performances. To avoid locks, another way is to schedule memory accesses so that they never interfere with each others. This general issue is known as the synchronization problem. Modern CPUs we have considered (Intel Core i7) have four cores and GPUs (NVIDIA GTX 275) have ten multiprocessors. Each core or multiprocessor features several execution units, such as vector units.

Vectorization. Another level of parallelism consists of processing the data as a vector. This is called vectorization. Performances can be greatly improved through the use of vector units, assuming that data are well aligned and structured. To take benefit from this technique an algorithm must be intrinsically built on vectorial primitives. Besides, the alignment property is also mandatory, and is often taken into account from the design phase of the algorithm. Modern CPUs have 128-bit vector units, enabling some instructions to work simultaneously on four 32-bit (single precision) operands.

Memory considerations. The main constraint of modern parallel architectures is the memory. Memory limitations are bandwidth and latency issues. These platforms have two kinds of memory : cache memory, which is typically structured in different levels, and standard RAM shared by the cores. Cache memory has for goal to reduce the time to access data in RAM. Cache memory is an order of magnitude faster than RAM but also smaller. Data parts are transferred back and forth between cache memory and RAM during computations. To avoid memory issues, the structure of the cache must be taken into account. Usually, this is done by scheduling in the algorithm data transfers, and by managing directly the memory in the source code. A poor memory management leads to the so-called computation starvation where a processor is waiting for data and thus is not computing. Each Intel Core i7 core has its own L1 and L2 cache and the Core i7 has a shared L3 cache for all of its cores. GPU cores have access to a very fast embedded memory of hundreds of MB. It allows an efficient use of the highly parallel and bandwidth-demanding units of GPUs. Standard RAM can also be accessed (but the transfers are very costly).

3 A Proximal Point Algorithm

This section is devoted to the design of our algorithm. First, we briefly present the Moreau-Yosida regularization before describing the proximal point and our algorithm.

Moreau Yosida Regularization. We consider a penalization method for solving the original problem. It yields to minimize

$$E_\mu(u) = \|u\|_1 + \frac{\mu}{2} \|Au - f\|_2^2, \tag{2}$$

where μ is a positive number that enforces the constraint $Au = f$. Following [8] a Moreau-Yosida regularization of the above energy consists in considering the following energy for any given point $u^{(k)}$:

$$F_\mu(u^{(k)}) = \inf_{u \in \mathbb{R}^n} \left\{ E_\mu(u) + \frac{1}{2} \|u - u^{(k)}\|_M^2 \right\}, \tag{3}$$

where M is a symmetric positive definite matrix. Following [8], this new energy is strictly convex and thus has a unique minimizer called the proximal point of $u^{(k)}$. Iterating the proximal point converges toward an optimal solution of (2) for a fixed μ (see [9] for a proof).

A specific proximal operator. Note that the convergence of the proximal point holds as long as M defines a metric. The versatility of such a condition allows us to design a specific one such that it yields an efficient implementation on parallel architectures. Optimality condition for problem (3) corresponds to find $u^{(k+1)}$ such that

$$s(u^{(k+1)}) + (\mu A^t A + M)u^{(k+1)} = \mu A^t f + Mu^{(k)},$$

where $s(u^{(k+1)})$ is a subgradient of $\|\cdot\|_1$ at the point $u^{(k+1)}$. This problem becomes easy to solve when it can be carried out dimension by dimension, i.e., it is separable. This means that we wish the matrix $(\mu A^t A + M)$ to be a diagonal one. Note that such a matrix yields an implementation that enjoys the technological requirement presented in Section 2 since parallelism follows straightforwardly and data can be easily reordered such that aligned for vectorized operations holds.

Separability amounts to define M such that it kills the non-diagonal elements of $\mu A^t A$. Besides, recall that we need to have M to define a metric, i.e., M should be positive definite, in order to ensure the convergence of the algorithm toward an optimal solution. In this paper we restrict ourself to the case where the eigenvalues of $A^t A$ are 1 or 0 (extension to the general case easily follows by considering the maximal eigenvalue instead of 1. This maximal eigenvalue can be estimated using a power iteration method for instance). Thus we define M as follows

$$M = (1 + \epsilon)\mu Id - \mu A^t A,$$

where ϵ is a small positive real number to make M positive definite. Then, optimal solution of problem (3) is given by a standard thresholding scheme [6,7]: if $|\mu A^t f + M u^{(k)}| > 1$, $u^{(k+1)} = \frac{1}{(1+\epsilon)\mu}(\mu A^t f + M u^{(k)} - \text{sign}(\mu A^t f + M u^{(k)}))$, where $\text{sign}(x) = \frac{x}{|x|}$ if $x \neq 0$, otherwise $u^{(k+1)} = 0$. Note that only matrix/vector multiplications and standard vectorized operations are needed for computing the update. Also, note that one wishes to set ϵ as small as possible for faster convergence. Setting ϵ to 0 empirically leads to convergence although the proof presented here does not hold for this case.

Our algorithm. So far, we have described how to minimize E_μ when μ is set to some non-negative value. In order to solve the original problem (1) we embed the partial optimization of E_μ into a continuation process that essentially consists in approximately minimizing E_μ for a series of increasing μ (recall that we wish μ as big as possible to enforce the constraint.) In this paper, μ will successively takes the values $\mu, 2\mu, \dots, 2^{l_{max}}\mu$, where l_{max} is a strictly positive integer.

We start with the null signal as an initial guess. Instead of starting with an arbitrary initial μ we look for one that is small enough so that convergence is fast but also big enough since otherwise the optimal solution remains to the null signal. Such an initial μ is searched through a dichotomic process that selects the smallest μ that yields a non-zero signal.

There are two stopping criteria for stopping the approximate minimization of E_μ . First, we wish to have the current solution close to the constraint in the following sense: $\frac{\|A u^{(k+1)} - f\|_2}{\|f\|_2} \leq e_{tol}$. Second, the variation between two consecutive solution must be small enough in the following sense: $E(u^{(k)}) - E(u^{(k+1)}) < e_{consec}$. The two values e_{tol} and e_{consec} are two parameters of the algorithm.

The whole algorithms is thus as follows:

1. Set $k = 0$ and $u^{(0)} = 0$
2. Apply the bitonic strategy for getting the initial μ
3. For $l = 1$ to l_{max} (number of continuation values μ)
 - (a) Do
 - (i) Set $k \leftarrow k + 1$
 - (ii) Compute $u^{(k+1)}$
 - (b) While $(E_\mu(u^{(k)}) - E_\mu(u^{(k+1)})) > e_{consec}$ and $(\frac{\|A u^{(k+1)} - f\|_2}{\|f\|_2} > e_{tol})$
 - (c) $\mu \leftarrow 2\mu$
4. Return $u^{(k+1)}$

4 Experimental Results

In this section, we describe the choices we have made in order to implement our algorithm on several parallel platforms. Then, we assess the efficiency of our method, running on these many-core platforms, through several experiments.

It should be noted that the operations involving the sampling matrix A are the most time consuming operations of our method. These operations can be

performed in two different ways depending on the nature of A . Either A is described explicitly and standard matrix/vector multiplication are needed; or A can be represented implicitly with the help of a transform. A typical example for the latter case is when A is a sub-matrix of the Discrete Fourier Transform (FFT), or Discrete Cosine Transform (DCT). Both representations are compatible with coarse and fine-grained parallelism implementations.

Storing matrices explicitly allows for flexibility in the design of the matrix. The drawback is that explicit representation are memory consuming. Moreover, matrix/vector multiplication needs $O(n \times m)$ operations, where $n \times m$ is the size of the matrix, and is also time consuming. On the contrary, the use of the implicit form is memory-wise. Indeed, it is then no longer needed to store A (and its transpose A^t). Besides, it generally allows for faster computations because of fast available transforms. However, considering only sampling matrices with implicit representation is a limitation on the variety of problems that can be adressed.

Here, we consider these two kinds of matrices. We consider orthogonalized Gaussian matrices where their elements are generated using i.i.d normal distributions $\mathcal{N}(0, 1)$ and where their rows are orthogonalized. We have also used partial DCT matrices. They are generated through a uniform sampling of m rows from the full Discrete Cosine matrix.

Orthogonalized Gaussian matrices, being explicitly represented, need a large amount of memory. Since GPUs are limited by their memory size, only CPU platforms can deal with interesting sizes of such matrices. On the contrary, partial DCTs can be used on GPU platforms. For the sake of fairness, we implement only partial DCTs on both platforms. This partial DCT has been implemented using a complex-to-complex Fast Fourier Transform with an additional $O(n)$ time for converting the results to real numbers.

We now present architecture-dependent implementation details. The code for the CPU is parallelized using OpenMP (Open Multi-Processing Application Programming Interface) and we vectorized the implementation using SSE instructions. We chose a thresholding approach (see section 3) for its separability property (i.e., each element can be processed independently), therefore parallel and vector computing increase the efficiency of the implementation. FFT computations are performed with FFTW 3.1 and we used the Intel C compiler 10.1.

The GPU implementation is based on CUDA (Compute Unified Device Architecture) 2.0. Because of PCI Express bus bandwidth limitations, we designed our implementation such that it uses very few data communications between central RAM and GPU embedded memory. This is a very common problem in General-Purpose computing on GPUs.

We now describe the experimental conditions of our experiments. We then present numerical results that show the effectiveness of our approach. We use several experimental platforms. The first one is an Intel Core 2 Quad Q6600 2.4GHz with 8MB of L2 cache and 4GB of RAM. The second processor is an Intel Core i7 920 2.66GHz with 1MB of L2 cache, 8MB of L3 cache and 6GB of RAM. We also use a NVIDIA GEFORCE 8800 GTS with 96 cores and 640MB

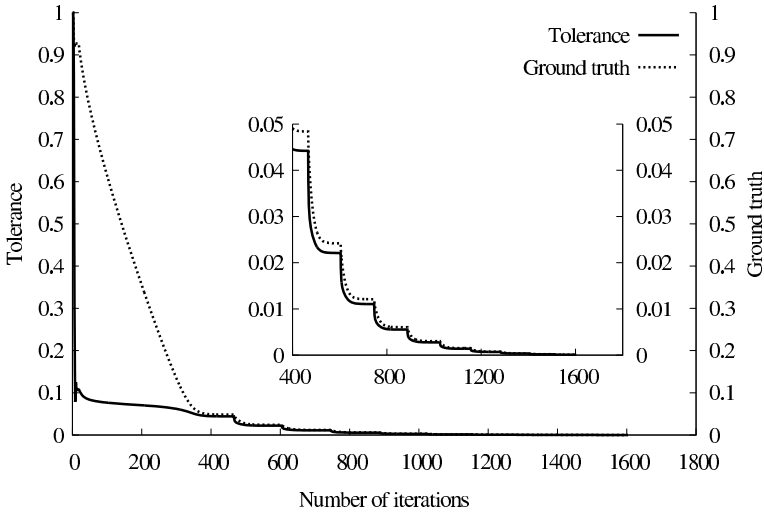


Fig. 1. Errors with respect to time for a representative example of orthogonalized Gaussian matrix of size 2048×16384

of memory, and a NVIDIA GEFORCE GTX 275 with 240 cores and 896MB of memory. The values of the parameters we have used for our experiments are the following. We set $m = n/8$ (recall that $m \ll n$). The number of non-zero values in the original sparse signal is $k = m/10$. Concerning the parameter for the stopping criteria, we set $e_{tol} = 10^{-5}$ and $e_{consec} = 10^{-3}$. For experiments where the signal size varies, we report the average result of 10 different instances for each signal size.

Figure 1 represents the variation of errors with respect to time. We chose a representative example of orthogonalized Gaussian matrix to illustrate the different steps of the optimization process. We consider two error criteria : e_{tol} and the relative error of the reconstructed signal $u^{(k+1)}$ to the ground truth defined as $e_{relative} = \frac{\|u^{(k+1)} - u^*\|_2}{\|u^*\|_2}$.

The bitonic search can be seen at the beginning of the process. Once an appropriate initial μ is found, the continuation process is launched. On the curve, each step of the continuation corresponds to a decrease of both error criteria. The ground truth error decreases almost linearly until 400 iterations, from then the decrease is far slower. On the contrary, the relative error does not show this first fast decrease process. We learn from this curve that our method could be stopped earlier using tighter parameters. This is very common for this kind of optimization methods. However, this increase the difficulty of the comparison with other methods because of the multiple biased involved. Indeed, each method could have its own set of peculiar parameters. In particular, our method has the advantage to not rely on a dt parameter representing the optimization process step, unlike linearization methods.

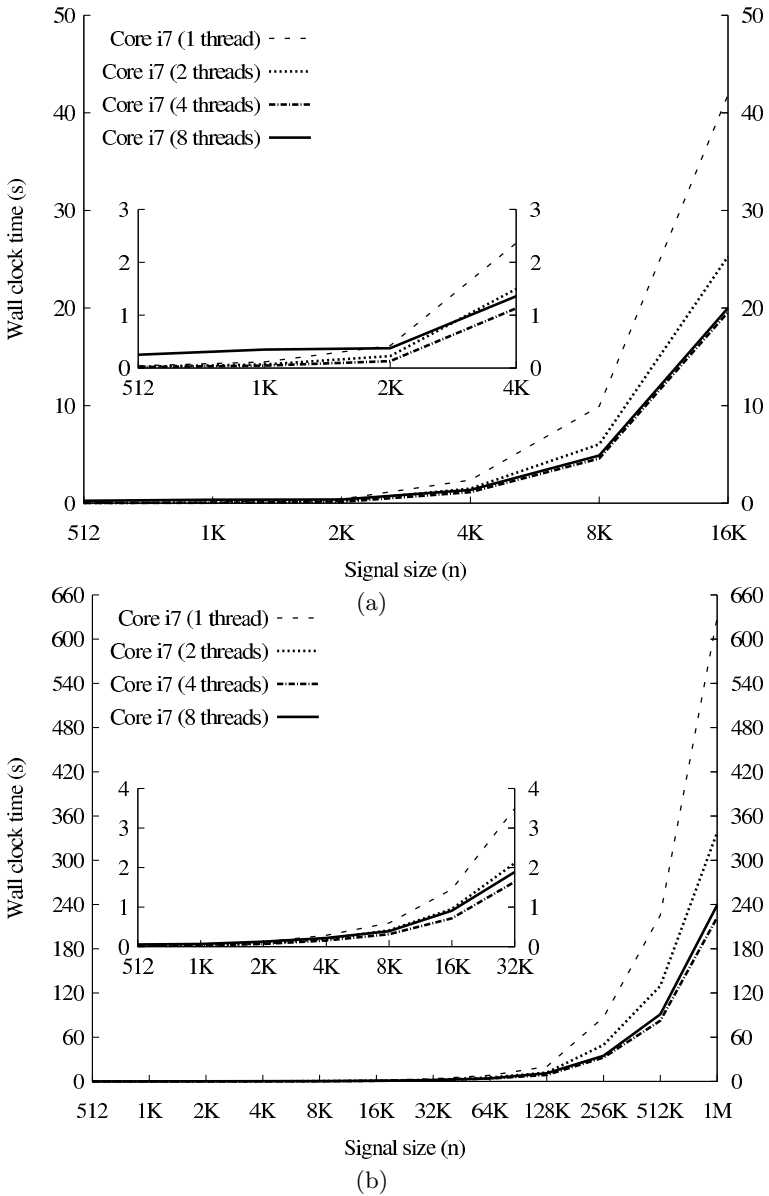


Fig. 2. Results for orthogonalized Gaussian matrices (a) and partial DCTs (b) on a Intel Core i7 920

Figure 2(a) (resp. 2(b)) presents results for orthogonalized Gaussian matrices (resp. partial DCTs) with an Intel Core i7 CPU for various number of threads and matrix sizes. Increasing the number of threads increases performance. Note

that the bigger the problem is, the better the scaling is. The CPU used here has four physical cores but each one is seen as two logical cores. This feature can yield better performance thanks to a better use of computational units. Nevertheless, for this particular application, using eight threads does not increase performance and a high overhead is observed for small FFT sizes.

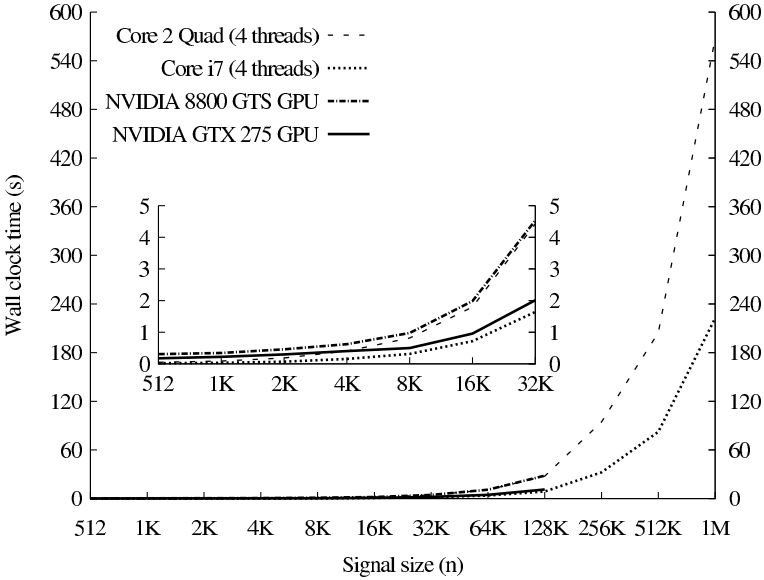


Fig. 3. Results for partial DCTs on various platforms

Figure 3 is a comparison between the GPU and CPU implementations. GPUs are limited by the size of the matrices they can handle. However, our GPU implementation is almost as fast as our optimized multi-core CPU implementation. More precisely, the 8800 GTS is head to head with the Core 2 Quad while the GTX 275 is head to head with the Core i7. Quad-core CPUs are always slightly faster than GPUs.

5 Conclusion

We have proposed a simple and efficient algorithm for solving the l^1 compressive sensing problem on parallel many-core architectures. This algorithm has been especially designed to take benefit of current parallel many-core architectures and achieves noticeable speedups. To validate our approach, we proposed implementations on various current high-end platforms, such as vectorized multi-core CPU, GPU and Cell. The results are promising and allow to hope very fast implementations on new architectures.

References

1. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory* 52, 489–509 (2006)
2. Tropp, J.: Just relax: Convex programming methods for identifying sparse signals. *IEEE Trans. on Information Theory* 51, 1030–1051 (2006)
3. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Trans. on PAMI* 31, 210–227 (2009)
4. Cevher, V., Sankaranarayanan, A., Duarte, M.F., Reddy, D., Baraniuk, R.G., Chellappa, R.: Compressive sensing for background subtraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 155–168. Springer, Heidelberg (2008)
5. Lustig, M., Donoho, D., Pauly, J.M.: Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine* 58, 1182–1195 (2007)
6. Combettes, P., Pesquet, J.C.: Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM J. on Opt.* 18, 1351–1376 (2007)
7. Figueiredo, M., Nowak, R., Wright, S.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Sig. Proc.* 1 (2007)
8. Hiriart-Urruty, J.B., Lemaréchal, C.: *Convex Analysis and Minimization Algorithms*. Springer, Heidelberg (1996)
9. Borghi, A., Darbon, J., Peyronnet, S., Chan, T., Osher, S.: A simple compressive sensing algorithm for parallel many-core architectures. Technical report, UCLA (2008)

An Incremental PCA-HOG Descriptor for Robust Visual Hand Tracking

Hanxuan Yang^{1,2}, Zhan Song^{1,2}, and Runen Chen³

¹ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

² The Chinese University of Hong Kong, Hong Kong, China

³ Dept. of Electronic & Information Engineering, South China Agricultural University, Guangzhou, China

{hx.yang, zhan.song}@sub.siat.ac.cn, crun@scau.edu.cn

Abstract. Hand tracking in complicate scenarios is a crucial step to any hand gesture recognition systems. In this paper, we present a novel hand tracking algorithm with adaptive hand appearance modeling. In the algorithm, the hand image is first transformed to the grids of Histograms of Oriented Gradients. And then an incremental Principle Component Analysis is implemented. We name this operator an incremental PCA-HOG (IPHOG) descriptor. The exploitation of this descriptor helps the tracker dealing with vast changing of hand appearances as well as clutter background. Moreover, Particle filter method with certain improvements is also introduced to establish a tracking framework. The experimental results are conducted on an indoor scene with clutter and dynamic background. And the results are also compared with some traditional tracking algorithms to show its strong robustness and higher tracking accuracy.

1 Introduction

Vision based Human Computer Interaction (HCI) systems have got a lot of applications in interactive gaming control, intelligent electronics and robotics etc. As an important HCI means, hand gesture recognition has attracted increasing attention for its convenience and accordance with the user's natural operating manners. Hand tracking is a crucial step in all hand gesture recognition systems and has been a classical research topic in computer vision domain. However, in real applications, current tracking algorithms are usually fail to deal with clutter and dynamic scenarios subject to the change of hand appearance and shape, dynamic and clutter background etc.

According to the principle of present hand tracking algorithms, they can be generally classified into two categories: model-based methods and appearance-based methods. In the model-based methods, state of a possible hand is estimated by projecting the pre-stored 3D hand model to the image plane and then the projection is compared with the image features. In [1], the idea of belief propagation on a graph consisting of local hand parts has been introduced. In [2], a concept of attractor is introduced to boost the tracking performance. The attractors are defined to some reference states and serve as prior knowledge to guide the tracking in a high-dimensional space. In [3], a hierarchical model-based approach is proposed to find and track hands. The features of hand shape and orientation are used to track the hands with articulated motion. The model-based

methods have the ability to cope with occlusion and to obtain detailed and accurate motion data. However, a common problem with model-based methods is the “curse of dimensionality” due to the large number of degrees of freedom of the articulated hand motion. Therefore, model-based methods often suffer from high computational cost and impractical to real applications. The appearance-based methods try to extract the hand states by analyzing the image features from the hand appearance directly [4-7]. Skin color is the most often used appearance clue. In [4], a Gaussian skin color model with histogram approach in HSV color space is introduced. It was showed to be efficient for skin color detection. However, the direct use of skin colors makes the algorithm sensitive to illumination variations. In [5], a boosting mechanism to deal with hand tracking problem based on Harr-like features is proposed. The method needs to construct a cascaded boosted classifier off-line. Some other methods are attempted to improve the tracking robustness by combing multiple visual clues. In [6], a model-based approach in combination with the edges, optical flow and shading information for tracking is proposed. Assuming that human hand can be represented efficiently by its shape feature using the spatial distribution of edges, the oriented edge energy histogram based features have been used successfully to represent hand shape [7]. Moreover, the recent great success of the Histograms of Oriented Gradients (HOG) descriptor [8] also confirmed the basic idea that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions. In comparison, the appearance-based tracking methods are more computation efficient than model-based methods. However, these methods usually fail in the presence of significant variations of the object’s appearance or illumination. The reason is that many algorithms employ fixed appearance models. Such models are trained using only appearance data available before tracking begins, which in practice limits the range of appearances that can be modeled, and ignores the large volume of information that becomes available during tracking. Recently, incremental or online learning methods have been proposed to construct adaptive appearance models. In [9], an on-line boosting framework that adaptively selects features to discriminate the object from the background has been proposed. In [10] an incremental Principle Component Analysis (IPCA) method combined with a Particle Filter (PF) inference framework [11] is proposed for visual tracking.

Inspired by the works in [8] and [10], an Incremental PCA-HOG (IPHOG) descriptor is proposed in this paper. The descriptor can be computed by transforming the hand image to the grids of HOG descriptor and then combine it with the incremental PCA algorithm. The Particle Filter method [11] with some specific improvements is introduced to establish our tracking framework. Experiments on dynamic scenes with complicate background are used to demonstrate its robustness and performance.

The paper is organized as follows. Section 2 presents our approach for constructing the incremental PCA-HOG descriptor. The detail of the proposed tracking algorithm is described in Section 3. Experimental results are given in Section 4. Conclusion and future work can be found in Section 5.

2 Incremental PCA-HOG Descriptor

To describe the hand mathematically, geometrical shape like the contours is the most remarkable feature. Inspired by the concept of HOG descriptor, a novel descriptor

named incremental PCA-HOG is proposed to represent the hand by its local shape, which is captured by the distribution over edge orientations within a region. The similar work can be found in [12-13], where an offline PCA-HOG descriptor has been proposed for human detection and recognition. However, such an offline PCA-HOG descriptor needs to collect all the training samples beforehand and is poor at dealing with visual tracking with various variations. On the other hand, our incremental PCA-HOG descriptor can adapt to these variations online during tracking.

2.1 Review of HOG Descriptor

Let $\theta(x, y)$ and $m(x, y)$ be the orientation and magnitude of the intensity gradient at image point (x, y) . The image gradients can be computed via a finite difference mask $[-1 \ 0 \ 1]$ and its transpose. The gradient orientation at each pixel is discretized into one of p values using a contrast insensitive definition as:

$$B(x, y) = \text{round}\left(\frac{p \cdot \theta(x, y)}{\pi}\right) \bmod p \quad (1)$$

Let $b \in \{0, \dots, p-1\}$ range over orientation bins. The feature vector at (x, y) is:

$$F(x, y)_b = \begin{cases} m(x, y) & \text{if } b = B(x, y) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Let F be a pixel-level feature map for a $w \times h$ image and $k > 0$ be a parameter indicates the side length of a square image region. A dense grid of rectangular ‘‘cells’’ [8] is defined and pixel-level features are aggregated to obtain a cell-based feature map C , with feature vectors $C(i, j)$ for $0 \leq i \leq [(w-1)/k]$ and $0 \leq j \leq [(h-1)/k]$. This aggregation can reduce the size of a feature map. After a bilinear interpolation to aggregate features, each feature can be normalized. The resulting feature vector is the HOG descriptor of the image region. Normally the parameters of HOG descriptor are set to $p=9$ and $k=8$, the size of the ‘‘cell’’ is 2×2 . This leads to a 36-dimensional feature vector.

2.2 Incremental PCA-HOG Descriptor

Traditional appearance-based tracking algorithms often fail in the presence of significant variations of the object’s appearance or illuminations due to the exploitation of fixed appearance models of the target. Recently, incremental subspace learning [10] has provided an effective way to deal with the changes of target based on the assumption of constant subspace. However, such an assumption is not always the case in practice, hence leading to a tracking failure. Inspired by the adaptation characteristic of incremental subspace learning [10] and the high robustness of HOG descriptor [8], we propose a new incremental PCA-HOG descriptor for visual hand tracking.

Assume $I_i \in \mathbb{R}^{w \times h}$, $i = 1, \dots, T$ denote an image sequence of width w and height h . The incremental PCA-HOG descriptor can be computed by the following procedures:

- a) Compute the corresponding HOG descriptor H_i , $i = 1, \dots, t$ for the first t frames.
- b) Assume U denote the top q principal components learned from the HOG descriptor of the first t frames during previous tracking. The value of t in our experiment is set to 5 empirically.
- c) Project the HOG descriptor H_{t+1} to the linear subspace spanned by the principal components U as:

$$Y = U^T (H_{t+1} - \bar{H}). \quad (3)$$

where \bar{H} indicates the mean HOG descriptor of the former frames and Y is the computed PCA-HOG descriptor.

- d) Update the eigenbasis online following the incremental PCA algorithm [10]. We can have a new eigenspace spanned by the updated principle components U' . Repeating process c) and d), the PCA-HOG descriptor Y can be updated online during tracking.

To obtain an optimal value of q , a number of q values are tested and the results are compared with the original 36-dimensional HOG features. The experiment shows that the eigenspace spanned by the top 12 eigenvectors captures essentially all the information in a HOG feature. Fig.1 (a) gives an example of input image. Fig.1 (c) shows the corresponding 36-dimensional HOG descriptor with 8 orientations. Fig.1 (d) shows the reconstructed HOG descriptor with $q=12$. We can observe that there is no significant difference between the original HOG descriptor and the reconstruction.

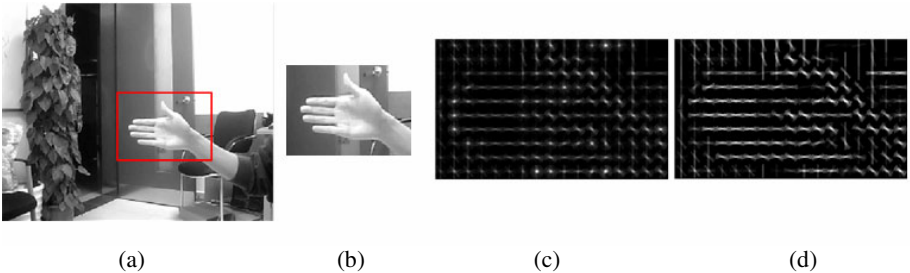


Fig. 1. Optimal selection of q value in the proposed HOG descriptor. (a) The input image. (b) The cropped hand image for descriptor computation. (c) The 36-dimensional HOG descriptor with 8 orientations. (d) The reconstructed HOG descriptor spanned by 12 principal components.

The incremental PCA-HOG descriptor has two main advantages. First, the representation is based on the distribution of gradients, which is more robust than representation based on direct image intensities, like incremental PCA method [10]. Second, it

can update adaptively during the tracking, which is more flexible than traditional HOG descriptor [8].

3 Incorporate IPCA-HOG Descriptor with Particle Filter

Particle filter has provided a flexible and effective tracking framework. Therefore, we embed the incremental PCA-HOG descriptor into particle filter framework to form a robust tracking algorithm. Suppose that $Z_t = \{z_1, \dots, z_t\}$ denote the observations at the time t . The corresponding states are $X_t = \{x_1, \dots, x_t\}$. Therefore, the tracking problem can be formulated as:

$$P(x_t | Z_t) = \sum_{n=1}^N w_t^n \delta(x_t - x_t^n). \quad (4)$$

$$w_t^n \propto \frac{s(z_t | x_t^n) p(x_t^n | x_{t-1}^n)}{q(x_t | x_{t-1}^n, Z_t)}. \quad (5)$$

where $\delta()$ refers to the Dirac function, $\{x_t^n, w_t^n\}_{n=1}^N$ denote the particles with associated weight. $s()$ and $p()$ denote the observation and dynamical model respectively. $q()$ indicates the proposal distribution.

The object state in any frame can be represented by an affine image warp as:

$$W(\Phi, P) = \begin{pmatrix} p_1 & p_3 & p_5 \\ p_2 & p_4 & p_6 \end{pmatrix} \begin{pmatrix} d_x \\ d_y \\ 1 \end{pmatrix}. \quad (6)$$

where $P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$ are the affine parameters and $s = (d_x, d_y)^T$ is a column vector containing the pixel coordinates. The Lucas-Kanade method [14] is used to provide a heuristic prediction of the particles generation process. The key idea is to minimize the sum of squared errors between two images and assume that the current estimate P is known and then iteratively solves for the parameters increments ΔP . Each parameter in x_t is modeled independently by the Gaussian distribution around its estimated state \tilde{x}_t as:

$$p(x_t | x_{t-1}) = G(x_t | \tilde{x}_t, \Sigma). \quad (7)$$

where $G()$ indicates the Gaussian distribution, $\tilde{x}_t = x_{t-1} + \Delta P$. Σ is a diagonal covariance matrix whose elements are the corresponding variances of affine transformation.

Selection of proposal distribution is a crucial step in particle filter. And the most popular choice is the dynamical transmission function, but this function does not

include the latest observations. It is known that this method is usually inefficient when the likelihood is situated in the prior's tail or when the likelihood is highly peaked. In this paper, we adopt the method in [15], which proposed an improved unscented particle filter method to compute the proposal distribution. The key idea is to introduce the SVD approach to compute the sigma points within the unscented transformation rather than the traditional Cholesky factorization method. Then following the steps in the unscented Particle Filter algorithm (UPF) [16], we can obtain the proposal distribution as:

$$q(x_t | x_{t-1}^n, z_{1:t}) = G(\bar{x}_t^{-(n)}, \hat{P}_t^{(n)}). \tag{8}$$

where \bar{x}_t^- , \hat{P}_t are the mean and covariance parameters of the state random variable $x_t^a = [x_t^T, u_t^T, v_t^T]^T$ respectively, which is the concatenation of the original state x_t and the process noise variables u_t, v_t . u_t, v_t is independently drawn from zero-mean Gaussian distribution. Comparing to the traditional UPF method, this approach is more numerically stable.

The observation model is a basic issue to be considered as the particle filter is adopted for tracking. Given the learning subspace U and the new observation z_t , the observation model is based on the reconstruction error between the observation and the reconstruction of $UU^T z_t$ [17]. Hence, the computation of observation model can be naturally expressed as:

$$p(z_t | x_t) = \exp(-\|z_t - UU^T z_t\|^2) \tag{9}$$

where z_t is the HOG descriptor of the input image. $UU^T z_t$ is the reconstruction of our incremental PCA-HOG descriptor.

4 Experimental Results

To evaluate performance of the proposed tracking algorithm, an image sequence which contain a moving hand and dynamic and complicate background were captured indoor for experiments. The frame size is 320×240 pixels. In our experiments, the object is initialized manually and affine transformation is considered only. Specifically, the motion is characterized by $P = (p_1, p_2, p_3, p_4, p_5, p_6)$ where $\{p_1, p_2\}$ denote the 2-D translation parameters and $\{p_3, p_4, p_5, p_6\}$ are deformation parameters. Each candidate image is rectified to a 32×32 patch. All of the experiments are carried out on an AMD Athlon X2 desktop with 3.0G CPU and 2G RAM. The program is implemented under MATLAB R2008a and run at 2 frames per second.

We first test the proposed incremental PCA-HOG descriptor for hand tracking in the environment where illumination and background are changing shown in Fig. 2. The parameters in our experiment are set as: $\Sigma = [5,5,0.01,0.02,0.002,0.001]$, $N = 200$, where N means the number of particle filters, Σ is a diagonal covariance

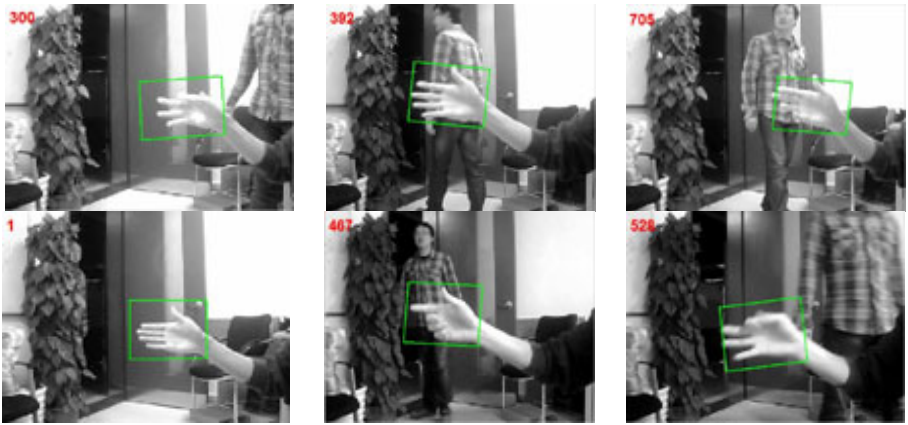


Fig. 2. A tracking scenario with dynamic background: the hand poses (#300, #392, #705) and illumination (#1, #467, #528) changes have slightly effect to the tracking result

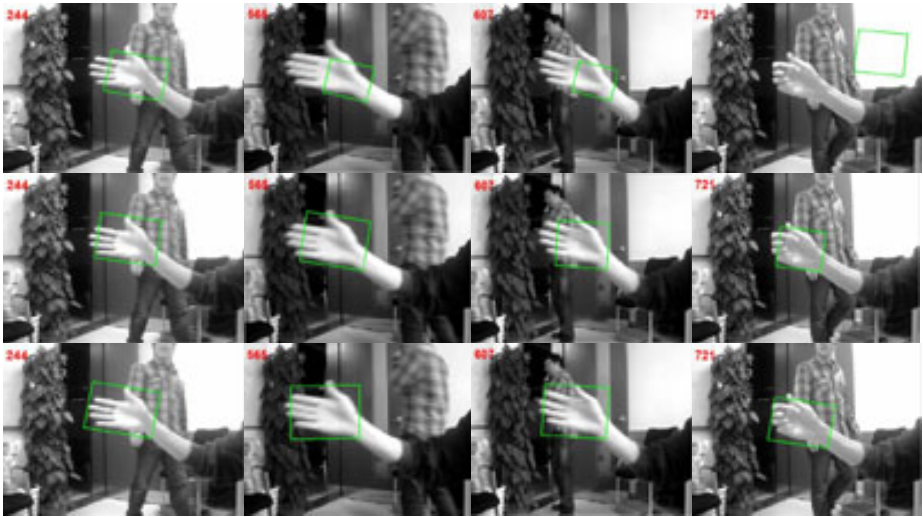


Fig. 3. Tracking result comparison between three methods: IVT (1st row), geometric particle filter on affine group (2nd row), and the proposed algorithm (3rd row)

matrix whose elements are the corresponding variances of affine transformation. The results show that the changing of hand pose, illumination and background has slightly effect to the tracking result.

For comparison, two state-of-the-art methods, the IVT [10] and geometric particle filter on affine group [18] are experimented on the test video which is sampled alternately to form a rapid motion sequence. As shown in Fig. 3, the first row is IVT method, the second row is geometric particle filter method and the last row is the proposed method. From the results we can see that the proposed method can outperform in a large scale in both robustness and tracking precision. The reasons are twofold. First, the

incremental PCA-HOG descriptor is based on the distribution of gradients, which is more robust than the IVT method based on intensities. Second, particles of IVT are simply sampled from dynamical transmission. Therefore, particles fail to cover the latest observations. While in our algorithm, the proposal distribution can cover the latest observations. In addition, the improvement of dynamical model also contributes to the final high robustness. Comparing to IVT, the geometric particle filter has introduced an improved dynamical model and sampling method based on the affine group. Such changes have made some improvements in the tracking outcome. However, its observation model is the same as that of IVT. Hence, it is still hard to deal with tracking problem with abrupt pose and illumination changes.

5 Conclusion and Future Work

In this paper, we have presented an incremental PCA-HOG descriptor for visual hand tracking in complicate scenario. Such a descriptor combines both the adaptation merit of incremental subspace learning as well as the high robustness merit of HOG descriptor. In addition, we adopt the particle filter method with some improvements as our tracking framework. Experimental results show that, the proposed tracking algorithm can work robustly and accurately in scenarios where the background, hand pose, and illumination are changing. The method is also compared with two other state-of-the-art algorithms to show its advantage in robustness and tracking precision. Future work can introduce more robust nonlinear incremental subspace learning method to further improve the tracking performance.

Acknowledgments

The work described in this article was partially supported by NSFC (Project no. 61002040) and Knowledge Innovation Program of the Chinese Academy of Sciences (Grant no. KG CX2-YW-156).

References

1. Sudderth, E.B., Mandel, M.I., Freeman, W.T., Willsky, A.S.: Visual Hand Tracking Using Nonparametric Belief Propagation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2004)
2. Chang, W.Y., Chen, C.S., Hung, Y.P.: Appearance-guided Particle Filtering for Articulated Hand Tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2005)
3. Stenger, B., Thayananthan, A., Torr, P., Cipolla, R.: Model-based Hand Tracking Using a Hierarchical Bayesian Filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 1372–1384 (2006)
4. Schmutge, S.J., Jayaram, S., Shin, M.C., Tsap, V.: Objective Evaluation of Approaches of Skin Detection Using ROC Analysis. *Computer Vision and Image Understanding* 105, 41–51 (2007)

5. Barreto, J., Menezes, P., Dias, J.: Human-robot Interaction Based on Haar-like Features and Eigenfaces. In: Proceedings of the 2004 IEEE Conference on Robotics and Automation, pp. 1888–1893 (2004)
6. Shan, L., Metaxas, D., Samaras, D., Oliensis, J.: Using Multiple Cues for Hand Tracking and Model Refinement. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 443–450 (2003)
7. Freeman, W.T., Roth, M.: Orientation Histograms for Hand Gesture Recognition. In: International Workshop on Automatic Face and Gesture Recognition, pp. 296–301 (1995)
8. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
9. Grabner, H., Bischof, H.: On-line Boosting and Vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 260–267 (2006)
10. Ross, D.A., Lim, J., Lin, R., Yang, M.: Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision* 77, 125–141 (2008)
11. Isard, M., Blake, A.: ICONDENSATION: Unifying Low-level Tracking in a Stochastic Framework. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 893–908. Springer, Heidelberg (1998)
12. Lu, W.L., Little, J.J.: Simultaneous Tracking and Action Recognition Using PCA-HOG Descriptor. In: The 3rd Canadian Conference on Computer and Robot Vision, pp. 6–13 (2006)
13. Kobayashi, T., Hidaka, A., Kurit, T.: Selection of Histograms of Oriented Gradients Features for Pedestrian Detection. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 598–607. Springer, Heidelberg (2008)
14. Baker, S., Matthews, I.: Lucas–Kanade: 20 Years on: A Unifying Framework. *International Journal of Computer Vision* 56(1), 221–255 (2004)
15. Yang, H.X., Song, Z., Chen, R.E.: An Improved Unscented Particle Filter for Visual Hand Tracking. In: Proceedings of 3rd International Congress on Image and Signal Processing (2010)
16. Merwe, R., Doucet, A., Freitas, N., Wan, E.: The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department (2000)
17. Black, M.J., Jepson, A.D.: Eigentracking: Robust Matching and Tracking of Articulated Objects Using View Based Representation. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 329–342. Springer, Heidelberg
18. Kwon, J., Lee, K., Park, F.: Visual Tracking via Geometric Particle Filtering on Affine Group with Optimal Importance Functions. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)

Probabilistic Learning of Visual Object Composition from Attended Segments

Masayasu Atsumi

Dept. of Information Systems Science, Faculty of Engineering, Soka University
1-236 Tangi, Hachioji, Tokyo 192-8577, Japan
matsumi@t.soka.ac.jp

Abstract. This paper proposes a model of probabilistic learning of object categories in conjunction with early visual processes of attention, segmentation and perceptual organization. This model consists of the following three sub-models: (1) a model of attention-mediated perceptual organization of segments, (2) a model of local feature representation of segments by using a bag of features, and (3) a model of learning object composition of categories based on intra-categorical probabilistic latent component analysis with variable number of classes and inter-categorical typicality analysis. Through experiments by using images of plural categories in an image database, it is shown that the model learns a probabilistic structure of intra-categorical composition of objects and context and inter-categorical difference.

1 Introduction

People can identify object categories in crowded scenes. However, they have not necessarily learned those categories one by one in a supervised manner but learned them in a semi-supervised or unsupervised manner. In this paper, under the presumption that object composition of categories is statistically acquired by semi-supervised learning through attention, we propose a model of probabilistic learning of object categories in conjunction with early visual processes of attention, segmentation and perceptual organization. This model consists of the following three sub-models: the first is a model of attention-mediated perceptual organization of segments [1]; the second is a model of local feature representation of segments by using a bag of features (BoF) [2]; and the third is a model of learning object composition of categories based on an extended probabilistic latent component analysis (PLCA) [3,4]. In attention-mediated perceptual organization of segments, concurrent figure-ground segmentation is performed on dynamically-formed Markov random fields (MRFs) around salient preattentive points and co-occurring segments of objects and their context are grouped in the neighborhood of selective attended segments. In order to represent local feature of segments, the SIFT features [5] of salient points in co-occurring segments are clustered by the K-tree method [6] to generate a set of key features (a code book) and a BoF of each segment is calculated by using this code book. In probabilistic learning of categorical object composition, multi-class classifiers

are learned based on intra-categorical PLCA with variable number of classes and inter-categorical typicality analysis. These processes are performed on a collection of categorized image sets which contain categorical objects in crowded scenes. There have been proposed several methods which incorporate attention into object learning [7] and context into object categorization [8,9,10]. The main difference of our method from those existing ones is that (i) it uses attended co-occurring segments which are perceptually organized for learning and (ii) it learns a probabilistic structure of typical and non-typical objects and their context in each category. This model makes learning of intra-categorical object composition and inter-categorical difference possible from cues that categorical objects exist somewhere in crowded scenes.

This paper is organized as follows. Section 2 presents attention-mediated perceptual organization and BoF representation of segments. Section 3 describes probabilistic learning of object composition of categories. Experimental results are shown in section 4 and we conclude our work in section 5.

2 Attended Co-occurring Segments

2.1 Attention-Mediated Perceptual Organization

The model of attention-mediated perceptual organization [1] consists of a saliency map [11,12] for preattention, a collection of dynamically-formed Markov random fields for figure-ground segmentation, a visual working memory for maintaining segments and perceptually organizing them around selective attention, and an attention system on a saliency map and a visual working memory. Fig. 1 depicts the organization and the computational steps of the model.

As features of an image, brightness, hue and their contrast are obtained on a Gaussian resolution pyramid of the image. A saliency map is obtained by calculating saliency from brightness contrast and hue contrast on each level of the Gaussian resolution pyramid and integrating the multi-level saliency into one

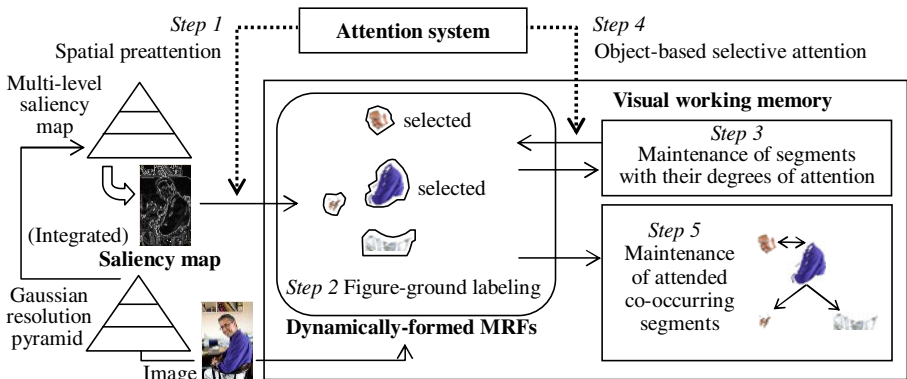


Fig. 1. Attention-mediated Perceptual Organization

map. The following procedure shows an outline of attention-mediated perceptual organization. The detail of the saliency map is described in [12] and the detail of segmentation and perceptual organization is described in [1].

- Step 1.** Preattentive points are stochastically selected from a saliency map according to their degrees of saliency.
- Step 2.** Figure-ground labeling is iterated on dynamically-formed 2-dimensional MRFs of brightness and hue around preattentive points until figure segments converge or a specified iteration is reached. Here the EM procedure with the mean field approximation [13] is used for MRF labeling. If plural figure segments satisfy a merge condition, they are merged into one segment.
- Step 3.** Attention degrees of segments are calculated from their saliency, closedness and attention bias. Saliency of a segment is defined by both the degree to which a surface of the segment stands out against its surrounding region and the degree to which a spot in the segment stands out by itself. Closedness of a segment is judged whether it is closed in an image, that is, whether or not it spans beyond the border of an image. Attention bias represents a priori or experientially acquired attentional tendency to a region with a particular feature such as a face-like region.
- Step 4.** From these segments, the specified number of segments whose degrees of attention are larger than others are selected as selective attended segments.
- Step 5.** Each selective attended segment and its neighboring segments are grouped as a co-occurring segment. If two sets of co-occurring segments overlap, they are combined into one co-occurring segment. This makes it possible to group part segments of an object or group salient contextual segments with an object.

2.2 BoF Representation of Segments

A segment or a co-occurring segment is represented by a BoF histogram [2] of local feature of its salient points. In order to calculate the BoF histogram, first of all, any points in a segment whose saliency are above a given threshold are extracted as salient points at each level of a multi-level saliency map. As a local feature, a 128-dimensional SIFT feature [5] is calculated for each salient point at its resolution level. Then, all the SIFT features of all segments are clustered by the K-tree method [6] to obtain a set of key features as a code book. Finally, a BoF histogram of each segment or co-occurring segment is calculated by using this code book. As a result, feature of a segment or a co-occurring segment is represented by a BoF histogram.

3 Probabilistic Learning of Object Categories

The problem to be addressed is statistically learning object composition of each category from a cue that the categorical objects exist somewhere in crowded scenes. Concretely, given a collection of categorized image sets which contain

the categorical objects in crowded scenes, it is required to learn a probabilistic structure of categorical objects and their context from attended co-occurring segments extracted in images of each category. In the proposed method, intra-categorical PLCA with variable number of classes is firstly applied to each category and learns a multi-class classifier from BoF histograms of segments in the category. Then each class in each category is judged how typical in the category through inter-categorical typicality analysis. We call this learning method a probabilistic latent component forest (PLCF).

In this paper, let s_{c,i_j} be a segment j extracted from an image i of a category c , S_c be a set of segments extracted from any images of a category c , and N_{c_s} be the number of segments in S_c . Let f_n be a n -th element of key features F , N_f be the number of key features and $\{H(s_{c,i_j}, f_n)\}_{f_n \in F}$ be a BoF histogram of a segment s_{c,i_j} . Let $q_{c,r}$ be a latent class of a category c , Q_c be a set of latent classes of a category c , N_{c_q} be the number of classes of a category c , C be a set of categories and N_c be the number of categories.

3.1 Learning Multi-class Classifiers of Object Categories

The problem of learning a multi-class classifier for segments in a category c is estimating probabilities $p(s_{c,i_j}, f_n) = \sum_r p(q_{c,r})p(s_{c,i_j}|q_{c,r})p(f_n|q_{c,r})$ namely $\{p(q_{c,r})|q_{c,r} \in Q_c\}C\{p(s_{c,i_j}|q_{c,r})|s_{c,i_j} \in S_c, q_{c,r} \in Q_c\}, \{p(f_n|q_{c,r})|f_n \in F, q_{c,r} \in Q_c\}$ and the number of latent classes N_{c_q} that maximize the following log-likelihood

$$L_c = \sum_{i_j} \sum_n H(s_{c,i_j}, f_n) \log p(s_{c,i_j}, f_n). \tag{1}$$

When the number of latent classes is given, these probabilities are estimated by the EM algorithm in which the following E-step and M-Step are iterated until convergence

[E-step]

$$p(q_{c,r}|s_{c,i_j}, f_n) = \frac{[p(q_{c,r})p(s_{c,i_j}|q_{c,r})p(f_n|q_{c,r})]^\beta}{\sum_{q_{c,r'}} [p(q_{c,r'})p(s_{c,i_j}|q_{c,r'})p(f_n|q_{c,r'})]^\beta} \tag{2}$$

[M-step]

$$p(f_n|q_{c,r}) = \frac{\sum_{s_{c,i_j}} H(s_{c,i_j}, f_n)p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{f_n'} \sum_{s_{c,i_j}} H(s_{c,i_j}, f_n')p(q_{c,r}|s_{c,i_j}, f_n')} \tag{3}$$

$$p(s_{c,i_j}|q_{c,r}) = \frac{\sum_{f_n} H(s_{c,i_j}, f_n)p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{s_{c,i_j'}} \sum_{f_n} H(s_{c,i_j'}, f_n)p(q_{c,r}|s_{c,i_j'}, f_n)} \tag{4}$$

$$p(q_{c,r}) = \frac{\sum_{s_{c,i_j}} \sum_{f_n} H(s_{c,i_j}, f_n)p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{s_{c,i_j}} \sum_{f_n} H(s_{c,i_j}, f_n)} \tag{5}$$

where β is a temperature coefficient.

The number of latent classes is determined through an EM iterative process with subsequent class division. The process starts with one or a few classes, pauses at every certain number of EM iterations less than an upper limit and calculates the following index, which is called the degree of scatter,

$$\delta_{q_{c,r}} = \frac{\sum_{s_{c,i_j}} (\sum_{f_n} |p(f_n|q_{c,r}) - D(s_{c,i_j}, f_n)|) \times p(s_{c,i_j}|q_{c,r})}{N_f \times N_{c_s}} \quad (6)$$

where

$$D(s_{c,i_j}, f_n) = \frac{H(s_{c,i_j}, f_n)}{\sum_{f_{n'}} H(s_{c,i_j}, f_{n'})} \quad (7)$$

for $\forall q_{c,r} \in Q_c$. Then a class whose degree of scatter takes a maximum value among all classes is divided into two classes. This iterative process is continued until $\delta_{q_{c,r}}$ -values for all classes become less than a certain threshold. The latent class is divided into two classes as follows. Let q_{c,r_0} be a source class to be divided and let q_{c,r_1} and q_{c,r_2} be target classes after division. Then, for a segment $s_{c,i_j^*} = \arg \max_{i_j} \{p(s_{c,i_j}|q_{c,r_0})\}$ which has the maximum conditional probability and its BoF histogram $H(s_{c,i_j^*}, f_n) = [h_{c,i_j^*}(1), \dots, h_{c,i_j^*}(N_f)]$, one class q_{c,r_1} is set by specifying its conditional probability of key features, conditional probabilities of segments and a class probability as

$$p(f_n|q_{c,r_1}) = \frac{h_{c,i_j^*}(n) + \alpha}{\sum_{n'} (h_{c,i_j^*}(n') + \alpha)} \quad \forall f_n \in F \quad (8)$$

$$p(s_{c,i_j}|q_{c,r_1}) = p(s_{c,i_j}|q_{c,r_0}) \quad \forall i_j \in S_c, \quad p(q_{c,r_1}) = \frac{p(q_{c,r_0})}{2}$$

respectively where α is a positive correction coefficient. Another class q_{c,r_2} is set by specifying its conditional probability of key features $\{p(f_n|q_{c,r_2})|f_n \in F\}$ at random, conditional probabilities of segments $\{p(s_{c,i_j}|q_{c,r_2})|i_j \in S_c\}$ as 0 for s_{c,i_j^*} and equal for other segments, and a class probability as $p(q_{c,r_2}) = \frac{p(q_{c,r_0})}{2}$. As a result of subsequent class division, latent classes form a binary tree, which we call a probabilistic latent component tree (PLCT).

The temperature coefficient β is set 1.0 until the number of classes is fixed and after that it is gradually decreased according to a given schedule of the tempered EM until convergence.

3.2 Inter-categorical Typicality Analysis

Inter-categorical typicality analysis evaluates each latent class of each category whether or not it is typical in the category and calculates a conditional probability of key features for the category by synthesizing those probabilities of its typical classes. The typicality of a class is judged based on whether it appears in its category with high frequency but does not appear in other categories only with low frequency. Typical classes consist of classes of object segments and co-occurring contextual segments which distinguish a category and a synthesized

conditional probability of key features encodes characteristics of the category. Here, in general, co-occurring contextual segments are objects of other categories or background. By the way, exceptional object segments, as well as typical object segments, are also encoded by conditional probabilities of key features of some non-typical classes.

For the inter-categorical typicality analysis, let the distance between classes $q_{c_1, r_1} \in Q_{c_1}$ and $q_{c_2, r_2} \in Q_{c_2}$ of any different categories c_1 and c_2 be

$$L_1(q_{c_1, r_1}, q_{c_2, r_2}) = \frac{\sum_{f_n} |p(f_n | q_{c_1, r_1}) - p(f_n | q_{c_2, r_2})|}{2} \quad (9)$$

by using their conditional probabilities of key features. Then the distance of a class $q_{c, r} \in Q_c$ of a category c from classes of other categories is defined as

$$d(c, q_{c, r}) = \frac{\sum_{c' \in C - \{c\}} \sum_{q_{c', r'} \in Q_{c'}} (L_1(q_{c, r}, q_{c', r'}) \times p(q_{c', r'}))}{N_c - 1}. \quad (10)$$

Now, for the mean distance $\bar{d}(c) = \frac{\sum_{q_{c, r} \in Q_c} d(c, q_{c, r})}{N_{c_q}}$ of all classes of a category c , the deviation of a distance $d(c, q_{c, r})$ is given by $\Delta(c, q_{c, r}) = d(c, q_{c, r}) - \bar{d}(c)$. Then the typicality index of a class $q_{c, r}$ of a category c is defined as

$$\gamma(q_{c, r}) = \frac{1}{1 + \exp\left(-\mu\left(p(q_{c, r}) - \left(\frac{1}{N_{c_q}} - \Delta(c, q_{c, r})\right)\right)\right)} \quad (11)$$

where μ is a gain coefficient. This index is called the degree of categorical class.

The conditional probability of key features for a category c is defined for a set of typical classes $Q_c^* = \{q_{c, r} | \gamma(q_{c, r}) \geq \theta, q_{c, r} \in Q_c\}$ as

$$p(f_n | Q_c^*) = \sum_{q_{c, r} \in Q_c^*} (\lambda(q_{c, r}) \times p(f_n | q_{c, r})) \quad (12)$$

$$\lambda(q_{c, r}) = \frac{p(q_{c, r})}{\sum_{q_{c, r'} \in Q_c^*} p(q_{c, r'})} \quad (13)$$

where θ is a threshold which determines whether or not a class is typical.

4 Experiments

4.1 Experimental Framework

To evaluate probabilistic learning of object composition of categories, experiments were conducted by using images of the Caltech-256 image database [14]. For each of 20 categories, 4 images, each of which contains one or a few categorical objects in a crowded scene, were selected and a set of co-occurring segments were extracted by attention-mediated perceptual organization for each category. Fig. 2 shows some categorical images and co-occurring segments with labels for

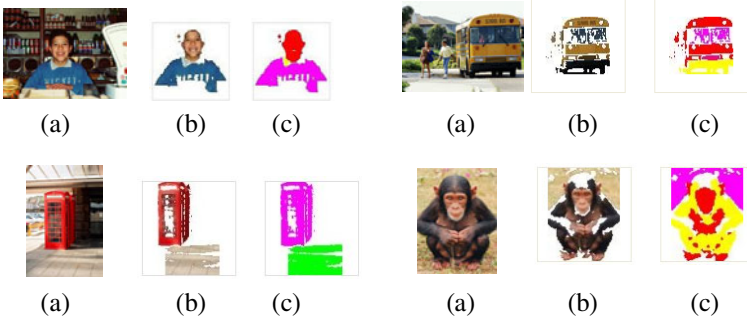


Fig. 2. Examples of (a)images, (b)co-occurring segments and (c)labels. Images of 20 categories (“bear”, “butterfly”, “chimp”, “dog”, “elk”, “frog”, “giraffe”, “goldfish”, “grasshopper”, “helicopter”, “hibiscus”, “horse”, “hummingbird”, “ipod”, “iris”, “palm-tree”, “people”, “school-bus”, “skyscraper” and “telephone-box”) were used in experiments. Different labels are illustrated by different colors.

the images. The number of salient points (that is, SIFT features) which were extracted from all segments is 76019. The code book size of key features which were obtained by the K-tree method is 438. In learning object composition of categories, 181 segments whose number of salient points were more than 100 were used.

Main learning parameters are set as follows. As for learning of multi-class classifiers, a threshold of class division is 0.07 and a correction coefficient in the expression (8) is 2.0. In the tempered EM, a temperature coefficient was decreased by multiplying it by 0.95 at every 20 iterations until it became 0.8. As for inter-categorical typicality analysis, a gain coefficient in the expression (11) is 5.0 and a threshold for determining typical classes is 0.47.

4.2 Experimental Results

The analysis was performed for class composition of categorical multi-class classifiers and characterization of categories by conditional probability of key features.

Fig. 3 shows PLCTs of multi-class classifiers for some categories. In Fig. 3, a typical segment of a class r of each category c is a segment s_{c,i_j} that maximizes $p(q_{c,r}|s_{c,i_j})$. Also, a typical co-occurring segment of each category c is a co-occurring segment $s_c = \{s_{c,i_k}|k \in K\}$ that maximizes the following typicality index $R(s_c) \equiv \sum_{k \in K} \max_{q_{c,r} \in Q_c^*} p(q_{c,r}|s_{c,i_k})$, where Q_c^* is a set of typical classes of the category c . The mean number of classes and typical classes per PLCT for 20 categories are 7.55 and 4.05 respectively. As shown in Fig. 3, typical classes mainly distinguish segments of categorical objects but classes of frequent co-occurring contextual segments also become typical ones. For example, all the 3 typical classes of a “butterfly” category distinguish segments of the categorical object, that is, segments of butterflies. In a “hibiscus” category, 4 classes out of 5 typical classes distinguish segments of the categorical object and another one

distinguishes contextual segments. Also in a “helicopter” category, 2 classes out of 3 typical classes distinguish segments of the categorical object and another one distinguishes contextual segments. According to the expression (□) of the degree of categorical class, a class becomes typical if it has a high class probability and its feature does not appear in other categories. In a “hibiscus” category, two classes of categorical objects with low class probabilities are selected as typical classes because their feature does not appear in other categories. Two classes of contextual segments with high class probabilities remain non-typical because their feature is shared in many categories. On the other hand, in a “helicopter”

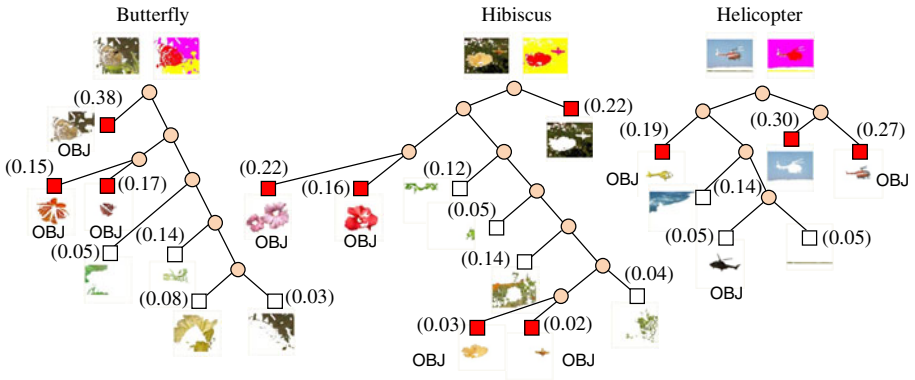


Fig. 3. PLCTs of multi-class classifiers for some categories. A colored square represents a typical class, a white square represents a non-typical class and a value in a parenthesis represents a class probability. A typical co-occurring segment of each category is depicted above a PLCT. A typical segment of each class is depicted beside the class and the “OBJ” mark shows that it is a categorical object segment.

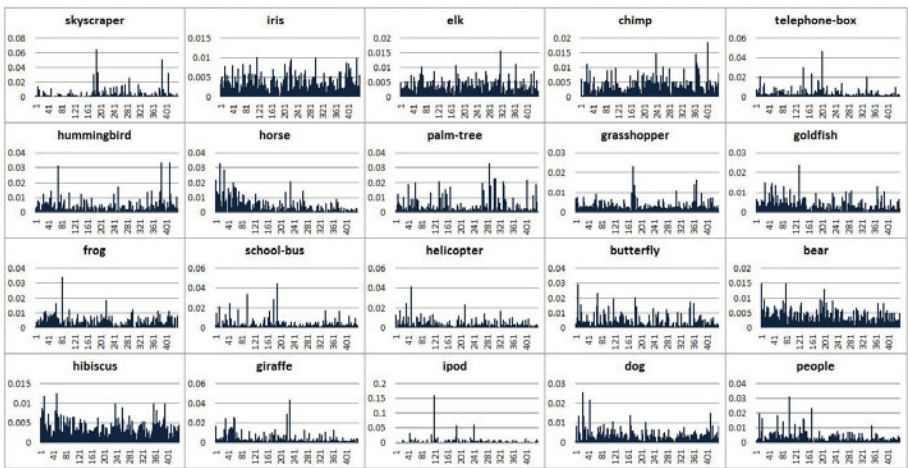


Fig. 4. Conditional probabilities of key features for all categories

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1.skyscraper	0	0.7	0.68	0.66	0.66	0.66	0.83	0.7	0.67	0.75	0.81	0.66	0.72	0.82	0.7	0.68	0.76	0.67	0.66	0.78
2.iris	0.7	0	0.46	0.47	0.62	0.62	0.59	0.68	0.48	0.48	0.47	0.65	0.57	0.56	0.4	0.39	0.58	0.81	0.49	0.55
3.elk	0.68	0.46	0	0.49	0.58	0.61	0.58	0.66	0.44	0.51	0.55	0.61	0.56	0.61	0.47	0.47	0.57	0.75	0.54	0.51
4.chimp	0.66	0.47	0.49	0	0.62	0.63	0.67	0.68	0.51	0.58	0.62	0.59	0.62	0.65	0.53	0.5	0.64	0.76	0.52	0.58
5.telephone-box	0.66	0.62	0.58	0.62	0	0.64	0.69	0.6	0.57	0.65	0.67	0.55	0.6	0.64	0.54	0.55	0.61	0.75	0.57	0.58
6.hummingbird	0.66	0.62	0.61	0.63	0.64	0	0.74	0.72	0.55	0.66	0.68	0.65	0.63	0.73	0.62	0.56	0.69	0.79	0.55	0.67
7.horse	0.83	0.59	0.58	0.67	0.69	0.74	0	0.72	0.61	0.57	0.57	0.77	0.66	0.65	0.58	0.55	0.54	0.91	0.66	0.59
8.palm-tree	0.7	0.68	0.66	0.68	0.6	0.72	0.72	0	0.61	0.66	0.73	0.7	0.67	0.68	0.64	0.61	0.66	0.79	0.65	0.62
9.grasshopper	0.67	0.48	0.44	0.51	0.57	0.55	0.61	0.61	0	0.51	0.56	0.61	0.57	0.54	0.48	0.44	0.59	0.75	0.49	0.53
10.goldfish	0.75	0.48	0.51	0.58	0.65	0.66	0.57	0.66	0.51	0	0.51	0.69	0.63	0.6	0.51	0.43	0.57	0.8	0.57	0.53
11.frog	0.81	0.47	0.55	0.62	0.67	0.68	0.57	0.73	0.56	0.51	0	0.72	0.64	0.64	0.56	0.5	0.6	0.85	0.57	0.59
12.school-bus	0.66	0.65	0.61	0.59	0.55	0.65	0.77	0.7	0.61	0.69	0.72	0	0.64	0.73	0.61	0.61	0.68	0.73	0.55	0.7
13.helicopter	0.72	0.57	0.56	0.62	0.6	0.63	0.66	0.67	0.57	0.63	0.64	0.64	0	0.62	0.57	0.56	0.64	0.8	0.59	0.57
14.butterfly	0.82	0.56	0.61	0.65	0.64	0.73	0.65	0.68	0.54	0.6	0.64	0.73	0.62	0	0.61	0.55	0.67	0.84	0.66	0.5
15.bear	0.7	0.4	0.47	0.53	0.54	0.62	0.58	0.64	0.48	0.51	0.56	0.61	0.57	0.61	0	0.43	0.52	0.8	0.52	0.58
16.hibiscus	0.68	0.39	0.47	0.5	0.55	0.56	0.55	0.61	0.44	0.43	0.5	0.61	0.56	0.55	0.43	0	0.54	0.78	0.5	0.54
17.giraffe	0.76	0.58	0.57	0.64	0.61	0.69	0.54	0.66	0.59	0.57	0.6	0.68	0.64	0.67	0.52	0.54	0	0.84	0.64	0.58
18.ipod	0.67	0.81	0.75	0.76	0.75	0.79	0.91	0.79	0.75	0.8	0.85	0.73	0.8	0.84	0.8	0.78	0.84	0	0.74	0.84
19.dog	0.66	0.49	0.54	0.52	0.57	0.55	0.66	0.65	0.49	0.57	0.57	0.55	0.59	0.66	0.52	0.5	0.64	0.74	0	0.58
20.people	0.78	0.55	0.51	0.58	0.58	0.67	0.59	0.62	0.53	0.53	0.59	0.7	0.57	0.5	0.58	0.54	0.58	0.84	0.58	0

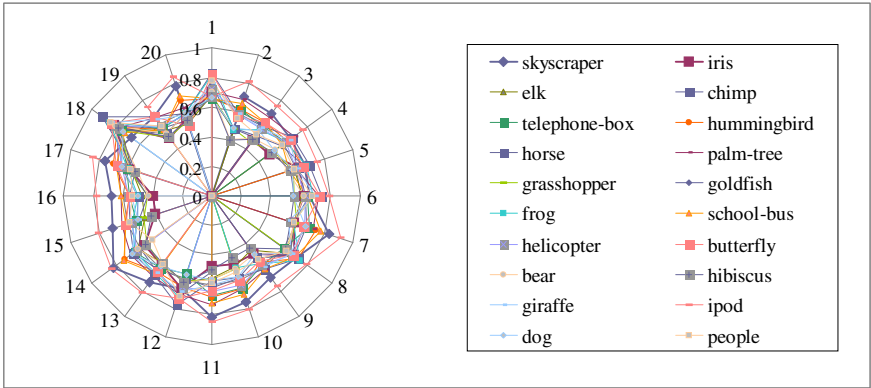


Fig. 5. Distance between conditional probabilities of categorical key features

category, there exists a non-typical class of a categorical object since its feature is exceptional in the category.

Fig. 4 shows conditional probabilities of key features for all categories and Fig. 5 shows distance between each pair of them which is defined by the expression (9). The mean distance of all pairs of categories is 0.62. It is possible to distinguish each category from others by the conditional probabilities of key features for categories since the distance between them is large and they encode mainly features of typical categorical objects through selective attention and typicality analysis.

5 Conclusions

We have proposed a probabilistic model that learns categorical object composition from attended segments. In this model, object composition of categories is learned from a set of BoF of attended segments based on intra-categorical probabilistic

latent component analysis with variable number of classes and inter-categorical typicality analysis. Through experiments by using images of plural categories in the Caltech-256 image database, it was shown that the model learned a probabilistic structure which distinguished intra-categorical composition of objects and context and inter-categorical difference.

The main future work is to build an integrated category and object recognizer which makes full use of this probabilistic structure.

References

1. Atsumi, M.: A probabilistic model of visual attention and perceptual organization for constructive object recognition. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5876, pp. 778–787. Springer, Heidelberg (2009)
2. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc. of ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
3. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 42, 177–196 (2001)
4. Shashanka, M., Raj, B., Smaragdis, P.: Probabilistic latent variable models as nonnegative factorizations. In: *Computational Intelligence and Neuroscience* (2008)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
6. Shlomo, G.: K-tree; a height balanced tree structured vector quantizer. In: Proc. of the 2000 IEEE Signal Processing Society Workshop, vol. 1, pp. 271–280 (2000)
7. Walther, D., Rutishauser, U., Koch, C., Perona, P.: Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding* 100, 41–63 (2005)
8. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: Proc. of IEEE ICCV, pp. 370–377 (2005)
9. Rabinovich, A., Vedaldi, C., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: Proc. of IEEE ICCV (2007)
10. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: Proc. of IEEE CS Conf. on CVPR (2008)
11. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20, 1254–1259 (1998)
12. Atsumi, M.: Stochastic attentional selection and shift on the visual attention pyramid. In: Proc. of the 5th International Conference on Computer Vision Systems, CD-ROM, p. 10 (2007)
13. Zhang, J.: The mean field theory in EM procedures for Markov random fields. *IEEE Trans. on Signal Processing* 40, 2570–2583 (1992)
14. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)

Propagating Uncertainty in Petri Nets for Activity Recognition

Gal Lavee¹, Michael Rudzsky¹, and Ehud Rivlin^{1,2}

¹ Computer Science Department, Technion, Israel Institute of Technology,
Haifa, Israel

² Google Inc., Mountain View, CA, USA

Abstract. Petri Nets is a formalism that has recently been proposed for the specification of models for use in activity recognition. This formalism is attractive because of its inherent ability to model partial ordering, concurrency, logical and temporal relations between the events that compose activities. The main novelty of this work is a probabilistic mechanism (based on the particle filter) for recognizing activities modeled as Petri Nets in video. This mechanism takes into account the observation and semantic uncertainty inherent in low-level events and propagates it into a probabilistic activity recognition.

1 Introduction

The problem of determining whether one or more *activities* are occurring in a surveillance video sequence is an important one which has received much attention in recent years. Informally, an activity is a partially ordered set of *events*. An event is a low-level occurrence which is detectable by low-level computer vision methods such as object detection and tracking.

It is generally agreed that events in video can only be observed up to a certain confidence. One type of uncertainty that exists is observation uncertainty, which results from probabilistic estimation of object locations in the video. Another type of uncertainty, semantic uncertainty, also exists in the interpretation of video occurrences. This kind of uncertainty is caused by the fuzziness inherent in semantic concepts such as ‘close to’ or ‘inside zone’. These type of concepts are often used to specify activity models.

Several recent works have proposed the Petri Net formalism for specification of manually constructed models of activities [1,2,3,4]. This formalism has several advantages for modeling semantic knowledge including straightforward modeling of concurrency and partial ordering.

These works however, do not take into account the uncertainty inherent in both low-level observations and semantic definitions of events. The event observations input into the activity analysis are generally considered to be absolutely certain. In fact, even those approaches that use event detectors which provide a confidence value, elect to threshold this value to arrive at a binary decision on whether the event did or did not occur.

In this paper we offer a method that , given a Petri Net specification of an activity and a set of uncertain events obtained by low-level processing of surveillance video, allows a recognition of the activity which takes into account the uncertainty of the events.

The remainder of this paper is organized as follows: Section 2 discusses related work in activity recognition. Section 3 and 4 provide some introductory background information on the concepts of Petri Nets and Particle Filters, respectively. Section 5 explains the details of our approach along with an illustrative example. Section 6 describes our experiments and show the results obtained. Section 7 provides some discussion of the results and compares our approach to those of other works. Finally, we conclude the paper in Section 8.

2 Related Work

Generally, activity models are hand-built by a human domain expert and formally specified to allow recognition by a machine algorithm. Some of the formalisms proposed for this specification include Predicate Logic[5], Stochastic Context Free Grammars [6], Situation Graph Trees [7,8], Propagation Nets[9] and Petri Nets [10,2,4,3].

In most cases, the algorithms proposed for the recognition of the activities are deterministic in nature. In [5] a set of recognized activities (scenarios) is kept. At each frame the observed events may complete a simple activity, which in turn may complete a larger activity. In [4,2] an event observation moves a token through a Petri Net model of the activity towards the place node indicating the recognition of the activity.

The Situation Graph Tree(SGT) [7,8] is another deterministic formalism which is robust in representing generalization and specialization, but is limited in representing concurrency. Furthermore non-sequential temporal relations are not straightforward to model.

Probabilistic Petri nets[3] have been proposed to give a probabilistic score of a sequence of event observations. The ‘probability’ of the activity in this work does not take into account the uncertainty in the observation of events, but rather describes how well the observations fit into some known probability distribution over sequences of events. All events are assumed to have occurred with absolute certainty.

Propagation Nets [9] describe a formalism which allows incorporating the uncertainty of observations as well as the representational strength of concurrency and partial ordering. These models are more restrictive than the ones we propose in this work. This approach makes assumptions about durations of their internal states that may not be robust to activities that can occur with a large variance in their temporal extent.

Particle Petri Nets [11] have been proposed for monitoring of aircraft. Unlike our representation scheme, this approach does not allow uncertain evidence to be considered and cannot be used to consider activities with multiple objects.

3 Background on Petri Nets

Petri Nets(PN) are specified as a directed bipartite graph. Graphically, *Place* nodes are represented as circles and *transition* nodes are represented as rectangles. Place nodes may hold *tokens* and transition nodes specify the movement of tokens between places when a state change occurs. A transition node is *enabled* if all input place nodes connected to it (those place nodes with directed arcs going to the transition node) have tokens. A special type of arc called an *inhibitor arc* requires that an input place must not have a token in order for the transition to become enabled. Enabled transition nodes may *fire* and change the distribution of tokens throughout the network. When an enabled transition node fires the tokens in the input place nodes are deleted and new tokens are placed in each of the output place nodes (those place nodes with directed arcs coming from the transition). *Conditional transition* nodes can have an enabling rule applied to them which imposes additional conditions on the enabling of the transition . A PN model *marking* is defined as the instantaneous configuration of tokens in the various place nodes in the PN graph. For further details on the PN formalism interested readers are referred to [12,13].

Modeling video activities with Petri Nets is done by connecting fragments corresponding to the composing events in such a way that enforces the temporal and logical relationships between them. Place nodes are ‘waypoints’ which indicate the progress throughout the activity. Special place nodes indicate the beginning and end of the activity. That is the ‘beginning’ place node will have a token in it when the recognition of the activity is still in its initial state, and the ‘end’ place node will have a token in it when the activity recognition has been completed. Otherwise one or more of the intermediate places will have a token in it. (see example activity PN in Fig. 1) Transition nodes model the observation of the events. That is each transition node is a conditional transition associated with an event label. This event must be observed in order for this transition to fire. The Petri Net formalism also allows modeling such aspects of video activities such as forbidden events and temporal durations. For details on the construction of Petri Net activity models, readers are referred to the literature [2,4,3].

4 Background on Particle Filter

Particle filters, also known as sequential Monte Carlo methods, are techniques for probability density estimation based on sampling. A set of *particles*, or hypotheses of the current state is kept and updated at each time step. Each particle is associated with a weight which indicates how likely its hypothesis is.

The *proposal distribution*, also called the importance distribution, is a distribution over the state of the system at the current time given the state at the previous time and taking into account all observations up to the current time.

Informally, at each discrete time slice, a set of particles is drawn from the proposal distribution based on the particle values at the previous discrete time slice. The weights of these particles are then updated based on how likely each particle hypothesis of the system state was to produce the current observation.

In the commonly used Sequential Importance Resampling particle filter algorithm the particles are resampled from the probability distribution implied by the particle weights, if the *effective number* of particles (the inverse of the sum of squared weights), becomes too small.

Particle filter algorithms are used in many application fields. One popular use in Computer Vision is visual tracking of objects [14]. For more on particle filter approaches the reader is referred to [15].

5 Our Approach

Our approach to activity recognition assumes several inputs: 1. The activity definitions(in the form of Petri Net specifications), 2.Low-level event detectors which provide event detections with a confidence value (based on algorithms such as object detection and tracking) 3. An unlabeled video sequence.

Given these inputs we first compute the transition and observation probabilities from the activity definition. We then apply a mid-level event generation component to the output of the low-level video sequence processing. The output of this component is a list of time-stamped events with associated confidences. These events are input to our activity recognition component which updates its confidence in the recognition of each activity with each event observed, taking into account the uncertainty of the event observation.

5.1 Defining the State Space of an Activity Using Petri Net

In this paper,(see also [3]), we define an *activity* as a Petri Net where each transition node is labeled with a low-level event. Each such activity has a place node designated as its ‘start’ node and a set of place nodes designated the ‘recognized’ place nodes.

More formally an activity Petri Net \mathcal{P} is a tuple $\langle P, T, C, events, \delta, S, F \rangle$ where P is the set of places, T is the set of transitions, C is the set of connecting arcs, and *events* is the set of events that are relevant to the activity $\delta : T \rightarrow events$ is a labeling function mapping transitions to an event label. $S \subset P$ is the place node representing the ‘start’ of the activity and $F \subset P$ is the set of place nodes representing the recognition of the activity.

It is straightforward to construct the set of all reachable states from the initial state. We will denote this set as A in the remainder of the paper. Subset $R \subset A$ denotes the set of all reachable activity states where the activity is considered to be recognized.

The structure of the PN allows us to define the Transition and Observation distributions we require to utilize the Particle Filter.

For each state $a \in A$ the transition probability distribution, $P(x_t|x_{t-1} = a)$, is constructed as follows: The unnormalized probability of staying in the same state , $\hat{P}(x_t = a|x_{t-1} = a)$, is set to 1. Let the minimum distance between $a \in A$ and some $b \in A$ be denoted as τ . The unnormalized probability to transition to state b is then $\hat{P}(x_t = b|x_{t-1} = a) = (\alpha)^\tau$, where $\alpha \in [0, 1]$ is

a parameter. Transition to some $c \in A$ that is not reachable from a is given some small non-zero probability ϵ . That is, $\hat{P}(x_t = c | x_{t-1} = a) = \epsilon$. The final transition probability is attained by normalizing the probability according to $P(x_t = x' | x_{t-1} = a) = \frac{\hat{P}(x_t = x' | x_{t-1} = a)}{\sum_{x'' \in A} \hat{P}(x_t = x'' | x_{t-1} = a)}$

More informally, the construction gives a large probability to staying in the same state, a probability inversely related (by parameter α) to the reachability distance to each reachable state, and a small non-zero probability for each non-reachable state.

The observation probability is computed in a similar fashion. For each relevant event q the unnormalized probability $\hat{P}(y_t = q | x_t = a) = 1$ if there is an enabled transition in marking a with an event label q . Additionally we also set the observation probability to 1 if there is an event labeled q that leads to marking a from some other marking. Otherwise, for any other event label the unnormalized probability is set to ϵ .

The observation probability is then normalized according to $P(y_t = q | x_t = a) = \frac{\hat{P}(y_t = q | x_t = a)}{\sum_{q' \in Y} \hat{P}(y_t = q' | x_t = a)}$

5.2 Propagating Uncertainty

Our approach to the propagation of uncertainty is based on the particle filter. Informally, at each discrete time slice we have a set of hypotheses on the state of the activity which we call particles. The sum over the weights of all particles in a particular state is our confidence that we are currently in this state.

At each time slice we sample a new set of particle values from the *proposal distribution*, which describes the distribution of the current state given the previous state. This distribution takes into account the observations previously seen.

When a new event is observed we update the weights of each of our particles. We also update the proposal distribution. This is because the observation of a particular event changes the likelihood of transition to the next state. For instance observing the event ‘*Visitor in Safe*’ while in a state where an enabled transition is labeled with this event makes the transition to the consequent state more likely. We repeat these steps for each discrete time slice.

More formally, we denote by the set of particles, $\mathbf{x}_t = \{x_t^{(1)}, x_t^{(2)}, x_t^{(3)}, \dots, x_t^{(N)}\}$, where N is the number of particles, t is the time, and each $x_t^{(i)} \in A$, for $i = 1..N$.

The set of weights corresponding to the particles at time t is denoted as $\mathbf{w}_t = \{w_t^{(1)}, w_t^{(2)}, w_t^{(3)}, \dots, w_t^{(N)}\}$. The sum over the weights is constrained to be equal to one. That is, $\sum_i^N w_t^{(i)} = 1$.

The proposal distribution at time t is denoted $\pi_t(x_t | x_{t-1}, y_{1:t})$ to indicate that it is a distribution over the current state x_t which takes into account the previous state x_{t-1} as well as all previous observations $y_{1:t}$.

The transition probability, denoted $P(x_t | x_{t-1})$, and observation probability $P(y_t | x_t)$ are derived from the activity Petri Net.

We initialize the particle set \mathbf{x}_0 by sampling from the Prior distribution $P(x_0)$, which is defined to put the majority of the probability mass in the “start” state of

the activity. The initial proposal distribution is set to be equal to the transition distribution. That is, $\pi_0 = P(x_t|x_{t-1})$.

The corresponding weights are initialized to $1/N$.

At each logical time slice t from 1 to T , we denote the observed event as y_t and its confidence as ψ . We then repeat the following steps:

1. Update the weights of existing particles based on the observation for all $i = 1 \dots N$:

$$\hat{w}_t^{(i)} = w_{t-1}^{(i)} \cdot P(y_t|x_t^{(i)})$$

2. Normalize the weights to sum to 1 for all $i = 1..N$:

$$w_t^{(i)} = \frac{\hat{w}_t^{(i)}}{\sum_{i=1}^N \hat{w}_t^{(i)}}$$

3. Update the proposal distribution :

$$\hat{\pi}_t(x_t|x_{t-1}, y_{1:t}) = \begin{cases} \pi_{t-1}(x_t|x_{t-1}, y_{1:t-1}) \cdot (1 + \psi) & \text{if } x_t \leftarrow x_{t-1}|y_t \\ \pi_{t-1}(x_t|x_{t-1}, y_{1:t-1}) \cdot (1 - \psi) & \text{otherwise} \end{cases}$$

where $x_t \leftarrow x_{t-1}|y_t$ indicates there is a path from state x_{t-1} to state x_t via the observation event y_t in the Petri Net activity specification.

4. Normalize the proposal distribution using the formula:

$$\pi_t^*(x_t|x_{t-1}, y_{1:t}) = \frac{\hat{\pi}_t(x_t|x_{t-1}, y_{1:t})}{\sum_{x_{t-1} \in A} \hat{\pi}_t(x_t|x_{t-1}, y_{1:t})}$$

Where A is the set of all reachable states.

5. Set the new proposal distribution to a weighted combination of the updated proposal distribution and the original transition distribution:

$$\pi_t(x_t|x_{t-1}, y_{1:t}) = \beta \cdot \pi_t^*(x_t|x_{t-1}, y_{1:t}) + (1 - \beta) \cdot P(x_t|x_{t-1})$$

Where $\beta \in [0, 1]$ is a parameter.

This is done so that the model reverts to the ‘natural’ transition model if no relevant events have been observed recently.

6. Sample a new set of particles, representing the distribution over the states in the next time slice using the new proposal distribution, for all $i = 1..N$.

$$x_{t+1}^{(i)} \sim \pi_t(x_{t+1}|x_t^{(i)}, y_{1:t}).$$

The particle weights provide an estimation of how the probability mass is distributed across the space of activity states. We sum the weights of those particles which have values in the set R , the set of recognized states for each activity, to determine the probability that the activity is recognized.

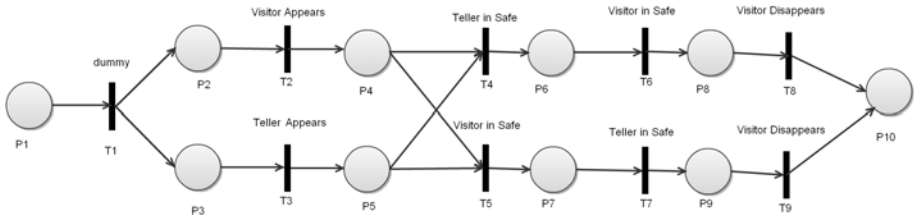


Fig. 1. Activity Petri Net Describing ‘Bank Attack’

5.3 An Example

Consider the Petri Net definition of an activity depicted in Figure 1. This is part of the ‘Bank Attack’ activity used in our experiments which is closely modeled after an activity Petri Net proposed in [3].

The activity Petri Net in the figure is denoted $\mathcal{P} = \langle P, T, C, events, \delta, S, F \rangle$ where $P = \{P_1, P_2, \dots, P_8\}$, $T = \{T_1, T_2, \dots, T_9\}$, C is the set of connecting arcs depicted in the figure, $events = \{ 'Visitor Appears', 'Teller Appears', 'Teller in Safe', 'Visitor in Safe', 'Visitor Disappears' \}$, $\delta(T_2) = 'Visitor Appears'$, $\delta(T_3) = 'Teller Appears'$, $\delta(T_4) = 'Teller in Safe'$, $\delta(T_5) = 'Visitor in Safe'$, $\delta(T_6) = 'Visitor in Safe'$, $\delta(T_7) = 'Teller in Safe'$, $\delta(T_8) = 'Visitor Disappears'$, $\delta(T_9) = 'Visitor Disappears'$, $S = \{P_1\}$, and $F = \{P_{10}\}$

6 Experiments

In our experiments we evaluated our approach on two datasets. Each of these datasets was selected to demonstrate our approach’s effectiveness in dealing with a different type of uncertainty.

The first dataset features activities in a bank surveillance application. We selected the bank domain because it has been used previously in several works [5,3] and activities in this domain have been defined in other works.

In this experiment, semantic components of the activity specification like ‘*the person is in front of the counter*’ and ‘*the two customers are near one another*’ are inherently fuzzy and create semantic uncertainty in the recognition of the activities. This experiment illustrates our methods handling of this type of uncertainty by propagation of the certainty measure associated with the observation to the final activity recognition.

The uncertainty measure for each type of event is calculated in a semantically meaningful way. For example, the uncertainty measure for event ‘*the person is in front of the counter*’ was given by plugging the distance of the tracked object from the pre-defined zone into a sigmoid function.

The second dataset we considered was the ‘Building Entrance’ subset of the ETISEO dataset [16]. We selected this dataset to show our methods can be applied to real data, and to consider the effectiveness of our approach with regards to observation uncertainty. We also show in these experiments that our

approach is straightforward for application in a multi-camera surveillance setting without geometric calibration.

In this dataset we applied particle object tracking methods to give a probabilistic location for each object in each frame (i.e. observation uncertainty). Objects were tracked in multiple camera views. The tracking information was semi-supervised. That is, we allowed a human observer to correct the tracking path when the object track was lost. This type of semi-supervised processing is routinely performed in the domain of activity recognition [2,4,3].

In our experiments we used the parameter values: $\alpha = 0.01$ and $\beta = 0.5$. However, we found our approach fairly robust to these parameters.

6.1 Synthetic Bank Dataset

In our first set of experiments we considered 141 short synthetic video clips lasting from 274-526 frames (9-17 seconds) each. Based on [3] we considered the activities 1. Bank Attack 2. Attempted Bank Attack 3. Normal Customer Interaction 4. Cashier accesses safe 5. Outsider enters safe. We also added the activity 6. Waiting in line. Most of the clips contain one or more of these activities. Some clips, classified as 'Other', contained none of the activities.

Our first experiment seeks to classify the activity(s) occurring in each video clip after having seen the entire clip. To accomplish this we used a threshold parameter θ (set to 0.5 in our experiments). If at the end of the clip the activity's confidence was above this parameter then the clip contains the activity, otherwise the clip does not contain the activity. The results were compared to the ground truth labeling of each of the clips. Recall that multiple or zero activities can be occurring in a clip. We also considered an activity definition called 'bank attack 2' which is an alternate more comprehensive definition of the attack activity.

The accuracy and recall reported in Table 1 are computed as follows : $\text{accuracy} = (tp + tn)/(tp + tn + fp + fn)$, $\text{recall} = tp/(tp + fn)$.

It is interesting to observe at the development of the confidence graph over time. Here each curve represents a different activity. The x axis is the time in the clip and the y axis represents the confidence in the occurrence of each of the activities. For example, let us consider a clip containing an attack event as seen in Figure 2. In this figure you can see that in the initial frames, the 'normal' activity seems to be the most likely, however around frame 140 when the visitor goes behind the counter the confidence of a 'normal' activity goes down, simultaneously the confidence that a 'bank attack 2' or 'attempted bank attack' activity (both containing an event where the attacker goes behind the counter) is taking place goes up. When the visitor enters the safe, the confidence of an attempted bank attack goes down, the confidence of 'bank attack 1' (defined by a safe access) and 'outsider accesses safe' activities increases at this point.

This example illustrates that our method can also be used as a situation is developing to make an assertion on the most likely activities at any given time.

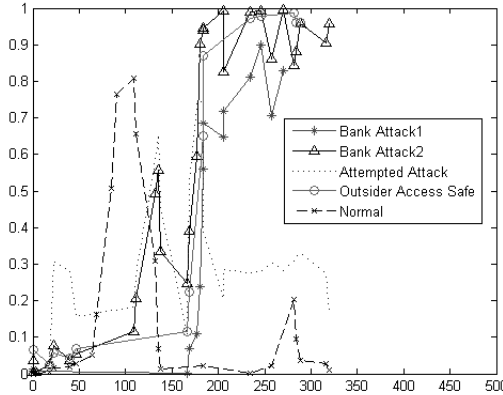


Fig. 2. A plot of the confidence in each of the activities considered vs time in a video sequence containing the activities 'Bank Attack' and 'Cashier Access Safe'

Table 1. Results for synthetic bank dataset **Table 2.** Results for ETISEO Building entrance Dataset

Activity	Accuracy	Recall
Bank Attack	.93	.90
Attempted Attack	.96	1.0
Cashier Access Safe	.84	.93
Normal Customer Interaction	1.0	1.0
Other Activity	1.0	-
Outsider Access Safe	1.0	1.0
Cashier Access Safe	1.0	1.0
Overall	.97	.99

Activity	Accuracy	Recall
Leaving on Foot	.92	.67
Leaving in Car	1.0	1.0
Arriving on Foot	.87	1.0
Arriving by Car	1.0	1.0
Meeting and Walking Together	0.78	1.0
Meeting and Walking Different Directions	.84	0.5
Overall	0.89	0.87

6.2 ETISEO Building Entrance Dataset

The ETISEO Building Entrance is a publicly available dataset of real videos which includes multiple camera views of the scene and includes non-trivial activities consisting of a combination of multiple small events.

This dataset contains 6 sequences from up to 4 camera angles (though not all sequences contain data for all cameras angles). We defined 6 activities that can take place in this domain. Note that different scene objects may be involved in one or more activities.

The sequences in this dataset ranged from 924 to 1649 frames in (30-54 seconds) in length. Each sequence contained one or more of the activities in question. Again we used the confidence of each activity at the end of the sequence in comparison with the ground truth to determine the true positive, true negative,

false positive and false negative detections. Again we used a θ parameter set to 0.5. The accuracy and recall of these experiments are presented in Table 2.

7 Conclusion

The main novelty of this work is enabling activities modeled as Petri Nets to be recognized with a probabilistic mechanism that takes into account the semantic and observation uncertainty of low-level events.

Our framework is based on the principles of the particle filter. We show that this approach allows reasonable results with respect to datasets that include both semantic and observation uncertainty.

References

1. Tessier, C.: Towards a commonsense estimator for activity tracking. Technical Report SS-03-05, AAAI (2003)
2. Lavee, G., Rudzsky, M., Rivlin, E., Borzin, A.: Video event modeling and recognition in generalized stochastic petri nets. *Circuits and Systems for Video Technology* 20, 102–118 (2010)
3. Albanese, M., Chellappa, R., Moscato, V., Picariello, A., Subrahmanian, V.S., Turaga, P., Udrea, O.: A constrained probabilistic petri net framework for human activity detection in video. *IEEE Transactions on Multimedia* 10, 982–996 (2008)
4. Perše, M., Kristan, M., Perš, J., Mušič, G., Vučkovič, G., Kovačič, S.: Analysis of multi-agent activity using petri nets. *Pattern Recogn.* 43, 1491–1501 (2010)
5. Vu, V.T., Bremond, F., Thonnat, M.: Automatic video interpretation: a novel algorithm for temporal scenario recognition. In: *IJCAI 2003*, pp. 1295–1300. Morgan Kaufmann Publishers Inc., San Francisco (2003)
6. Ivanov, Y., Bobick, A.: Recognition of visual activities and interactions by stochastic parsing. In: *CVPR 1998*, vol. 22, p. 852 (1998)
7. Gerber, R., Nagel, H.H.: Representation of occurrences for road vehicle traffic. *Artif. Intell.* 172, 351–391 (2008)
8. Fernández, C., Baiget, P., Roca, X., González, J.: Interpretation of complex situations in a semantic-based surveillance framework. *Image Commun.* 23 (2008)
9. Shi, Y., Huang, Y., Minnen, D., Bobick, A., Essa, I.: Propagation networks for recognition of partially ordered sequential action. In: *CVPR*, vol. 02, pp. 862–869 (2004)
10. Lavee, G., Borzin, A., Rudzsky, M., Rivlin, E.: Building Petri Nets from video event ontologies. In: *International Symposium on Visual Computing* (2007)
11. Lesire, C., Tessier, C.: Particle petri nets for aircraft procedure monitoring under uncertainty. In: *ICATPN*, pp. 329–348 (2005)
12. Kartson, D., Balbo, G., Donatelli, S., Franceschinis, G., Conte, G.: *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, Inc., New York
13. Murata, T.: *Petri Nets: Properties, analysis and applications*. *Proceedings of the IEEE*, 541–580
14. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision* 29, 5–28 (1998)
15. Arnaud, Doucet, Johansen: *A tutorial on particle filtering and smoothing: Fifteen years later*. Technical report (2008)
16. Nghiem, A.T., Bremond, F., Thonnat, M., Valentin, V.: ETISEO, performance evaluation for video surveillance systems. In: *AVSS 2007*, London, UK (2007)

Mixture of Gaussians Exploiting Histograms of Oriented Gradients for Background Subtraction

Tomas Fabian

Department of Computer Science, FEECS,
VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava – Poruba,
Czech Republic
tomas.fabian@vsb.cz

Abstract. Visual surveillance systems include a wide range of related areas ranging from motion detection, moving object classification and tracking to activity understanding. Typical applications include traffic surveillance, CCTV security systems, road sign detection. Each of the above-mentioned applications relies greatly on proper motion segmentation method. Many background subtraction algorithms have been proposed. Simple yet robust frame differencing, statistically based Mixture of Gaussians or sophisticated methods based on wavelets or the optical flow computed by the finite element method. In this paper we focus on novel modification of well known MoG. The intrinsic motivation stems from the inability of regular MoG implementation to handle many camera related phenomena. Here presented method exploits Histograms of Oriented Gradients to significantly reduce the influence of camera jitter, automatic iris adjustment or exposure control causing severe degradation of foreground mask. The robustness of introduced method is shown on series of video sequences exhibiting mentioned phenomena.

1 Introduction

During the last two decades we can see a huge development in area of video surveillance. Many successful approaches and algorithms had appear but in spite of the fact that this area of image processing is extremely various in terms of working environments and objects of interest, there is still a lack of system capable of working in real-world conditions. In this paper we focus on a common approach to identify the moving objects – a background subtraction and we introduce a novel method for robust background modeling.

The modeling of background image plays a very important role in the general process of background subtraction [1]. The resulting foreground mask is processed afterward by various high-level algorithms realizing desired functionality of surveillance or tracking systems used in various places and environments. Solutions for urban traffic surveillance attract great attention of local authorities, especially in the large cities all around the world [2]. The fruitfulness of the system depends greatly on quality of foreground mask which discriminate moving foreground from static background. Many background subtraction methods

work on per-pixel basis, e.g. Wren *et al.* [3] treats every pixel as an independent Gaussian random variable, Stauffer and Grimson [4] proposed the mixture of 3 to 5 Gaussians to represent multi-modal pixel distributions, Elgammal *et al.* [5] introduced the use of kernel density estimation to exploit statistical correlation among neighbor pixels. Mentioned methods suffer from lower foreground mask quality during sudden camera jitter or auto-exposure control intervention. Pre-processing video sequence before segmentation is inefficient while stabilization algorithms for camera motion compensation are unable to restrain sub-pixel motion sufficiently in most cases. The camera vibration may also suffer from rotation and scaling as nonlinear motions that may embarrass the matching process [6].

There also exist methods reflecting the naturalness of pixel adjacency. Im *et al.* in [7] propose wavelet based moving object segmentation using high frequency coefficients in wavelet subbands. Antić *et al.* [8] propose a novel wavelet based method for robust feature extraction and tracking. They also claim, that extremely harsh conditions can occur and violate the premises of the statistical regularity and predictability of background pixels.

The goal of this paper is to eliminate the unfavourable influence of camera related phenomena like sudden changes of overall image brightness caused by the auto-exposure control and to overcome camera jitter from unstable support. The rest of paper is organized as follows. The algorithm is described in Section 2. Evaluation and experimental results are shown in Section 3. Conclusions are given in Section 4.

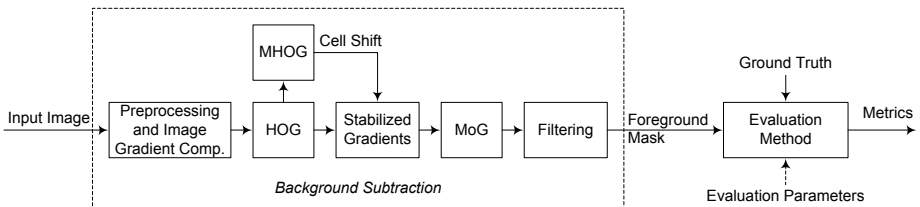


Fig. 1. An overview of proposed background subtraction method. The input image is preprocessed in order to obtain the image of desired resolution and brightness level, and also the image gradient is computed. HOG and mean HOG (MHOG) evaluation follows resulting in the locally dominant gradient stabilized by proper cell-shift. Two-dimensional MoG with the full covariance matrix takes place afterwards. Generated foreground mask is filtered and compared with ground-truth data.

2 Algorithm

In previous section we mentioned two main problems associated with camera jitter and automatic exposure control. To reduce the unwanted influence of such events on quality of foreground mask, we replace the brightness value acting as a random variable in regular MoG algorithm by a local image gradient. The

motivation behind this step is quite easy. Image gradient, and especially its orientation, is invariant against changes in overall image brightness. Of course, this applies only to a certain extent given by limited range of pixel values. In practice, this assumption holds quite well and we are able to reduce the aftereffects of automatic control over exposure that way. But this still will not help us to achieve the independence of foreground mask on camera jitter. We also need to handle small image movements. To do so, we evaluate the gradient not for every single position in the image, but for small squared area, e.g. 8×8 pixels called cell. Further we introduce the on-line spatial rearrangement of cells to minimize the variance of dominant gradient for every cell.

We shall discuss the first step of our approach in more details now. We need to estimate the locally dominant gradients computed on a dense grid of uniformly spaced cells. Histograms of Oriented Gradients (HOG) [9] suits our needs perfectly, i.e. we use $a \times b$ grids of $c \times c$ pixel cells each containing β orientation bins corresponding to certain sector to the full angle. We will refer to the HOG of particular rectangular area i, j of image f using the following notation $\text{hog}_{i,j}(\nabla f) \rightarrow (v_0, v_1, \dots, v_{\beta-1}) \in \mathbb{N}^\beta$, where v_k represents weighted votes for individual angle spans. The weight is proportional to the gradient magnitude. The very first step in HOG evaluation is to estimate the image intensity gradient ∇f in every pixel of input image f . More specifically, the five point central difference formula is used to calculate the image gradient.

Now we need to evaluate HOG for every cell in the image. This is quite straight forward process briefly described in previous paragraph. For more details about computing HOG the reader is referred to [9]. As a result we obtain the most significant orientation bin for every cell in the image.

The computation of the mean HOG, denoted by $\text{mhog}_{i,j}$, follows. The purpose of mean histogram is to provide the reference values for minimizing the variance of gradients and especially resulting bins. We suppose that proper cell-shift vector $\Delta \mathbf{x}$ effectively eliminates camera jitter and other high frequency noise. Let us note that the cell-shift is evaluated for every single cell independently. If the whole image undergoes some small movement or is locally disturbed by some opto-physical process then obtained cell-shifts help us to stabilize the dominant bin values across the time. We also suppose that this will not harm the desired segmentation ability as the changes of the image gradient caused by transit of detected object will be much higher. This can be formalized by the following expression

$$\arg \min_{\Delta \mathbf{x}} (\text{var} (\underbrace{\max (\text{hog}_{i,j} (\nabla f_t (\mathbf{x} + \Delta \mathbf{x})))}_{\text{dominant gradient bin}}))). \quad (1)$$

The validity of both assumptions is discussed in Section 3. To estimate the most probable distribution of bins across the histogram we use on-line averaging as follows $\text{mhog}_{i,j}^t = (1 - \alpha) \text{mhog}_{i,j}^{t-1} + \alpha \text{hog}_{i,j}^t$, where α is the learning rate. Obtained histogram mhog represents the characteristic distribution of gradient orientations for a particular image area during a certain period of time. The similarity of both values mhog and hog indicates the presence of almost the

same pattern in corresponding cell. We are looking for the best match following the spiral like trajectory (with maximal radius r) until the similar bin is found or the border of search area is reached. In the later case we use the shift with smallest distance to the reference value.

Optimal cell-shift can be alternatively found by well-known Lucas-Kanade image alignment algorithm [10]. Just recall that in this case we are looking for alignment minimizing the difference between template image and an input image so that $\sum_x [f(W(\mathbf{x}, \mathbf{p})) - f(\mathbf{x})]^2$, where W represents the set of allowable warps parameterized by \mathbf{p} . For the simplest case of pure translation motion it holds that $\mathbf{p} = \Delta \mathbf{x}$.

To measure the distance between two different bins, we need to introduce some reasonable metric. We have started with very simple metric in a discrete metric space defined as follows

$$\rho_1(x, y) = \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases} \tag{2}$$

The obvious property of metric (2) is it makes no difference between two different bins. Two different bins are equally good and don't matter if they are very similar or completely different. The positive effect of such simplified metric is that the evaluation is very fast. The apparent disadvantage is that in situation when a same bin doesn't exist (and it is true for most cases) we have to make an ad-hoc decision (e.g. we pick the first one). Due to this reason we have defined more complex metric as follows

$$\rho_2(x, y) = \left| uv - v \lfloor u \rfloor - v \left[u - \lfloor u \rfloor + \frac{1}{2} \right] \right|, \tag{3}$$

where $u = |x - y|/v$ and $v = \beta$. With metric (3) we are able to find the best solution of Expression 1 among all valid cell-shifts. As stated before, such solution minimizes the variance of winning bin for each cell and we effectively reduce the unwanted influence of camera jitter. Typical example of resulting cell-shift can be seen in Fig. 3. We provide the Tables 1 and 2 for a better understanding of the meaning of both metrics.

Table 1. Example of the distance matrix of metric ρ_1

ρ_1	0	1	2	3
0	0	1	1	1
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0

Table 2. Example of the distance matrix of metric ρ_2 for particular 4-bin histogram

ρ_2	0	1	2	3
0	0	1	2	1
1	1	0	1	2
2	2	1	0	1
3	1	2	1	0

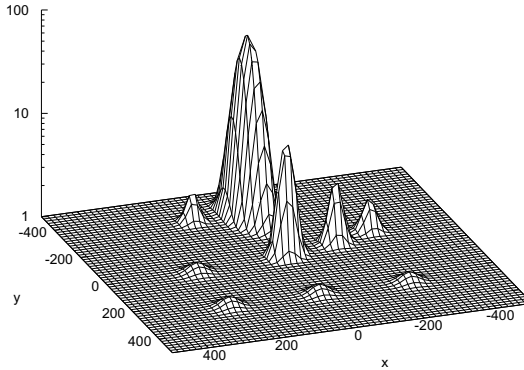


Fig. 2. An example of mixture of 9 Gaussians. Snapshot was taken after the initial stage of background model adaptation so the modes are reflecting the statistical distribution of dominant bins in a single cell. The corresponding cell covers an image area with the presence of a significant horizontal edge as the largest peak is located approximately at the point $(0,-300)$.

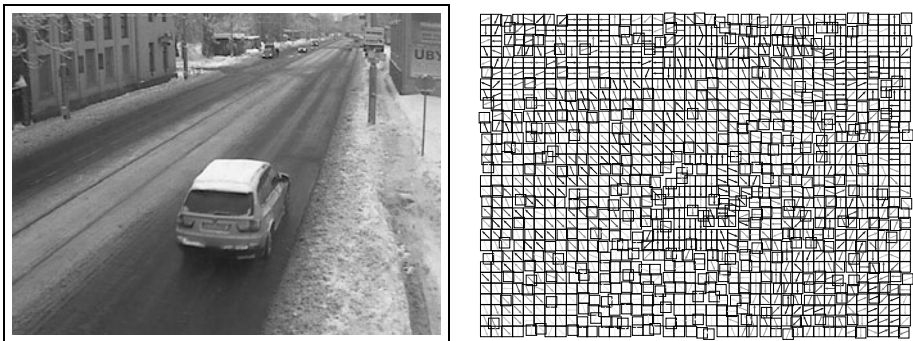


Fig. 3. Histograms of Oriented Gradients; only the most significant bin for each cell is shown. Note the different shifts of cells minimizing the time-variance of dominant bins. Position of every cell obey the minimization criterion of Expression (11), hence it will reduce the influence of camera jitter as well as the high frequency noise.

Now we need to incorporate the dominant bin into the MoG scheme. The general MoG is very well described in [41] and the related maximum-likelihood parameter estimation problem can be found in [12], thus we will not repeat here the whole theory again and we restrict ourselves to only those parts that are different. Authors in [4] consider the values of particular pixels over time as a "pixel process". Similarly in our approach we can talk about "edge process" which consist of time series of observations $\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_t\}$, where \mathbf{X}_i represents the dominant gradient. The process is modelled by a mixture of K Gaussian densities with the set of parameters Θ_k , one for each state k

$$f_{\mathbf{X}|k}(X|k, \Theta_k) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu_k)^T \Sigma_k^{-1}(X-\mu_k)}, \quad (4)$$

where $\Theta_k = \{\mu_k, \Sigma_k\}$ and Σ_k is the full covariance matrix (see Fig. 2). Since we have two-dimensional gradient, we set $n = 2$. For computational reasons and the assumption that the red, green, and blue pixel values are independent and have the same variances, many authors assume the covariance matrix to be of the form $\Sigma_k = \sigma_k^2 \mathbf{I}$. This is not our case as we need to maintain the ability of general Gaussian distribution to represent the elongated data sets (as the orientation of bin is likely to be more stable than the gradient magnitude).

The rest of MoG implementation is same as described in [11]. Estimation of parameters μ_k and Σ_k strictly follows [11, 12]. The segmentation phase depends on proper estimation of value ω_τ which represents the classification threshold. If the probability of certain surface k is higher than ω_τ then the surface is regarded as a background. The overall diagram of the proposed method is shown in Fig. 1.

3 Performance Evaluation

In this section we need to define how to assess the quality of foreground detection. Proposed algorithm, as well as other background subtraction methods, produces a binary image in which we need to identify correctly identified pixels. To do so, we have generated a ground truth mask for three test sequences. Sequences were captured under different lighting conditions. Manual generation of ground-truth data in case of long sequences is too difficult and time-consuming, so we have decided to generate reference mask in the following way. Test sequences were captured under nearly ideal conditions, i.e. no automatic exposure adjustment nor exposure change nor camera jitter has occurred. In this case, we can use even simple background subtraction algorithm (e.g. frame differencing with carefully selected single frame acting as a reference background image) to obtain high quality foreground mask. During the testing phase we apply predefined filters on original video sequence to simulate desired camera related phenomena. Namely, we use translation to simulate camera jitter and we also perform both linear and nonlinear point operations to introduce the effect of automatic camera exposure control (for further details refer to the Fig. 4). In this way our approach allows us to control the amount of spurious effects introduced into the test sequences. Other authors propose the use of semi-synthetic ground-truth sequences where previously segmented tracked objects are artificially inserted into real video sequences.

We adopt two types of pixel-based metrics. The first one comes from [13]. The absolute error e_a is defined as follows

$$e_a = \frac{N_{FP} + N_{FN}}{a \cdot b}. \quad (5)$$

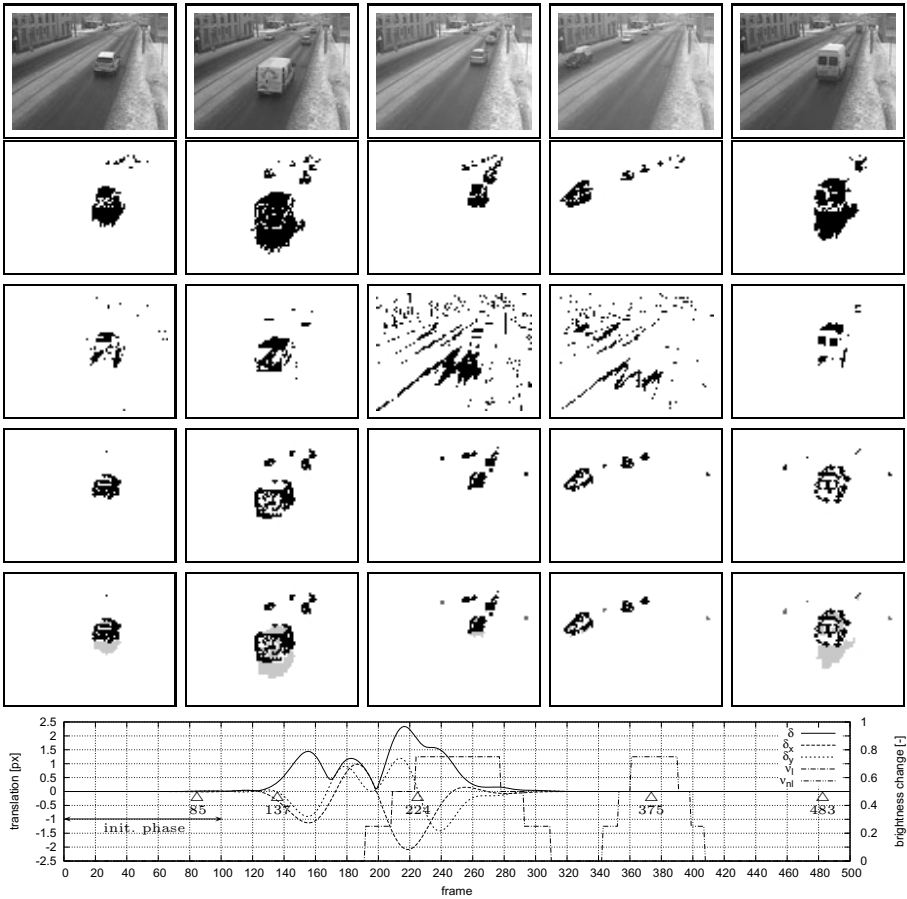


Fig. 4. Evaluation of proposed method. The first row represents five selected frames from the first video sequence captured by the stationary camera, the second row shows reference foreground mask, the third row contains MoG output, the fourth row presents proposed method’s output, and the fifth row compares output of presented method with reference mask (black - TP, white - TN, light grey - FN, dark grey - FP). It is apparent, that the original MoG cannot handle the sudden changes of pixel values and the proposed method generates valid foreground mask (frames 224 and 375). Note that the reference segmentation also includes shadows, what is slightly inappropriate, since the proposed method correctly evaluates shadowed areas as a background. The plot at the bottom of the figure shows values of three coefficients; δ , resp. δ_x and δ_y - image translation simulating camera jitter, ν_l - linear point operation (brightness change equals to $30 \cdot \nu_l$), ν_{nl} - nonlinear point operation (gamma correction with $\gamma = 1 / (1 + 0.5\nu_{nl})$). Positions of selected frames are also marked. Initialization phase takes 100 frames and the learning rate α is 100-times higher during this period.

Table 3. Contingency table of four possible conclusions that can be drawn in a statistical hypothesis test

Real output	Ground Truth	
	Foreground	Background
Foreground	True Positive (TP)	False Positive (FP)
Background	False Negative (FN)	True Negative (TN)

For the meaning of subscripts of number of pixels N please refer to the Table 3. We recall that a and b stand for foreground mask dimensions. The second metric e_g is defined subsequently

$$e_g = \frac{1}{|\Omega_f|} \sum_{x \in \Omega_f} \min_{y \in \Omega_f^1} \|\mathbf{x} - \mathbf{y}\|_2, \quad (6)$$

where Ω_f represents the set of foreground pixel coordinates in actual foreground mask, Ω_f^1 stands for the set of foreground pixels in reference foreground mask. Simply put, metric e_g returns the average minimum distance between all pixels of the current mask and the foreground pixels of the reference mask. Both metrics are calculated for each frame and overall values are obtained by averaging over the entire test sequence.

Table 4. The results of original MoG and proposed method compared with the reference background mask

	Method	TP	TN	FN	FP	e_a	e_g
seq. 1	MoG	95.8	4462.5	146.5	95.2	5.0%	1755.5
	MoG + HOG	82.1	4528.3	160.2	29.4	3.9%	81.7
seq. 2	MoG	398.8	3637.2	665.8	98.2	15.9%	832.4
	MoG + HOG	333.7	3657.0	730.9	78.3	16.9%	199.0
seq. 3	MoG	44.3	4651.5	63.2	41.0	2.2%	692.8
	MoG + HOG	55.3	4669.2	52.2	23.3	1.6%	186.3

The most important columns in Table 4 are FP and e_g . We can see that the number of pixels falsely marked as a foreground is significantly lower in the case of presented method and direct observation from Fig. 4 is consistent with the measured values. Higher values of FN can be explained by the presence of shadows in reference foreground mask. We should also mention that all parameters and their associated values are stated in Table 5.

Table 5. Essential values of presented method's parameters used during evaluation

a	b	c	r	α	β	metric	K	σ_{init}	ω_τ
80	60	8	5	0.001	16	ρ_2	9	500.0	0.3

4 Conclusion

In this paper we have proposed a new algorithm for background subtraction which is based on two well known methods: Mixture of Gaussians and Histograms of Oriented Gradients. Resulting method is significantly less vulnerable to the most common camera related phenomena like jitter and automatic exposure control. Higher level algorithms may benefit from the more reliable foreground mask and may track the moving objects even during the moments, when the ordinal MoG is failing. Tests performed on three video sequences with a total length of 1500 frames confirmed the assumptions made in the introductory section. Presented method is computationally more expensive than ordinal MoG. Focus should be placed on more effective implementation of this method and especially on evaluation of the gradient histograms and probability density functions with full covariance matrix.

Acknowledgement

This work was partially supported by the grant FR-TII/262 of the Ministry of Industry and Trade of the Czech Republic.

References

1. Cheung, S.C.S., Kamath, C.: Robust techniques for background subtraction in urban traffic. In: Chen, L., Ryan, M.D., Wang, G. (eds.) ICICS 2008. LNCS, vol. 5308, pp. 881–892. Springer, Heidelberg (2008)
2. Buch, N., Yin, F., Orwell, J., Makris, D., Velastin, S.A.: Urban vehicle tracking using a combined 3d model detector and classifier. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based and Intelligent Information and Engineering Systems. LNCS, vol. 5711, pp. 169–176. Springer, Heidelberg (2009)
3. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 780–785 (1997)
4. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252 (1999)
5. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S., Duraiswami, R., Harwood, D.: Background and foreground modeling using nonparametric kernel density for visual surveillance. *Proceedings of the IEEE*, 1151–1163 (2002)
6. Pisheh, M.A.Z., Sheikhi, A.: Detection and compensation of image sequence jitter due to an unstable ccd camera for video tracking of a moving target. In: *3DPVT 2004: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, Washington, DC, USA, pp. 258–261. *IEEE Computer Society, Los Alamitos* (2004)
7. Im, T.H., Eom, I.K., Kim, Y.S.: Wavelet-based moving object segmentation using background registration technique. In: *SIP 2007: Proceedings of the Ninth IASTED International Conference on Signal and Image Processing*, Anaheim, CA, USA, pp. 84–88. *ACTA Press* (2007)

8. Antic, B., Castaneda, J., Culibrk, D., Pizurica, A., Crnojevic, V., Philips, W.: Robust detection and tracking of moving objects in traffic video surveillance. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 494–505. Springer, Heidelberg (2009)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (eds.) International Conference on Computer Vision & Pattern Recognition. INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, vol. 2, pp. 886–893 (2005)
10. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI 1981: Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco (1981)
11. Power, P.W., Schoonees, J.A.: Understanding background mixture models for foreground segmentation. In: Proceedings of the Image and Vision Computing, pp. 267–271 (2002)
12. Bilmes, J.: A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute (1998)
13. Stefano, L.D., Neri, G., Viarani, E.: Analysis of pixel-level algorithms for video surveillance applications. In: 11th International Conference on Image Analysis and Processing, ICIAP 2001, pp. 542–546 (2001)

Human Pose Recognition Using Chamfer Distance in Reduced Background Edge for Human-Robot Interaction

Anjin Park and Keechul Jung

Department of Digital Media, Soongsil University, Seoul, Korea
{anjin.park, jungkeechul}@gmail.com

Abstract. Human pose estimation and recognition have recently attracted a lot of attention in the field of human-computer interface (HCI) and human-robot interface (HRI). This paper proposes human pose recognition method using a chamfer distance that computes similarities between an input image and pose templates stored in database. However, the chamfer distance has a disadvantage that it may produce false-positive in regions where similar structures in edge images as templates exist, even when no human pose is present. To tackle this problem, the proposed method tries to adaptively attenuate the edges in the background while preserving the edges across foreground/background boundaries and inside the foreground. The proposed algorithm builds on a key observation that edge information in the background is static when a human takes pose as the interface. Moreover, the algorithm additionally considers edge orientation to minimize loss of foreground edges, caused by edge attenuation. In the experiments, the proposed method is applied to the HRI. Edge information for the background is modeled when the robot stops in front of the human for interaction with gesture. The performance of the proposed method, time cost and accuracy, was better than the chamfer distance and pictorial structure method that estimates human pose.

1 Introduction

The human pose recognition is one of methodologies that interact computers with a human body, and thus can be utilized in several applications, such as robotics controlled by the human's posture in the field of human-robot interface (HRI) and interactive games played with the human's behaviors in the field of human-computer interface (HCI) [1].

Pose recognition algorithms have been intensively studied for past decades. For examples, Haritaoglu et al. [2] and Guo and Miao [3] proposed a posture recognition system based on projection histograms on horizontal and vertical axes, and Bradski and Davis [4] and Boulay et al. [5] used Hum moments and Mahalanobis distance to recognize the human pose. Moreover, SVM (support vector machine) and RVM (relevance vector machine), which are the most popular classification algorithms, were also used to recognize the pose by Schindler et al. [6] and Guo and Qian [7]. However, most of the approaches to human pose recognition required the human extraction from an input image as an essential introductory step. This precondition limits the use of these techniques to scenarios where good extractions are made available by

enforcing strict studio condition like blue-screening. Otherwise, a preprocessing step, which is modeling background information, must be performed in an attempt to extract the human, such as [8]. However, since it is not easy to model background information in the case of non-stationary camera, e.g. used as eyes of robot, extracting accurate human posture to be recognized in the next step is one of the most challenging tasks for the human pose recognition system.

The human pose estimation has also been intensively studied in non-stationary camera environments, which does not rely on the foreground extraction [9,10]. This procedure first detects humans from an input image, using several methods, such as Histogram of Gaussian (HoG) features classified by SVM [11], and then, extracts each part (limbs, torso, and head) based on data learnt by learning algorithm, such as Adaboost. This procedure can extract human posture from images captured by non-stationary camera. However, since the parts of human body are extracted based on the learnt data, the method often did not detect each part correctly in real input images, not images used for learning. Moreover, the human pose estimation is time-consuming work, thus it is not possible to perform the pose estimation in real-time application.

Some posture recognition algorithms that do not need foreground extraction or pose estimation exist. They relied on chamfer distance [12,13]. The chamfer distance-based method performs pose recognition by calculating chamfer distance between edges of all of the templates stored in database and distance matrix of the input image. However, the chamfer distance-based matching can easily produce many false positives and some false negative, especially when the background is cluttered.

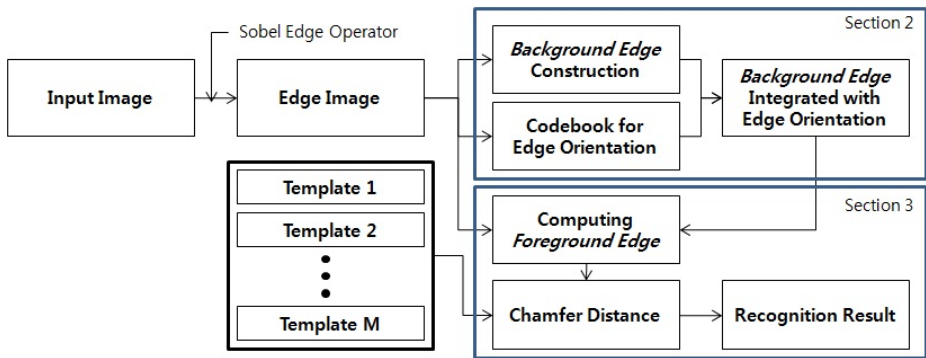


Fig. 1. Flow chart of proposed method

The proposed method is on basis of the chamfer distance-based method. To minimize the false positive caused by the chamfer distance, the proposed method adaptively attenuates the edges regarded as the background while preserving the edges across foreground and background boundaries and inside the foreground. Here, the *background edges* are defined as pixels that are marked as an edge for most of frames, and are constructed by modeling edge information on pixel-basis. The proposed method is applied for the HRI, and the defined *background edges* are reasonable, as the robot should be standing in front of the human for interaction. The *background edges*

construction is started when the human is standing in front of the robot, and is initialized when the robots move to other places. In this paper, the human is detected by detecting face regions. Detecting human or upper body itself is better applicable, but we detects face regions, as detection rates of face regions are better than human body. Edges across the background/foreground boundaries or inside the foreground may often be attenuated if only edge pixel is considered. Therefore, the proposed method additionally considers the edge orientation to construct better *background edges*. The edge orientations at each pixel are quantized into codebooks, as the orientation has not always same value even if the stationary camera captures the image. Fig. 1 shows the flow chart of the proposed method.

The reminder of this paper is as follows. Section 2 describes how to construct *background edge* integrated with edge orientation, and section 3 describes how to obtain *foreground edge* from input images and recognize human pose using the *foreground edge*. Then, some experimental results are presented in section 4, and the final conclusions are given section 5.

2 Background Edge Map Construction

The proposed method is on basis of the chamfer distance that relies on edge information, as edge information is more robust than color or intensity value in the environments where leave all parameters in the web camera at the default setting (auto gain control and auto white balance). Before applying for the chamfer distance to recognize human pose, the proposed method constructs background edge map to reduce the error caused by the chamfer distance, which is mentioned in section 1. Constructing the background edge map consists of two steps: background edge construction and codebook construction for edge orientation. Section 2.1 describes the methods to construct the background edge map and then goes on to describe how to integrate the edge orientation into the background edge map.

2.1 Background Edge Construction

Since the robot is standing in front of the human for interaction, a camera to capture the image including human is stationary while recognizing the human pose. *Background edge* is then constructed for one pixel; the construction is identical for each pixel.

Let us denote the value computed by sobel edge operator on each pixel p by e_p . If the e_p is below the threshold T_e , $fe_p^1 \leftarrow fe_p^1 + 1$, where fe_p^1 denotes the frequency in which the pixel p was an edge for t sequences used for the construction. Otherwise, $fe_p^0 \leftarrow fe_p^0 + 1$, where fe_p^0 denotes the frequency that the pixel p was not an edge. Then, the $t+1^{\text{th}}$ *background edge* is determined by taking winner between fe_p^1 and fe_p^0 . The reason why we used this procedure for the *background edge* construction is that edge information for the background may often be lost if one frame or the small number of frames is used for the construction, as noise or lighting conditions may be added to the input image.

Given an input image, the foreground edge is extracted from an input edge image by differencing the image from the constructed *background edge*. Then, the

background edge is updated with the input edge image. Section 3 will explain how to differ the edge image from constructed edge.

2.2 Codebook for Edge Orientation

The *background edge* described in section 2.1 is constructed with edge intensity. However, it can cause the case that edges for the foreground are lost if edges on the foreground in an input image correspond to ones on the background in the *background edge*. Therefore, the proposed method additionally considers the edge orientation for constructing more reasonable *background edge*.

There is a problem to be solved while considering the edge orientation for the *background edge* construction. The edge orientation is not always holding the same value even if stationary camera is capturing the static background without any moving foreground objects. Therefore, the proposed method adopts a quantization algorithm, codebook, to construct the background edge orientation. For each pixel, the algorithm builds a codebook consisting of one or more codewords. The edge orientations at each pixel are quantized into a set of codewords. The edge map is then encoded on a pixel-by-pixel basis.

I.	$n_c \leftarrow 0, \mathbf{C} \leftarrow \emptyset$
II.	for $t = 1$ to n_θ do
1.	$\theta_t \leftarrow \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) + \frac{\pi}{2}$
2.	Find the best \mathbf{c}_m in $\mathbf{C} = \{\mathbf{c}_i 1 \leq i \leq n_c\}$ that matches θ_t based on the condition.
i.	$\check{\theta}_i < \theta_t < \hat{\theta}_i$
3.	If $\mathbf{C} = \emptyset$ or there is no match, then $n_c = n_c + 1$.
i.	Create a new codeword \mathbf{c}_{n_c} by setting
1.	$\theta_{n_c} \leftarrow \theta_t$ and $\mathbf{aux}_{n_c} = \langle \check{\theta}_i - t_\theta, \hat{\theta}_i + t_\theta, n_c - 1, n_c, 1 \rangle$.
4.	Otherwise, update the matched codeword \mathbf{c}_m , consisting of θ_m and \mathbf{aux}_m by setting
i.	$\theta_m \leftarrow \frac{f_m \theta_m + \theta_t}{f_m + 1}$ and
ii.	$\mathbf{aux}_m = \langle \min\{\check{\theta}_i, \theta_m\}, \max\{\hat{\theta}_i, \theta_m\}, \max\{\tau_m, t - q_m\}, t, f_i + 1, \rangle$
	end for

Fig. 2. Algorithm for *edge map* construction

Fig. 2 shows algorithm to construct codebook for the *edge map*. Let Θ be a sequence for single pixel consisting of n_θ orientation-vector: $\Theta = \{\theta_1, \dots, \theta_{n_\theta}\}$, and let \mathbf{C} be the codebook for a pixel consisting of n_c codewords. Each pixel has a different codebook size based on its orientation variation. Each codebook $\mathbf{C}_i, i = 1, \dots, n_c$ consists of the orientation θ_i and 4-tuple $\mathbf{aux}_i = \langle \check{\theta}_i, \hat{\theta}_i, \tau_i, q_i, f_i \rangle$, where $\check{\theta}_i$ and $\hat{\theta}_i$ denote the minimum orientation and maximum orientation, respectively, of the i th codeword, τ_i denotes the maximum negative run-length (MNRL), which is defined as the longest interval during the constructing period in which the codeword did not recur, q_i denotes the last access time at which the codeword occurred, and f_i is the frequency with which the codeword occurs.

After construction, the codebook may be sizeable because it contains all of the codewords that may include human who is moving his/her body and noise. Therefore, the codebook is refined by eliminating the codewords that contain moving human. The MNRL in the codebook is used to eliminate the codewords that include moving human, based on the assumption that pixels of moving foreground human appear less frequently than backgrounds. Thus, codewords having a large τ are eliminated by the following equation: $\mathbb{C} = \{\mathbf{c}_m | \mathbf{c}_m \in \mathbb{C} \wedge \tau_i \leq T_{\mathbb{C}}\}$, where \mathbb{C} denotes the background model, which is a refined codebook, and $T_{\mathbb{C}}$ denotes the threshold value. In the experiments, $T_{\mathbb{C}}$ was set to be equal to half the number of constructing sequences.

3 Pose Recognition

The *background edge* is constructed for extracting *foreground edge* from an input edge image, which can reduce the errors on pose recognition caused by chamfer distance. The foreground edge is constructed by comparing the input edge image with the background edge constructed at t th sequence.

The comparison is done by satisfying one of two conditions as follows. The first condition is that the pixel is considered as the foreground edge if the pixel that is an edge in the background map is an edge in the input image, as following equation:

$$E_p^F = \{E_p^I | E_p^I = \text{edge} \wedge E_p^B = \text{edge}\}, \quad (1)$$

where E_p^I and E_p^B denote an input edge and the *background edge*, respectively, and E_p^F denotes the *foreground edge* that the edge pixels corresponding to the background edge are attenuated from the input edge image.

The second condition is implemented to save foreground edges removed by the first condition. The codebook for edge orientation \mathbb{C} constructed in the previous stage is used for the condition. After computing edge orientation of each pixel, two parameters, $\check{\theta}_i$ and $\hat{\theta}_i$ for all of the codewords $i = 1, \dots, n_{\mathbb{C}}$, are used to check the edge orientation θ_p of the pixel is within an edge or not, as following equation:

$$E_p^F = \{E_p^I | \check{\theta}_i - t_{\theta} \leq \theta_p \leq \hat{\theta}_i + t_{\theta}\}, \quad (2)$$

After extracting the foreground edge, the proposed method performs pose recognition. We compute Distance Transform f the input image I and each template-size patch of it we match against the template silhouette S represented by oriented silhouette contour. Distance Transform (DT) image is floating point image of the same size as input image, in which value of each pixel represents a distance from that pixel to the nearest feature in the input image. There are many matching measures that can be defined on the distance distribution. In the proposed method, we use Chamfer distance, which, formally, can be defined as follows.

Given the two point sets $S = \{s_i\}_{i=1}^n$ and $C = \{c_j\}_{j=1}^n$ the Chamfer distance function $d_{\text{chamfer}}(S, C)$ is the mean of the distances between each point, $s_i \in S$ and its closest point in C :

$$d_{\text{chamfer}}(S, C) = \frac{1}{n} \sum_{s_i \in S} \delta(s_i, C), \quad (3)$$

where $\delta(s_i, C)$ can be taken to be the distance between s_i and the closest point on C . In our case S represents a template silhouette, and C represents a set of foreground edges. In practice, if we position a template image over the DT input image, $d_{\text{chamfer}}(S, C)$ is simply mean of the pixel values of the DT image which lie under the edge pixels of the template. A silhouette is matched at locations where $d_{\text{chamfer}}(S, C)$ is below a threshold T_{chamfer} and is a local minimum over a specified image area. With the latter condition we secure that we get only one response per person.

In the presence of clutter, defining $d_{\text{chamfer}}(S, C)$ in this manner lacks robustness. To increase it, we take edge orientation into account by introducing a penalty term as follows:

$$h(s_i, c_j) = \tan(\alpha_{s_i} - \beta_{c_j}), \quad (4)$$

where α_{s_i} and β_{c_j} are the edge orientation respectively at the silhouette point s_i and at the foreground edge c_j . To effectively compute the penalty term, we modifies the DT algorithm so that each location in the DT image also contains the edge orientation of the closest edge pixel.

To further reduce the effect of outlier and missing edges, we use the Tukey robust estimator [14], defined by:

$$\rho(d) = \begin{cases} \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{d}{c} \right)^2 \right)^3 \right] & \text{if } |d| \leq c \\ \frac{c^2}{6} & \text{if } |d| > c \end{cases}, \quad (5)$$

The constant c defines a maximal value of the Chamfer distance in pixels after which an edge point is treated as an outlier.

Since our database contains templates at different scales, to allow effective comparison between Chamfer distances, we also introduce a scale factor κ . Its role is to normalize the distance to the value that would be computed, had the template not been scaled. We therefore take $\delta(s_i, C)$ in Eq. 3 to be

$$\delta(s_i, C) = \rho\left(\frac{1}{\kappa} \|s_i - c(s_i)\| + h(s_i, c(s_i))\right), \quad (6)$$

where $c(s_i)$ is the closest point to point s_i , $h(\cdot)$ is the angle penalty term given by Eq. 4 and $\rho(\cdot)$ is the Tukey robust estimator.

4 Experimental Results

The proposed method was applied to HRI. Fig. 3 shows a schematic drawing of a scenario and robots, named RoMAN, used in the experiments. The scenario is as follows. The robot moves to the human and reads the human pose, and then, makes a decision for the next step.

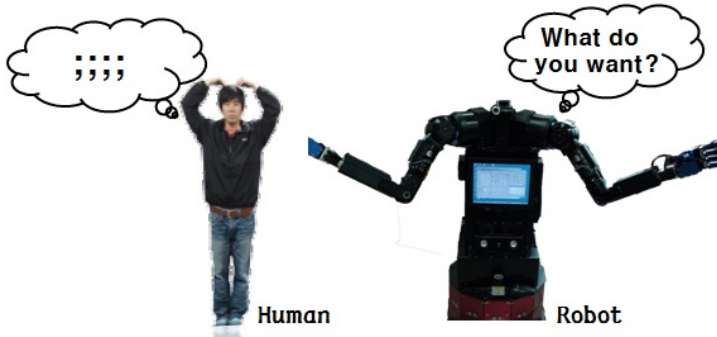


Fig. 3. Schematic drawing

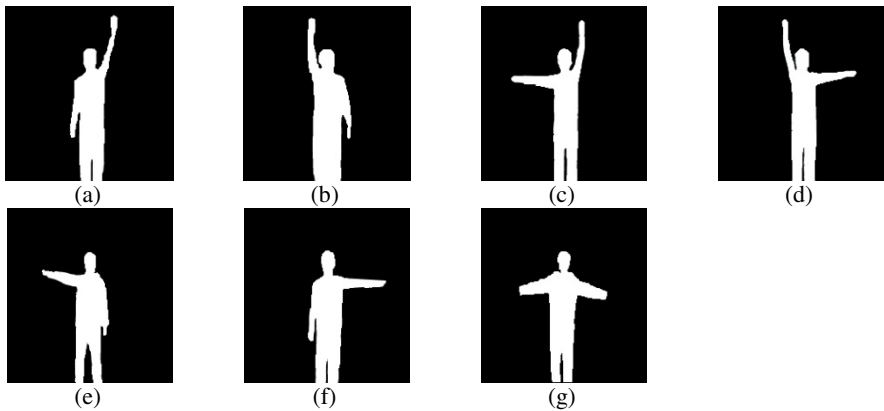
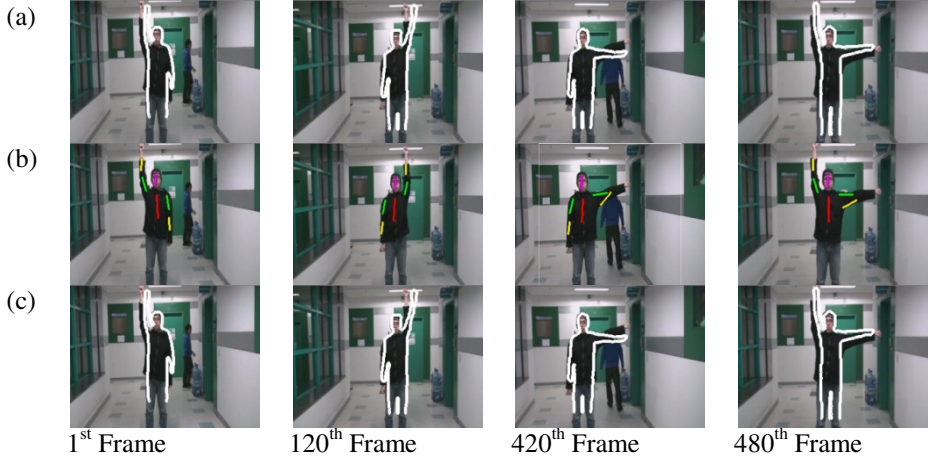


Fig. 4. Set of templates used in the experiments

The data set sequences used to evaluate the proposed method consists of two types: (1) indoor and (2) outdoor environments with simple and cluttered background. Each type consists of four sets, and each set has about 250-500 frames. The pictorial structure-based method [10] that estimates human pose was also compared with the proposed method. The method detected human body and eliminated background as a prerequisite stage, which results in improving precision rates and decreasing the computational time. The human detection is performed by a SVM recognizing features extracted from HoG, and then, the human extraction is tentatively performed by the GrabCut algorithm [15]. The final step is to use the pictorial structure to extract accurate human pose from the extracted human. Fig. 5 shows examples of the proposed method that relies on foreground edge, the chamfer distance-based method that relies on input edge image, and pictorial structure-based method. The proposed method showed better recognition rates on cluttered background.

Indoor Example



Outdoor Example including Cluttered Background

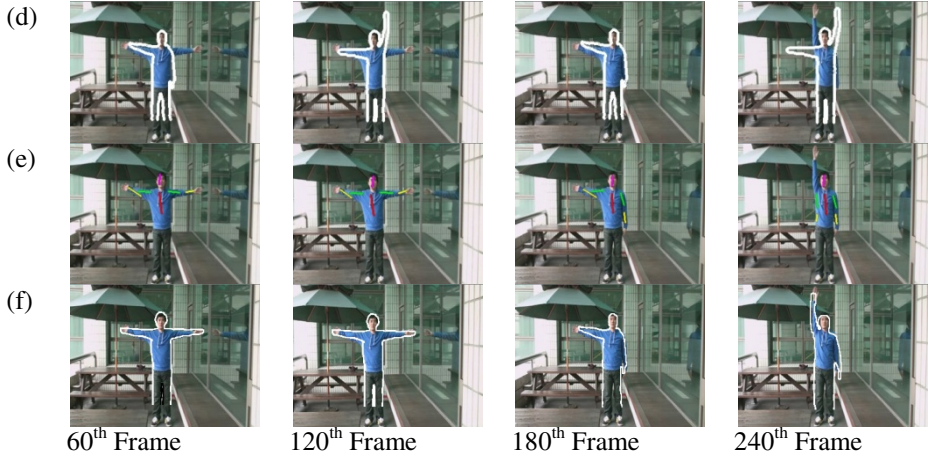


Fig. 5. Resultant images of pose recognition using chamfer distance (a,d), pictorial structure (b,e), and proposed method (c,f)

As shown in Fig. 5, the chamfer distance-based method showed good performance on the simple background, but did not come up to our expectation of accuracy on more cluttered background including complex edges. The pictorial structure-based method showed low recognition rates, as the GrabCut did not extract accurate foreground in cluttered background. Moreover, the method did not estimate the human pose in the case where the human wears full-flowing clothes. Moreover, the computational time for one frame is about 8 second, whereas the proposed method achieved a frame rate of 10 fps. This makes the method well-suited to applications that require real-time processing. Table 1 shows the recognition rate of the proposed method and comparative methods. The performance of the proposed method was better than that of the other methods being compared. We demonstrated the proposed human pose



Fig. 6. Demonstration of the proposed pose recognition system

Table 1. Recognition rates of proposed method and two comparative methods (%)

	Chamfer distance	Pictorial structure	Proposed method
Recognition rates	63.3	57.8	93.3

recognition system at World IT Show 2010¹ held in Korea, as shown in Fig. 6. In the system, the robot recognized the human pose, and then was just trying to be a copycat of the human.

5 Conclusions

This paper proposes human pose recognition method using a chamfer distance that computes similarities between an input image and pose templates stored in database. The proposed method tried to solve the disadvantage of the chamfer distance that it may produce false-positive in regions where similar structures in edge images as templates exist, by adaptively attenuating the edges in the background while preserving the edges across foreground/background boundaries and inside the foreground. The proposed algorithm builds on a key observation that edge information in the background is static when a human takes pose as the interface. Moreover, the algorithm additionally considers edge orientation to minimize loss of foreground edges, caused by edge attenuation. In the experiments, the proposed method is applied to the HRI. Edge information for the background is modeled when the robot stops in front of the human for interaction with gesture.

In the experiments, the proposed method performed better than two comparative methods in finding and recognizing human pose for cluttered background environments.

¹ <http://www.worlditshow.co.kr/>

Acknowledgement. This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency). (NIPA-2009-(C1090-0902-0007)).

References

1. Peng, B., Qian, G., Ma, Y.: Recognizing Body Poses using Multilinear Analysis and Semi-Supervised Learning. *Pattern Recognition Letter* 30, 1289–1297 (2009)
2. Haritaoglu, I., Harwood, D., Davis, L.S.: A Human Body Part Labeling System using Silhouettes. In: *Proceedings of International Conference on Pattern Recognition*, pp. 77–82 (1998)
3. Guo, P., Miao, Z.: Projection Histogram based Human Posture Recognition. In: *Proceedings of International Conference on Signal Processing*, pp. 16–20 (2006)
4. Bradski, G.R., Davis, J.: Motion Segmentation and Pose Recognition with Motion History Gradients. *Machine Vision and Applications* 13, 174–184 (2002)
5. Boulay, B., Bremond, F., Thonnat, M.: Applying 3D Human Model in a Posture Recognition System. *Pattern Recognition Letter* 27, 1788–1796 (2006)
6. Schindler, K., Gool, L.V., Gelder, B.: Recognizing Emotions Expressed by Body Pose: A Biologically Inspired Neural Model. *Neural Networks* 21, 1239–1246 (2008)
7. Guo, F., Qian, G.: Dance Posture Recognition using Wide-Baseline Orthogonal Stereo Cameras. In: *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 481–486 (2006)
8. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfnder: Real-Time Tracking of Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7), 780–785 (1997)
9. Andriluka, M., Roth, S., Schiele, B.: Pictorial Structure Revisited: People Detection and Articulated Pose Estimation. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1014–1021 (2009)
10. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive Search Space Reduction for Human Pose Estimation. In: *Proceeding of International Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
11. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: *Proceeding of International Conference on Computer Vision and Pattern Recognition*, pp. 886–893 (2005)
12. Gavrilla, D., Philomin, V.: Real-Time Object Detection for “Smart” Vehicles. In: *Proceeding of International Conference on Computer Vision*, pp. 87–93 (1999)
13. Dimitrijevic, M., Fua, L.P.: Human Body Pose Detection using Bayesian Spatio-Temporal Templates. *Computer Vision and Image Understanding* 104, 127–139 (2006)
14. Huber, P.: *Robust Statistics*. Wiley, New York (2006)
15. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics* 23, 309–314 (2004)

Modeling Clinical Tumors to Create Reference Data for Tumor Volume Measurement*

Adele P. Peskin¹ and Alden A. Dima²

¹ NIST, Boulder, CO 80305

² NIST, Gaithersburg, MD 20899

Abstract. Expanding on our previously developed method for inserting synthetic objects into clinical computed tomography (CT) data, we model a set of eight clinical tumors that span a range of geometries and locations within the lung. The goal is to create realistic but synthetic tumor data, with known volumes. The set of data we created can be used as ground truth data to compare volumetric methods, particularly for lung tumors attached to vascular material in the lung or attached to lung walls, where ambiguities for volume measurement occur. In the process of creating these data sets, we select a sample of often seen lung tumor shapes and locations in the lung, and show that for this sample a large fraction of the voxels representing tumors in the gridded data are partially filled voxels. This points out the need for volumetric methods that handle partial volumes accurately.

1 Introduction

The change in pulmonary nodules over time is an extremely important indicator of tumor malignancy and rate of growth. Physicians base both diagnoses and treatment on perceived changes in size, so accurate and precise measurement of such changes can have significant implications for the patient. Many articles have described a variety of techniques for calculating tumor volumes and/or the change in tumor size between two sets of data on the same tumor [1]-[6]. To compare volumetric techniques, we need reference data with a variety of known tumor volumes with different levels of background noise. Although phantom tumor data are currently available and published studies [7] [8] have compared volumetric methods on these phantom data sets, the phantom data settings are often not realistic, because the synthetic phantoms are placed in a synthetic background. A realistic and valid assessment of these volumetric methods needs realistic calibrated data sets. Lung tumor volume calculations are often difficult because of the physical location of tumors in the lung: typically highly vascular regions where tumors are hard to discriminate from overlaying blood vessels or along the pleural lining of the lung where boundaries are also hard to define. Ground truth sets of lung tumor data are needed to compare different volumetric measurement methods. Of particular importance are sets with a single

* This contribution of NIST, an agency of the U.S. government, is not subject to copyright.

tumor acquired over time. In a previous paper we presented a method for embedding spheres of known size into lung CT data to create reference synthetic data sets in realistic lung environments [9]. In this paper, we expand on these ideas to replicate eight sets of clinical data with a range of lung tumor shapes and locations. We explain why these sets were chosen, and then describe the methods to recreate clinical tumor shapes with synthetic tumors whose volumes are computed accurately as the shapes are created.

2 Examination of Lung Tumor Edges

The pixel intensities associated with a lung tumor appearing on a CT scan typically lie in the same intensity range as the blood vessels of the lung (between -100 and +200 Hounsfield units). The non-vascular lung tissue and air passageways, which serves as the background in these scans, appear at significantly lower intensities (-1000 to -700 Hounsfield unit range). The regions at or near the lung tumor surface lie in a broad range between the two (-700 to -100 Hounsfield units), and are only partially filled by a tumor or contain both tumor and other vascular material. Figure 1 shows a section of a CT slice through a tumor, along with a 3-D picture of an isosurface from 41 slices of the same data set that include the tumor. The isosurface is created at a Hounsfield value of -400, and includes both the tumor and the surrounding blood vessels. A histogram of the pixel intensities in a small region containing the tumor, also shown in Figure 1, is representative of diseased lung tissue. There is a peak for the background tissue, a peak for the tumor, and a spread of intensities in between the two that represents the partial volumes and vascular material surrounding the tumor.

3 Clinical Tumor Sets

Eight clinical tumors were selected with the help of a radiologist from a larger set of available data that included a range of sizes and physical locations in the lung. Images were obtained from the public NCI Reference Image Database to Evaluate Response (RIDER) [10]. Data from each of these patients are available at two different time points. We cannot precisely measure the volumes of these clinical tumors and their changes over time, but we can create synthetic models of known volume. Several of the selected tumors are small and centrally located in the lung. Others are larger and attached in some way to the lining of the lung. The larger tumors appear to be less round than the smaller ones, and require more complex geometric models. These eight tumors were selected because they had shapes and attachments representative of those seen in our data sets. Several of the tumors were selected because they represented lesions with commonly occurring challenges in terms of reading the data. If several sets of CT data included similar tumors, we chose a sample representative of that shape and location.

We are able to recreate clinical tumor shapes using combinations of spheres, ellipsoids, cones, and cylinders, which can be assembled at arbitrary sizes and

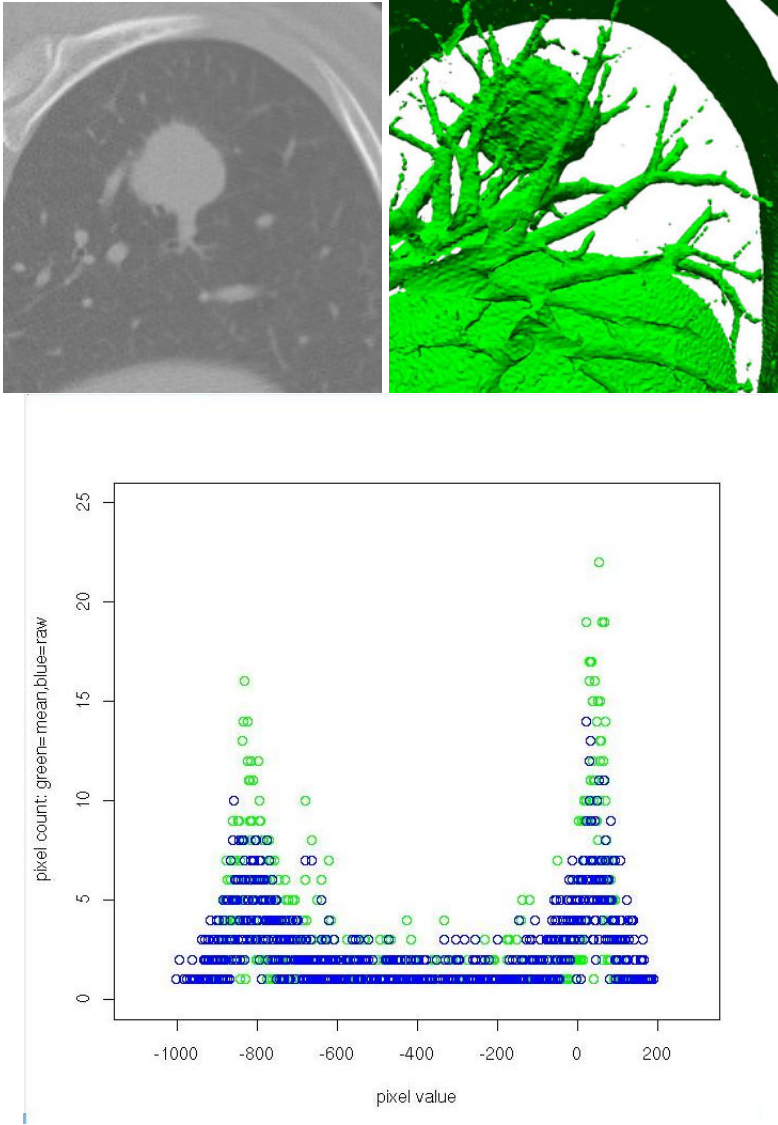


Fig. 1. Section of a slice of CT data containing a tumor; Isosurface at -400 Hounsfield units of 41 slices containing the same tumor; Histogram of pixel intensities in a small grid immediately surrounding the tumor

angles with respect to one another. During assembly, partial voxel volumes are calculated by subdividing each voxel and testing for the appearance of each geometric object in the subsections. Using this technique, we are able to get an accurate estimate of the final volume for the combined set of objects. A particular set of geometric objects is assembled in a grid whose grid points contain either a 1.0 for a completely filled voxel, a 0.0 for a completely empty voxel, or a number between 0.0 and 1.0 representing the fraction of the voxel filled by the geometric objects.

As described in a previous paper [9], pixel intensities are then selected for each grid point in the grid of geometric objects. We first approximate the pixel intensity distribution of a clinical tumor in the same data set, using a normal distribution and the mean and standard deviation of the clinical tumor's pixel distribution. Any grid point completely inside of the synthetic tumor is then assigned a random value from this normal distribution.

Pixel intensities for the partially filled voxels are computed based on parameters that we derive from the clinical tumor data, which define the upper and lower boundaries for pixel intensities at the tumor surface. We find the lower boundary for surface pixel intensity from the gradient field of the pixel intensities at the tumor surface. For each intensity at the tumor edge, we find an average value for the magnitude of pixel gradient. K_2 , the lower boundary for surface pixel intensity, is the intensity with the maximum average gradient magnitude at the tumor edge. The upper limit for edge pixels, K_1 , is determined from a ratio based on our previous work. Spheres of different sizes were embedded into clinical CT data such that the pixel intensity gradient at the sphere surface resembled the gradient of clinical tumor data. Pixel intensities at the edge contributed to the overall volumes of the spheres, but were weighted depending upon their intensity. All intensities greater

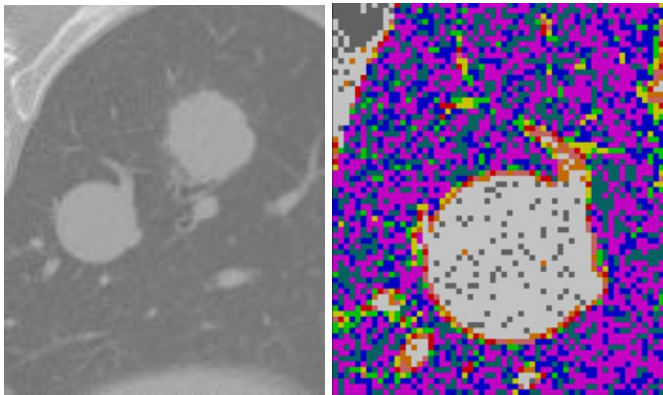


Fig. 2. One slice of a sphere inserted into clinical CT data, DICOM data on the left and pixels individually colored on the right, to differentiate the background, edge, and tumor intensities; pixel intensities in Hounsfield units: white:-150 to 100, orange:-250 to -150, pink:-350 to -250, red:-450 to -350, yellow:-550 to -450, green:-650 to -550, blue:-750 to -650, purple:-850 to -750, teal:less than -850

Table 1. Ranges of pixel intensities for partial volumes of synthetic tumors

Data set	K2	K1 (Hounsfield units)
1	-439	-204
2	-420	-194
3	-413	-190
4	-412	-189
5	-356	-167
6	-388	-182
7	-415	-188
8	-426	-197

than K_1 were assumed to be inside the sphere, and because the sphere volumes were known, values of K_1 were estimated. Over all of the data sets in our previous study, a common ratio emerged from our values of K_1 , K_2 , and the mean value of the peak of the tumor pixel intensities, K_m :

$$\frac{K_m - K_1}{K_m - K_2} = 0.46. \quad (1)$$

We therefore estimate K_1 in our current data sets once we know K_2 and K_m . Grid values with partial volumes are assigned a pixel intensity between K_2 and K_1 according to their partial volume. The new grid is then embedded into the CT data. Figure 2 shows an example of one slice of a sphere as both DICOM data and as colored individual pixels. Table 1 provides data for the range of pixel intensities for the eight synthetic tumor partial volumes, calculated from a clinical tumor found in each of the eight data sets. (See 9 for a lengthier discussion of how and why these parameters are chosen.)

4 Modeling Clinical Tumor Shapes

Figure 3 displays the eight geometric objects that were inserted into clinical data. These objects are shown in the figure as polygonal surfaces created from gridded data and represent a range of tumor sizes. Some were embedded in the wall of the lung to simulate more challenging cases; others were kept separate from the wall and are intended to be easier to measure. The clinical tumors that served as a basis for these shapes were selected with the help of a radiologist as representative samples from 26 sets available for this study. Each object in Figure 3 is made up of between four and 13 different geometric shapes chosen so that their cross sections in each CT slice resemble the cross sections of the clinical data. Figures 4-11 display sections of a single slice from each of the eight amended data sets at two different time points. The image data is taken from a single patient at two different times. In several of the figures, the clinical tumors that were used to model the synthetic tumors are seen. The volume change is specified for each set as either a 30 % increase in volume or a 10 % decrease in volume.

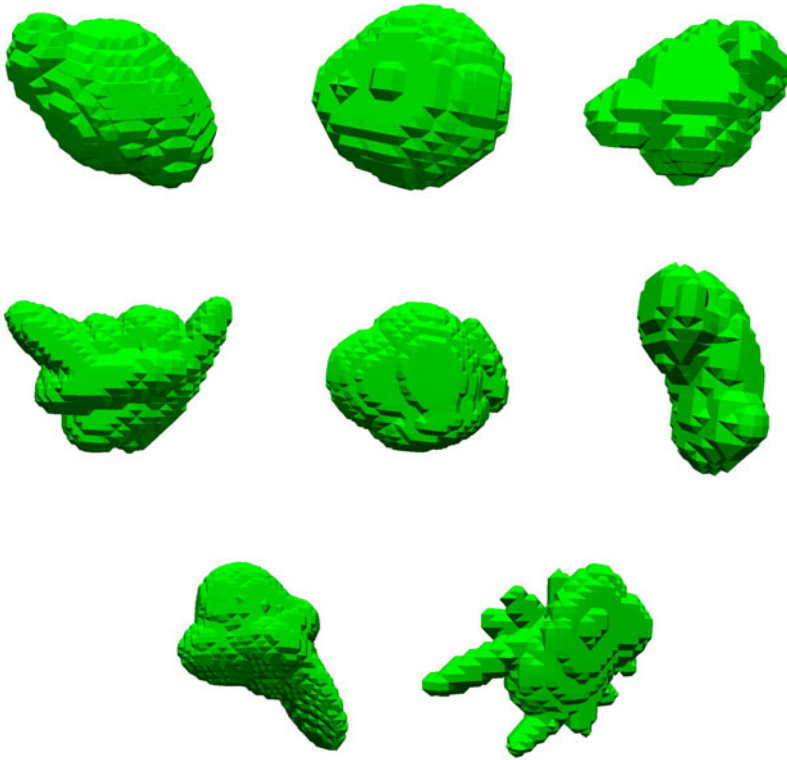


Fig. 3. The eight geometric objects that were inserted into clinical data at the initial time of the test

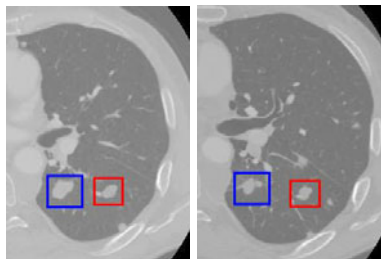


Fig. 4. Object 1 at two time points; the clinical tumors boxed in red, the synthetic tumors in blue, modelled after the tumor at time point 2. Tumor was reduced in size by 10 % for the second time point. The clinical tumor shrank by a larger amount during the corresponding time period.

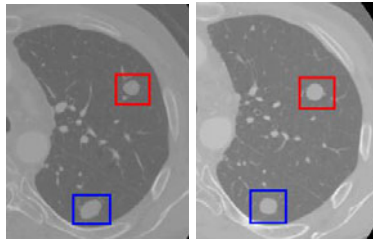


Fig. 5. Object 2 at two time points; the clinical tumors boxed in red, the synthetic tumors in blue, modelled after the tumor at time point 2. Tumor was increased in size by 30 % for the second time point.

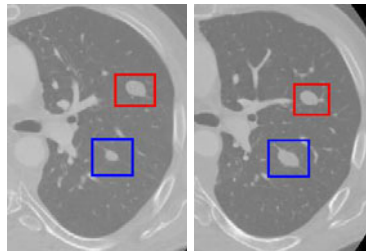


Fig. 6. Object 3 at two time points; the clinical tumors boxed in red, the synthetic tumors in blue, modelled after the tumor at time point 2. Tumor reduced in size by 10 % for the second time point.

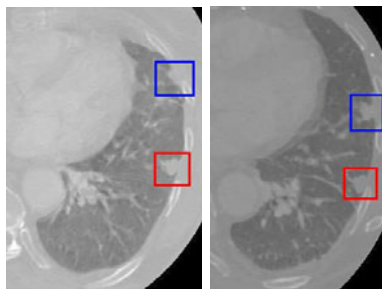


Fig. 7. Object 4 at two time points; the clinical tumors boxed in red, the synthetic tumors in blue, modelled after the tumor at time point 2. Tumor is increased in size by 30 % for the second time point.

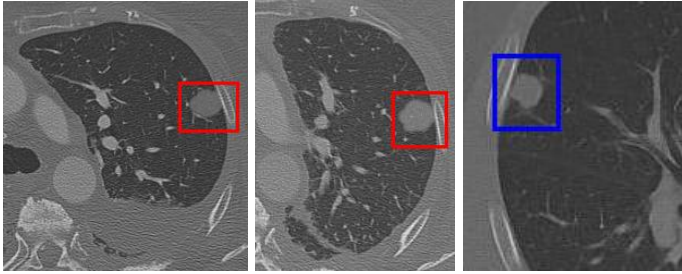


Fig. 8. Object 5 at two time points, boxed in red. The clinical tumor is not in these images. Tumor is increased in size by 30 % for the second time point. The third image is a section of a slice of the CT data containing the clinical tumor from which the synthetic tumor was derived, boxed in blue.

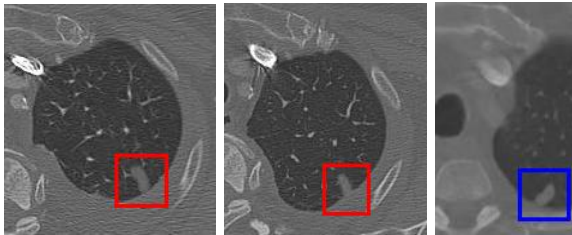


Fig. 9. Object 6 at two time points, boxed in red. The clinical tumor is not in this slice. Tumor was reduced in size by 10 % for the second time point. The third image is a section of a slice of the data containing the clinical tumor, boxed in blue, from which the synthetic tumor was derived.

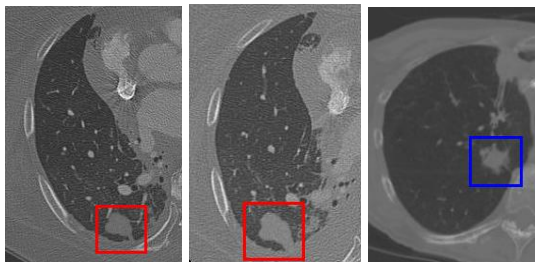


Fig. 10. Object 7 at two time points, boxed in red. The clinical tumor is not in this slice. Tumor was increased in size by 30 % for the second time point. The third image is a section of a slice of the data containing the the clinical tumor, boxed in blue, from which the synthetic tumor was derived.

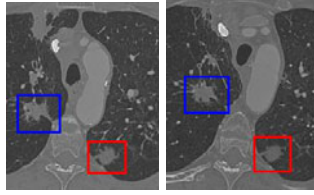


Fig. 11. Object 8 at two time points; the clinical tumors boxed in red, the synthetic tumors in blue. Tumor was reduced in size by 10 % for the second time point.

Changes in tumor size over time have traditionally been measured using the Response Evaluation Criteria in Solid Tumors (RECIST). One recent study [11] with phantom tumors has shown that size measurements are dramatically improved when using volumetric measurements instead of one-dimensional RECIST measurements. However, for small tumors, volume measurements often are dependent upon determining exact surface locations and are heavily influenced by the difficulties of measuring partial volumes. This is illustrated in Table 2, which shows the tumor volumes associated with the eight tumor shapes described here along with the fraction of partially filled voxels for each grid. These partially filled voxels are at risk for error in volume calculations. The fractions vary between 0.247 and 0.410; i.e., in our sample, approximately one quarter to one half of the voxels representing tumors are partially filled. Volumetric methods that do not accurately account for partial volume data will lead to errors of this extent. The fractions are high, either because the tumors are small and the surface area to volume ratios are high, or because the tumors, although larger, are less round, with more surface area in extended appendages. This underscores the need for accurate methods to determine tumor boundaries in CT data. We have suggested such a method in [12], which assigns weighted volumes to voxels that are not completely filled. Further work investigating changes in tumor shapes with time are needed to determine if the data sets used here are truly representative.

Table 2. Volumes and partial volumes for the eight objects

Data set	volume (mm^3)	fraction of partial volume voxels
1	27266.40	0.304
2	20800.67	0.323
3	10094.76	0.384
4	91280.65	0.336
5	18832.44	0.299
6	6136.94	0.410
7	57754.50	0.247
8	15158.10	0.377

5 Conclusions

Reference CT data sets that contain time-dependent lung tumor size change data are needed in order to compare volumetric methods for measuring lung tumor growth. Phantom lung tumor data can supply time dependent information, but not necessarily in the clinical environment of common lung tumors. These data also often have the disadvantage of not replicating the complications associated with tumor size measurements, such as highly concentrated vasculature or air passageways, or growth out of the pleural lining. Creating synthetic data in realistic environments is one way of mitigating these disadvantages. We have shown a selection of clinical tumors from available data that we believe exhibits representative shapes and placements. We describe the insertion of these synthetic shapes into clinical CT data. These representative synthetic tumors can then be used as ground truth data for comparison of algorithms that calculate tumor volumes. We show from an analysis of the gridded data representing all eight of our tumor shapes that a large fraction of voxels are only partially filled. Future work is needed to further investigate tumor shapes and their changes over time to see if our small study is representative of lung tumor data in general.

References

1. Kostis, W.J., Reeves, A.P., Yankelevitz, D.F., Henschke, C.I.: Three-Dimensional Segmentation and Growth-Rate Estimation of Small Pulmonary Nodules in Helical CT Images. *IEEE Trans. on Medical Imaging* 22(10) (October 2003)
2. Reeves, A.P., Chan, A.B., Yankelevitz, D.F., Henschke, C.I., Kressler, B., Kostis, W.J.: On measuring the change in size of pulmonary nodules. *IEEE Trans. Med. Imaging* 25(4), 435–450 (2006)
3. Mendonca, P., Bhotika, R., Sirohey, S., Turner, W., Miller, J., Avila, R.S.: Model-based Analysis of Local Shape for Lesion Detection in CT Lung Images. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3749, pp. 688–695. Springer, Heidelberg (2005)
4. McCulloch, C.C., Kaucic, R.A., Mendonca, P.R., Walter, D.J., Avila, R.S.: Model-based Detection of Lung Nodules in Computed Tomography Exams. *Academic Radiology* (March 2004)
5. Preim, B., Bartz, D.: *Image Analysis for Medical Visualization*. *Visualization in Medicine*, 83–131 (2007)
6. Preim, B., Bartz, D.: *Exploration of Dynamic Medical Volume Data*. *Visualization in Medicine*, 83–131 (2007)
7. Das, M., Ley-Zaporozhan, J., Gietema, H.A., Czech, A., Nuhlenbruch, G., Mahnken, A.H., Katoh, M., Bakai, A., Salganicoff, M., Diederich, S., Prokop, M., Kauczor, H., Gunther, R.W., Wildberger, J.E.: Accuracy of automated volumetry of pulmonary nodules across different multislice CT scanners. *Eur. Radiol.* 17, 1979–1984 (2007)
8. Ko, J.P., Rusinek, H., Jacobs, E.L., Babb, J.S., Betke, M., McGuinness, G., Naidich, D.P.: Small Pulmonary Nodules: Volume Measurement at Chest CT-Phantom Study. *Radiology* 228, 864–870 (2003)

9. Peskin, A.P., Kafadar, K., Dima, A., Bernal, J., Gilsinn, D.: Synthetic Lung Tumor Data Sets for Comparison of Volumetric Algorithms. In: The 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition, pp. 43–47 (July 2009)
10. NCI Reference Image Database to Evaluate Response (RIDER) database, <https://wiki.nci.nih.gov/display/CIP/RIDER>
11. Levine, Z.H., Borchardt, B.R., Brandenburg, N.J., Clark, C.W., Muralikrishnan, B., Shakarji, C.M., Chen, J.J., Siegel, E.L.: RECIST versus volume measurement in medical CT using ellipsoids of known size. *Optics Express* 18(8), 8151–8159 (2010)
12. Peskin, A.P., Kafadar, K., Santos, A.M., Haemer, G.G.: Robust Volume Calculations of Tumors of Various Sizes. In: The 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition (July 2009)

Spectral Image Decolorization

Ye Zhao^{*,**} and Zakiya Tamimi

Kent State University, Ohio 44242

Abstract. We propose a novel method for decolorizing a color image to grayscale, which preserves the chromatic contrast to be lost if using a simple luminance mapping. Unlike previous work, the conversion performs in spectral domain after applying Fourier transforms on luminance and chromatic channels, and the resultant intensity is recovered from an inverse transform. Frequency spectra inherently provide the magnitude of changes (i.e. contrast) among the channels, in all spatial scales. Consequently, only simple arithmetic operations are needed to add chromatic differences to the final grayscale intensity, leading to realtime computational speed. Moreover, users can flexibly control the decolorization effects by interactively adjusting visually-intuitive parameters, in addition to our predefined values. The method has great potential to be incorporated in commercial software and printer drivers due to its simplicity, speed and repeatability.

1 Introduction

We are living in an era that people acquire, generate, manipulate and display innumerable colorful digital images (or documents) with cameras, scanners and computers, while meanwhile, black-white TVs are very hard to find, and black-white pictures are only presented by professional photographers. However, ironically, image decolorization that degrades a color image to a grayscale image, is of increasing importance and interest for researchers and consumers. First, grayscale prints are budget-friendly as well as environmentally friendly, in comparison with the color ones. Therefore, they are widely used in newspapers, journals and daily handouts. Second, many image manipulation techniques, e.g. in pattern recognition, rely on the grayscale mapping of a colorful input. Effective conversion technique that keeps original color information as much as possible, and meanwhile, provides results as soon as possible, will have wide use in many applications.

Color to grayscale conversion inevitably leads to the loss of perceptual information. This procedure is typically implemented by discarding chromatic information while retaining the luminance channel. However, two perceptually contrasted colors may become one similar luminance value, with chromatic difference disappearing in results. The approach may end up with inferior grayscale prints of no use, or risk consequent image analysis by losing significant identification patterns (e.g. edges). A handful of pursuits have been conducted to enhance the decolorization results in order to preserve the original color (both luminance and chrominance) differences. Most previous approaches regard the grayscale conversion as an optimization problem, trying to find

* Corresponding author.

** See color images at <http://www.cs.kent.edu/~zhao/SpectralDecolor.htm>

an optimal color-to-gray mapping that preserves the contrast in fixed-size (or global) spatial neighborhoods, or among the colors appearing in a particular image. Solutions to the optimization problem suffers slow numerical computing, typically consuming many seconds or minutes on a medium size image. Random sampling of contrast in paired pixels, combined with a predominant component analysis, can provide fast computation and enhance contrasts within varying neighborhood sizes. Nevertheless, its random contrast sampling makes the conversion results not repeatable, unless a fixed random seed is stored and reused, which restricts its applicability in many scenarios. In this paper, we propose a novel decolorization algorithm which provides controllable contrast preservation in all spatial scales, and furthermore, achieves excellent computational performance enabling realtime feedbacks for flexible user control and adjustment.

Due to the perceptual characteristics of human visual system, colors are represented by variable combinations of three component channels. Many color spaces with different components are used, for example, the RGB space is popular in digital displays. Luminance measures the energy an observer perceives, which typically forms a grayscale image. To pick up the chromatic difference, compared with the RGB space, the perceptually designed CIE *Lab* space provides a good operational basis, since the L channel directly links with luminance, and the a and b channels accommodate chrominance. We use the *Lab* space throughout this paper.

Our method seeks to implement the color-to-gray mapping in spectral domain, in particular using the Fourier space in this paper. Other spectral techniques, such as the wavelet domain, can also be applied, since our contrast enhancement strategy is not affected. We first compute Fourier transforms on the luminance channel (L), and the two chromatic channels (a and b) of an image, respectively. The results, \hat{L} , \hat{a} and \hat{b} , are directly related to spatial rates of the intensity change at all spatial scales. Each frequency spectrum (i.e. magnitude) inherently reflects the contrast level at each corresponding scale, among the three components of the image. A luminance-mapping grayscale image can be recovered from an inverse Fourier transform of \hat{L} . Our method implements the augmented decolorization by modifying \hat{L} incorporating compensation from \hat{a} and \hat{b} . Thereafter the inverse transform achieves a desired grayscale image preserving visual differences of luminance and chrominance. We provide a predefined scheme to compute two coefficients used in adding chromatic contrast to luminance contrast: one for defining the degree of chromatic contrast to be incorporated, and another one for determining different levels of chromatic contrast from two different channels, respectively. Besides the automatic computation, the coefficients can also be flexibly defined by users for various tasks.

The method only involves simple arithmetic operations, other than several Fourier transforms. The transforms are implemented theoretically with $O(N \log N)$ complexity by a Fast Fourier Transform (FFT) algorithm, with the number of pixels N . In practice, our method efficiently accomplishes the decolorization with realtime feedback for most images on typical consumer computers. In summary, we innovate a spectral decolorization method that effectively converts color images to grayscale ones, which is realtime, repeatable, easy-programming and with flexible control. Due to these merits, we are looking forward to it being used in image processing software and commercial printer drivers.

2 Related Work

Several methods have achieved success in decolorization. Bala et al. [1] presented a method to convert color images to grayscale while preserving color edges. Chromatic edges are gained by applying a spatial high-pass filter and combined with weighted luminance values. The method successfully preserves the chrominance edges while other chromatic contrasts are not well captured. Rasche et al. [2] proposed to solve the color to grayscale conversion and recoloring as a constrained, multivariate optimization. They defined a quadratic objective function incorporating contrast preservation among all colors present in an image, ignoring the spatial distribution of the colors. Then a sequence of linear programming problems were solved to find desired optimum of the objective function. However, the solutions were slow to converge and prone to local minima. Instead of optimization in the colors, Gooch et al. [3] sought to tackle the problem between neighboring pixels. They created target differences based on local luminance and chrominance contrasts between pixels, while the neighborhood size was controlled by a user-specified parameter. Optimizing the target difference in grayscale results led to a very slow $O(S^4)$ algorithm for a full neighborhood (i.e. whole image) computation, where S was the resolution scale of the image. To overcome the slow computation of the optimization techniques, and to capture contrasts in varying-size neighborhoods, random sampling of pixel pairs was explored with a predominant component analysis, to provide fast and direct conversion results [4]. Simply pairing pixels with a Gaussian distribution, as a statistical sampling method, may affect the effectiveness of the contrast acquisition in some cases. Random sampling also required additional efforts, such as keeping a fixed seed as suggested by the authors, to enable repeatable conversion results, which restrained its usage in some tasks. Neumann et al. [5] also presented a method focusing on perceptual based approach by measuring the color and luminance contrasts as a gradient contrast of the Coloroid space, whose running time is linear and no user intervention is needed. It requires comparison and/or optimization over many variables. Recently, Kim et al. [6] proposed a fast algorithm preserving feature discriminability and color ordering by optimization of a nonlinear global mapping. They further applied the method to video conversion.

In comparison, our method tackles the problem through an alternative route while a very simple algorithm is proposed without complex optimization involved. Diverting from the pixel space of an image, we perform contrast preservation based on a transformed image in the frequency space. The transformation inherently computes the spatial changes (i.e. contrasts) of the color components, providing the luminance and chromatic differences within different feature scales. Consequently, our method is able to augment contrasts in all scales simultaneously, and completes in a very fast speed based on the fast Fourier transform on both consumer CPU and GPU. Besides suggested mapping results with default parameters, we provide full control for users who can easily adjust parameters for immediate feedback of tentative grayscale images. One or even many satisfied final results are chosen by users instead of a predetermined optimization result, which in many cases are not appropriate or plausible due to various application scenarios and user perceptual discretion.

3 Spectral Decolorization

3.1 Contrast Enhancement

Performing the Fourier transform on each channel (L,a,b) of a color image I , we acquire three spectral images: \hat{L} , \hat{a} and \hat{b} , with complex values. A conventional grayscale image, G_I , can be achieved through an inverse transform of \hat{L} : $G_I = \text{IF}(\hat{L})$, where $\text{IF}()$ is the inverse Fourier transform. We introduce the chromatic contrast into an enhanced grayscale image, \tilde{G}_I , in the Fourier domain. \tilde{G}_I is computed by the inverse Fourier transform as

$$\begin{aligned}\tilde{G}_I &= \text{IF}(\hat{E}), \\ \hat{E} &= \text{H}(\hat{L}, \hat{a}, \hat{b}).\end{aligned}\quad (1)$$

Here, a function H computes a modified Fourier-domain grayscale intensity, \hat{E} , from the Fourier-domain counterparts of the original luminance channel and the two chromatic channels. H is implemented at each frequency as

$$\text{H}(\hat{L}, \hat{a}, \hat{b}) = (1 - \theta)\hat{L} + \theta(\phi\hat{a} + (1 - \phi)\hat{b}), \quad (2)$$

where θ controls the degree of chromatic contrast added to the grayscale result, and ϕ is a coefficient to determine the relative contributions of the a and b channels. In Eqn. 2, all the Fourier values and coefficients are dependent on the frequency \mathbf{w} , which is omitted for clearance.

3.2 Parameter Control

Two controllable coefficients θ and ϕ determine various contrast augmentation effects on grayscale results. They can be computed automatically based on the data fact of the Fourier spectra. θ models the degree of the incorporated chromatic contrast, which can be determined by linking it with the relative conversion loss measured by comparing the RGB difference and the luminance difference. In our scheme, these differences are modeled by the spectrum operation at each frequency:

$$\theta = \frac{|\hat{R}| + |\hat{G}| + |\hat{B}| - |\hat{L}|}{|\hat{R}| + |\hat{G}| + |\hat{B}|}. \quad (3)$$

Here $||$ represents the spectrum of the complex values, $\hat{R}, \hat{G}, \hat{B}$ are Fourier transform results of R, G, B channels. Meanwhile, ϕ is computed by the relative proportion of the P and Q spectrum:

$$\phi = \frac{|\hat{a}|}{(|\hat{a}| + |\hat{b}|)}. \quad (4)$$

θ and ϕ can be automatically computed at each \mathbf{w} and applied in Eqn. 2. In practice, using an averaged θ and an averaged ϕ from all frequencies can generally provide clear results with no artifacts caused by individual frequency operations. In this way, the results satisfy global consistency, i.e. pixels with same color mapping to same grayscale, due to the linearity of Fourier transform applied to Eqn. 2.

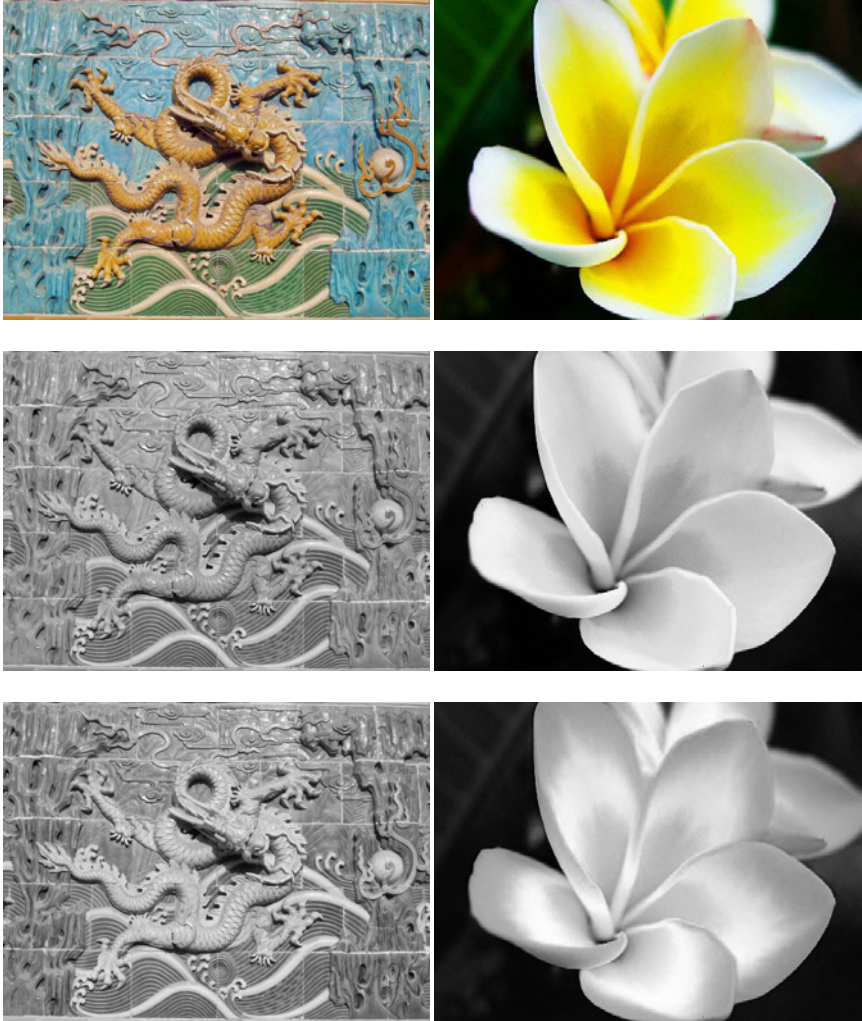


Fig. 1. Grayscale mapping of color images: (top) Color images; (middle) Luminance images; (bottom) Spectral decolorization results

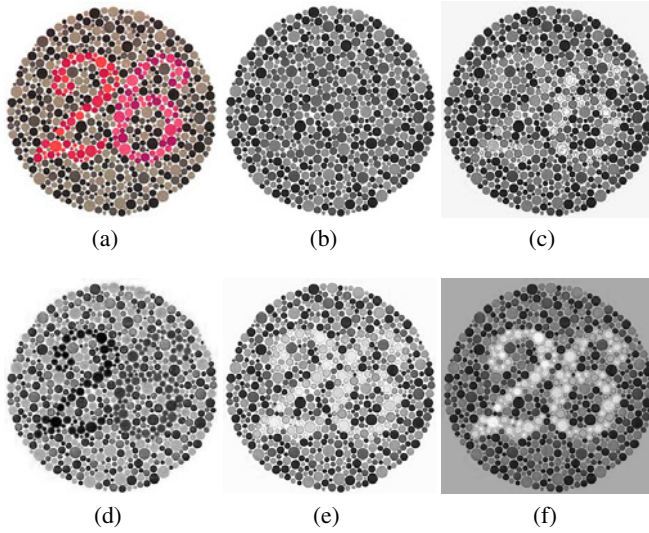


Fig. 2. Grayscale mapping of a colorblind test image: (a) Color image; (b) Luminance; (c) Gooch et al. [3]; (d) Grundland et al. [4]; (e) Spectral decolorization with an automatically-computed $\theta = 0.44$; (f) Spectral decolorization with a larger $\theta = 0.6$.

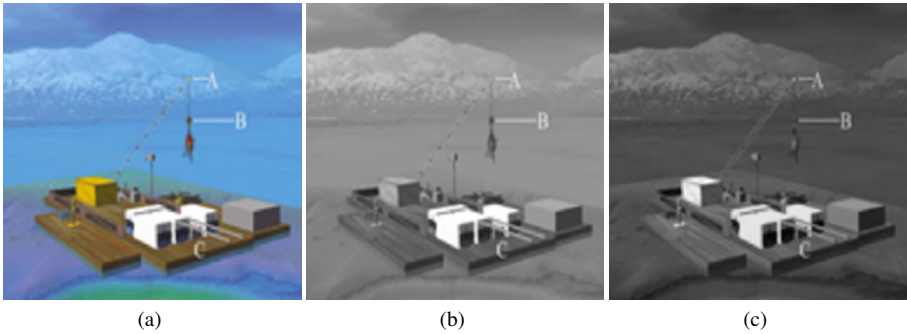


Fig. 3. Grayscale mapping of a color image with lettered labels: (a) Color image; (b) Luminance; (c) Spectral decolorization with legible letters. Color image from [3]

3.3 Discussion

More complex optimization on all frequencies can also be applied to choose θ and ϕ , however, such totally-automatic choice is not appropriate for various purposes from different users and applications. It is obvious there exist no perceptually best mapping that can satisfy everybody. A practical strategy is to provide flexibility and control to field users. We prefer to give users a handful interface using the averaging values as defaults. If the automatically-selected parameters do not produce desired results, our system enable interactive adjustment with immediate responses, thanks to the fast computation. In

comparison to pixel domain methods using fixed local regions and partial contrast scales, our method provides a full spectrum of color contrast information in all spatial scales, which leads to a new framework and thus makes it possible for better configuration of grayscale mapping. Future study will focus on providing color-to-gray mapping with interactive control of quality on different frequencies: (1) mapping that bases on user examples, e.g., from existing good mapping; (2) mapping with importance, e.g., guided by user's selection of critical regions on the whole image, in which the converted result should have best quality; etc. We also anticipate this novel frequency-based approach being combined with other methods to help solving this critical problem in a better way.

4 Results

Due to the very simple arithmetic operations of our algorithm, most computing time is consumed by the FFT operations implemented by the `fftw` package [7]. The algorithm is also implemented in Matlab with a few lines of codes. Our method runs very fast: for a 512×512 color image, it uses about 0.25 seconds on a consumer PC with an Intel Core2 Quad 2.4GHz CPU and 3GB memory. Thanks to the FFT acceleration on GPU,

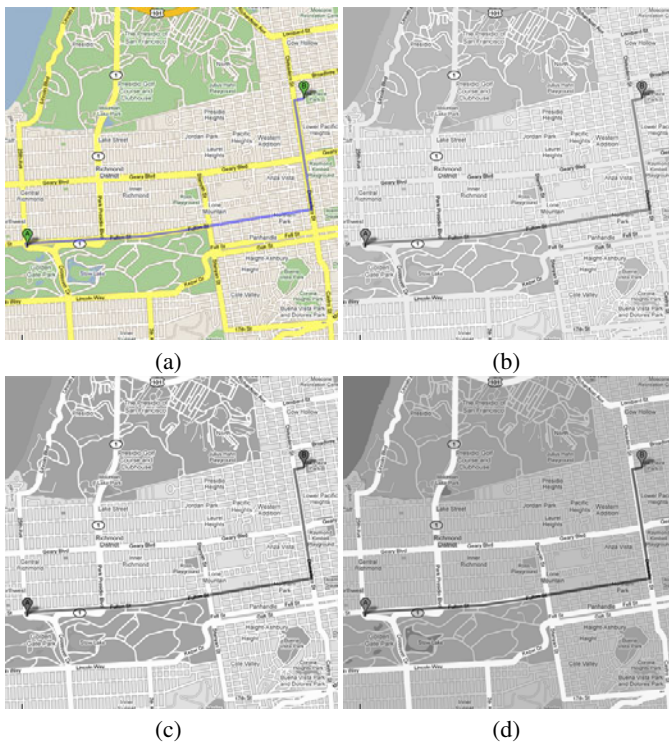


Fig. 4. Grayscale mapping of a route map: (a) Color image; (b) Luminance; (c) Grundland et al. [2007]; (d) Spectral decolorization with a pre-computed $\theta = 0.47$. Copyright of the map belongs to Google Inc.

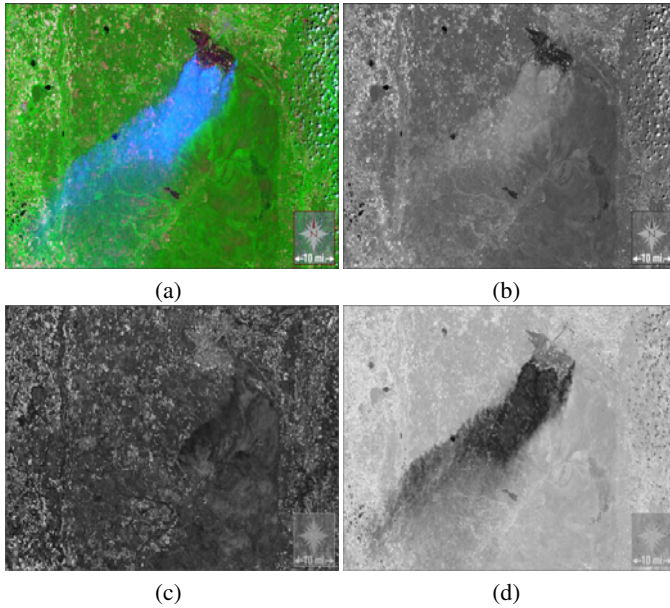


Fig. 5. Grayscale mapping of a satellite landscape image of record-breaking fires by a severe drought in Georgia 2007: (a) Color image; (b) Luminance; (c) Spectral decolorization with $\theta = 0.6$, $\phi = 0.1$; (d) Spectral decolorization with $\theta = 0.6$, $\phi = 0.9$. Color image courtesy of NASA.

it improves to around 0.02 seconds on an nVidia 8800 GT 512MB. For an 800×600 image, it uses 0.29 seconds on the CPU and 0.04 seconds on the GPU.

Fig. 1 shows the conversion results compared with their classic luminance images. A yellow dragon is visually separable in grayscale from the background on the top row. At the bottom, the yellow areas become divisible on the white petals after conversion.

In Fig. 2, we convert a colorblind test image to grayscale. Fig. 2c is the result of Gooch et al. [3], using full image optimization with a large chromatic variation value ($\alpha = 30$), which shows mildly enhanced contrast. In Fig. 2d, the algorithm of Grundland et al. [4] generates an augmented image with a very large degree of enhancement ($\lambda = 2.0$). Fig. 2e uses our spectral decolorization with a pre-computed $\theta = 0.44$ from Eqn. 3. Fig. 2f uses a larger $\theta = 0.6$, adding more chromatic contrast to the result.

In Fig. 3, we apply our algorithm to a color image having been used in by Gooch et al. [3] and Grundland et al. [4]. In contrast to the previous methods, in which “the lettered labels typically are made less legible” [4], Fig. 3c depicts that our algorithm, performing on all scales, can preserve background details and also keep the labels clear.

Decolorization is probably more widely used in daily documents, for budget-saving printing, than in scenery pictures. Fig. 4 illustrates our method in mapping a digital route map to grayscale. Fig. 4c shows a good result by Grundland et al. [4] with $\lambda = 0.5$. Our spectral decolorization result in Fig. 4d shows further enhanced information of the route, together with the enhanced major roads (yellow in Fig. 4a) which play a significant reference role in route map understanding and usage.

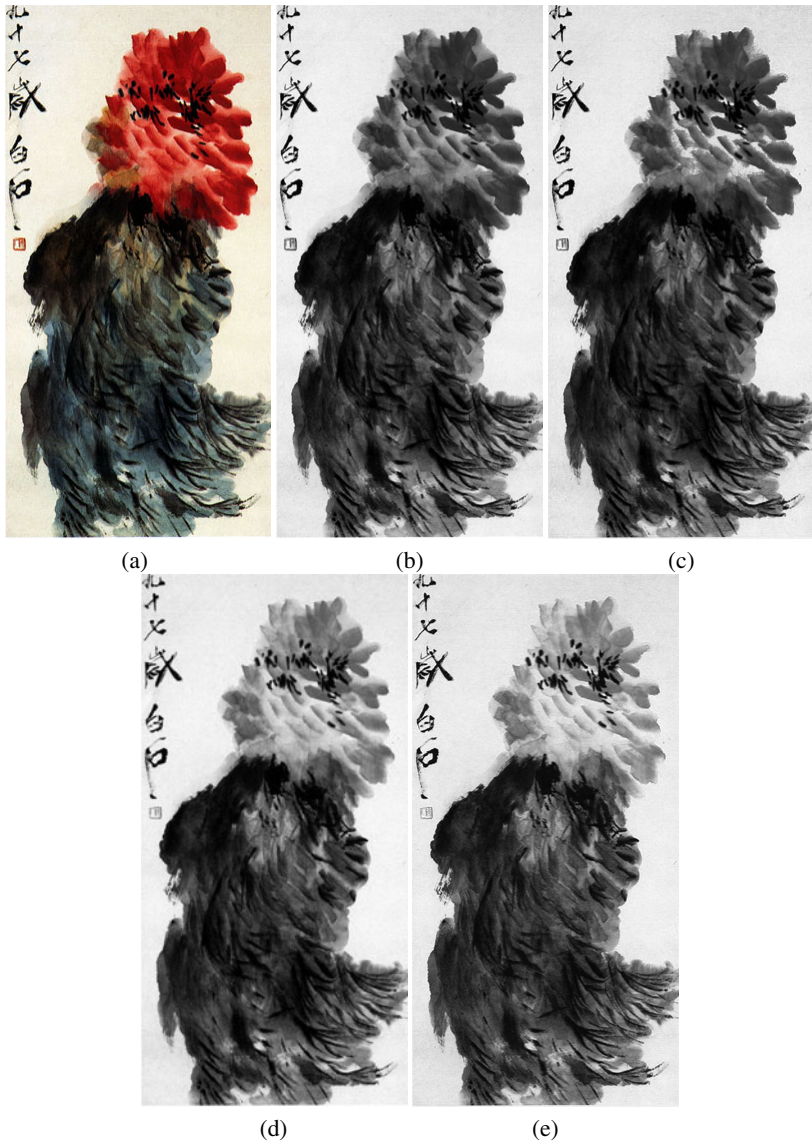


Fig. 6. Grayscale mapping of a Chinese painting: (a) Color image; (b) Luminance; (c) Gooch et al. [2005]; (d) Grundland et al. [2007]; (e) Spectral decolorization. Color image courtesy of a famous Chinese artist Qi Baishi.

In Fig. 5 we show the effects of different ϕ values of mapping a satellite landscape image with fires by a severe drought. Two extreme cases are used: Fig. 5c augments many red spots not obvious in luminance with $\phi = 0.1$, and Fig. 5d enhances blue regions of the smoke, with $\phi = 0.9$. Fig. 6 converts a Chinese painting using different

methods. Our result in Fig. 6e presents clear contrast between the red and black regions of the origin image.

5 Conclusion

A spectral image decolorization method performs grayscale conversion in Fourier domain, which lends itself to a good tool in many applications, due to its simple implementation, fast speed, and repeatable results. In the future, we will further extend the method to recoloring images for helping color-deficiency people.

References

1. Bala, R., Eschbach, R.: Spatial color-to-grayscale transform preserving chrominance edge information. In: Proc. Color Imaging Conference, pp. 82–86 (2004)
2. Rasche, K., Geist, R., Westall, J.: Re-coloring images for gamuts of lower dimension. Computer Graphics Forum 24, 423–432 (2005)
3. Gooch, A.A., Olsen, S.C., Tumblin, J., Gooch, B.: Color2gray: salience-preserving color removal. In: SIGGRAPH 2005: ACM SIGGRAPH 2005 Papers, pp. 634–639. ACM, New York (2005)
4. Grundland, M., Dodgson, N.A.: Decolorize: Fast, contrast enhancing, color to grayscale conversion. Pattern Recogn. 40, 2891–2896 (2007)
5. Neumann, L., Cadík, M., Nemcsics, A.: An efficient perception-based adaptive color to gray transformation. In: Proc. Computational Aesthetics, pp. 73–80 (2007)
6. Kim, Y., Jang, C., Demouth, J., Lee, S.: Robust color-to-gray via nonlinear global mapping. In: SIGGRAPH Asia, pp. 1–4. ACM, New York (2009)
7. Frigo, M., Johnson, S.G.: The design and implementation of FFTW3. Proc. IEEE 93, 216–231 (2005)

Author Index

- Abdelrahman, Mostafa III-9
Alba, Alfonso I-35, I-417
Alger, Jeffrey R. II-292
Al-Hamadi, Ayoub I-153, I-253, II-574
Al-Huseiny, Muayed S. II-252
Ali, Asem III-79
Alvarez, Damián III-171
Ambrosch, Kristian I-437
Amresh, Ashish I-54
Antani, Sameer K. III-261
Arandjelović, Ognjen III-89
Arbab-Zavar, Banafshe III-625
Arceneaux IV, George II-564
Arce-Santana, Edgar I-35, I-417
Argyros, A.A. II-584
Argyros, Antonis A. I-405
Artinger, Eva II-429
Asari, Vijayan K. III-49, III-474
Assoum, A. I-276
Atsumi, Masayasu II-696
Azary, Sherif II-606
- Badakhshannoory, Hossein II-342
Bai, Li I-371, II-637
Balázs, Péter III-339
Bales, M. Ryan I-211
Barman, Sarah II-669
Bärz, Jakob I-582
Basile, Teresa M.A. I-571
Baumann, Florian I-286
Bebis, George III-161
Beichel, Reinhard II-312
Belhadj, Farès III-524
Berger, Marie-Odile I-231
Beyer, Ross I-688
Bhooshan, Sunil III-458
Bi, Chongke I-328
Bloch, Isabelle I-393
Boggess, Erin I-350
Boggus, Matt II-213, II-501
Borghì, A. II-678
Borst, Christoph W. I-719, I-729
Bouchrika, Imed III-625
Boyer, Vincent III-504, III-524
- Branch, John William I-602
Bresson, Xavier II-97
Brillet, Pierre-Yves II-302
Brimkov, Valentin E. I-592
Bringay, Sandra III-534
Bukhari, Faisal II-11
Burch, Michael I-338, III-447
Buthpitiya, Senaka III-1
- Caldairou, Benoît I-13
Camargo, Aldo I-698
Campos-Delgado, Daniel U. I-35
Cao, Yanpeng I-654
Caponetti, Laura I-571
Caracciolo, Valentina II-491
Carter, John N. III-625
Casanova, Manuel F. III-9
Caunce, Angela I-132
Cavalcanti, Pablo G. I-190
Cernuschi-Frías, Bruno III-271
Cerón, Alexander III-349
Chalfoun, Joe I-23, I-549
Chambon, Sylvie II-182
Chang, Remco II-564
Chan, T.F. II-678
Cheikh, Faouzi Alaya II-491
Chelberg, David III-417
Chen, Bertha III-604
Chen, Dongqing III-9
Chen, Huiyan III-368, III-407
Chen, Qian I-449, III-229
Chen, Runen II-687
Chen, Wei I-427
Chen, Xi I-612
Cheng, Heng-Tze III-1
Cheng, Irene II-406
Chesi, G. III-109
Chiang, Pei-Ying I-108
Cho, Siu-Yeung II-129
Choe, Yoonsuck II-322
Choi, Inho III-199
Chung, Ronald III-280
Coffey, Dane II-351
Constantinou, Christos E. III-604

- Cootes, Tim I-132
 Cope, James S. II-669
 Cordes, Kai I-264
 Corsi, Christopher I-74
 Cottrell, Garrison W. I-199
 Couture-Veschambre, Ch. II-416
 Crawfis, Roger II-213, II-501, II-511
 Crawford, P. II-533
 Cretu, Ana-Maria II-232
 Crivelli, Tomás III-271
 Crouzil, Alain II-182
 Cummings, Alastair H. II-332
- Dailey, Matthew N. II-11
 Daněk, Ondřej III-387
 Darbon, J. II-678
 Das, Dipankar II-439
 Das, Kaushik I-719
 Davis, James W. I-120, I-381, III-613
 de Santos Sierra, Alberto I-479
 Denzler, Joachim II-459
 Desquesnes, Xavier II-647
 Dhome, Michel III-219
 Dickerson, Julie A. I-350
 Diem, Markus III-29
 Dillard, Scott E. II-64
 Dima, Alden A. I-23, I-549, II-736
 Do, Phuong T. III-484
 Dodds, Z. III-151
 D'Orazio, Tiziana III-291
 Dornaika, F. I-276
 Du, Shengzhi III-320
 Dubois, Eric III-189
 Dubuisson, Séverine I-393
 Duschl, Markus II-429
- Ehlers, Arne I-286
 Eichmann, David III-139
 Eikel, Benjamin I-622
 Elhabian, Shireen III-9, III-79
 Elias, Rimón II-161
 Elliott, John T. I-23, I-549
 Elmoataz, Abderrahim I-539, II-647
 English, C. II-53
 Ernst, Katharina I-286
 Esposito, Floriana I-571
- Fabian, Tomas II-716
 Fabián, Tomáš III-310
 Fairhurst, M.C. I-461
- Falk, Robert III-79
 Fanello, Sean R.F. II-616
 Farag, Ahmed III-9
 Farag, Aly III-9, III-79
 Farag, Amal III-79
 Fechter, Todd A. II-394
 Feijóo, Raúl I-529
 Fellner, Dieter W. III-514
 Feltell, David I-371
 Fetita, Catalin II-302
 Fierrez, J. I-461, I-489
 Fijany, Amir II-469
 Filliben, James J. I-23
 Finnegan, David I-666
 Fiorio, Christophe II-85
 Fischer, Matthias I-622
 Flores, Arturo I-199
 Fong, Terry I-688
 Fontaine, Jean-Guy II-469
 Forsthoefel, Dana I-211
 Förstner, Wolfgang I-654
 Fowers, Spencer G. III-368, III-407
 Foytik, Jacob III-49
 Fujishiro, Issei I-328
 Fujiwara, Takanori I-306
 Fünfzig, Christoph I-54
- Gales, Guillaume II-182
 Gao, Zhiyun III-129
 García-Casarrubios Muñoz, Ángel I-479
 García, Hernán III-171
 Garz, Angelika III-29
 Gaura, Jan III-310
 Geiger, Cathleen I-666
 Geismann, Philip I-243
 Geist, Robert I-74
 Giménez, Alfredo II-554
 Gomez, Steven R. II-373
 Gong, Jianwei III-407
 Gong, Minglun II-481
 Gori, Ilaria II-616
 Graham, James III-9, III-79
 Grammenos, D. II-584
 Grand-brochier, Manuel III-219
 Grazzini, Jacopo II-302
 Grenier, Philippe II-302
 Grout, Randall III-129
 Grover, Shane II-361
 Gschwandtner, Michael III-19

- Gu, Yi III-437
 Gueorguieva, S. II-416
 Guerra Casanova, Javier I-479
 Guo, Yu I-96
 Gupta, Raj Kumar II-129
 Gutierrez, Marco I-529
- Hamann, Bernd II-554
 Hanbury, Allan II-75
 Hansen, Tina I-582
 Hantos, Norbert III-339
 Hao, Qing II-292, III-359
 Hardeberg, Jon Y. I-361
 Hatsuda, Hiroshi III-594
 He, Qiang I-698
 He, Zifen III-377
 Hempe, Nico II-202
 Hishida, Hiroyuki III-39
 Hlawitschka, Mario II-554
 Hödlmoser, Michael II-1
 Hoerber, Orland II-481
 Hoffman, Eric III-129
 Hoffmann, Kenneth R. III-359
 Hoi, Yiemeng III-359
 Holtze, Colin III-129
 Hoque, Enamul II-481
 Horiuchi, Takahiko I-181
 Hosseini, Fouzhan II-469
 House, Donald II-192
 Hu, Xiao II-292, III-359
 Huang, Rui II-139
 Hung, Y.S. II-21, III-109
 Hussain, Muhammad I-64
- Ibrahim, Mina I.S. I-499
 Ignakov, D. II-53
 Ikeda, Osamu I-678
 Imiya, Atsushi I-561
 Iwamaru, Masaki I-306
- Jähn, Claudius I-622
 Jeon, Ju-II II-659
 Jia, Ming I-350
 Jiang, Caigui I-96
 Jianu, Radu II-373, III-494
 Johnson, Gregory II-222
 Ju, Myung-Ho II-273
 Jung, Keechul II-726
- Kambhamettu, Chandra I-519,
 I-666, II-170
 Kampel, Martin I-163, II-1
 Kanan, Christopher I-199
 Kang, Hang-Bong II-273
 Kang, Hyun-Soo II-659, III-239
 Kao, Chiu-Yen II-117
 Karpenko, Simon I-173
 Kashu, Koji I-561
 Kawai, Takamitsu I-634
 Keefe, Daniel F. II-351, II-564
 Kerren, Andreas I-316
 Keshner, Emily II-222
 Khalili, Ali II-469
 Khan, Rehanullah II-75
 Kim, Daijin III-199
 Kim, Jibum III-119
 Kim, Myoung-Hee I-45, III-209
 Kim, Taemin I-688, II-283
 Klinker, Gudrun II-429
 Knoll, Alois I-243
 Kobayashi, Yoshinori II-439
 Kohlmeyer, Axel II-382
 Korbelt, M. III-151
 Korchek, Dennis P. III-484
 Korsakov, Fedor II-351
 Kotera, Hiroaki I-221
 Koutlemanis, P. II-584
 Kovács, Levente III-59
 Kozubek, Michal III-387
 Krumnikl, Michal III-310, III-465
 Kuhl, Ellen III-604
 Kuno, Yoshinori II-439
 Kuo, C.-C. Jay I-108
 Kuo, May-Chen I-108
 Kwon, Yunmi I-86
- Laidlaw, David H. II-373, III-494
 Lavee, Gal II-706
 Leach, Andrew I-592
 Leece, M. III-151
 Lee, Dah-Jye III-407
 Lei, K. III-151
 Leo, Marco III-291
 Lesperance, N. III-151
 Lézoray, Olivier I-539, II-647
 Li, Baoxin II-449, III-249
 Li, Bo II-151
 Li, Chunming II-117
 Li, Ling I-350

- Li, Xin III-368
 Liebeskind, David S. II-292, III-359
 Lima, Jesus Romero III-574
 Ling, Haibin II-222
 Ling, Haibin I-296
 Lipari, Nicholas G. I-729
 Liu, Jundong III-417
 Liu, Wei II-242, II-262
 Liu, Yonghuai I-644
 Loménie, Nicolas I-1
 Lopes, Carlos B.O. I-190
- Ma, Yingdong I-449, III-229
 Maeder, Anthony II-545
 Mahalingam, Gayathri I-519
 Mahmoodi, Sasan I-499, II-252
 Mailing, Agustin III-271
 Mandal, Mrinal II-406
 Mannan, Md. Abdul II-439
 Mansouri, Alamin I-361
 Marín, Mirna Molina III-574
 Mark, L.H. II-53
 Martin, Ralph R. I-644
 Martin, Rhys III-89
 Mas, Andre II-85
 Mason, J.S.D. I-489
 Mastroianni, Michael I-592
 Matsumoto, S. III-151
 Matsunaga, Takefumi I-751
 Matsushita, Ryo I-306
 Matula, Pavel III-387
 Maška, Martin III-387
 Mayol-Cuevas, Walterio W. II-596
 Mazzeo, Pier Luigi III-291
 McDonald, John I-654
 McGraw, Tim I-634
 McInerney, T. II-533
 McLaughlin, Tim II-394
 McLean, Linda III-604
 Meyering, Wietske I-529
 Meziat, Carole I-1
 Michaelis, Bernd I-153, I-253, II-574
 Michikawa, Takashi III-39
 Min, Kyungha I-86
 Moan, Steven Le I-361
 Moratto, Zachary I-688, II-283
 Moreland, John R. III-484
 Morimoto, Yuki I-707
 Morita, Satoru III-554, III-584
- Mourning, Chad III-417
 Müller, Christoph III-447
 Müller, Oliver I-264
 Müller, Stefan I-582
- Naegel, Benoît I-13
 Nazemi, Kawa III-514
 Nedrich, Matthew I-120
 Nefian, Ara III-181
 Nefian, Ara V. I-688, II-283, III-1
 Neo, H.F. III-427
 Nguyen, Hieu V. II-637
 Nguyen, Quang Vinh II-545
 Nicolas, H. III-635
 Nicolescu, Mircea III-161
 Nikolaev, Dmitry I-173
 Nixon, Mark S. I-499, II-252, II-332,
 III-625
 Noury, Nicolas I-231
 Nykl, Scott III-417
- Ohtake, Yutaka III-39
 Okamoto, Koji I-306
 Okouneva, G. II-53
 O'Leary, Patrick II-361
 Ono, Kenji I-707
 Oota, Satoshi III-39
 Orozco, Álvaro III-171
 Ortega-Garcia, J. I-461, I-489
 Ortner, Margarete II-302
 Osher, S. II-678
 Osher, Stanley II-97, II-117
 Oshita, Masaki I-751
 Ostermann, Jörn I-264
 Othmani, Ahlem I-1
- Pang, Xufang I-612
 Papoutsakis, Konstantinos E. I-405
 Park, Anjin II-726
 Park, Dong-Jun III-139
 Park, Jae-Hyeung III-239
 Parrigan, Kyle I-143
 Pasing, Anton Markus II-394
 Passat, Nicolas I-13
 Pathan, Saira Saleem I-153
 Payeur, Pierre II-232
 Pecheur, Nicolas III-534
 Pedersen, Marius II-491
 Pelfrey, Brandon II-192
 Peña, B. Adán II-394
 Peng, Kun II-151

- Pérez, Eduardo Islas III-574
 Peskin, Adele P. I-23, I-549, II-736
 Petpon, Amnart III-69
 Petriu, Emil M. II-232
 Peyronnet, S. II-678
 Pirri, Fiora II-616
 Prasad, Lakshman II-64
 Pree, Wolfgang III-19
 Prêteux, Françoise II-302
 Prieto, Flavio III-349
 Ptucha, Raymond III-301
 Purgathofer, Werner II-41
 Puxbaum, Philipp I-437

 Rada, Jessica Bahena III-574
 Rahman, Md Mahmudur III-261
 Raschke, Michael I-338
 Rashid, Omer I-253
 Rastgar, Houman III-189
 Rebelo, Marina I-529
 Reichinger, Andreas II-41
 Reina, Guido III-447
 Reisner-Kollmann, Irene II-41
 Reitz, Judith II-394
 Remagnino, Paolo II-669
 Ribeiro, Eraldo II-242, II-262
 Rivera, Mariano I-417
 Rivlin, Ehud II-706
 Roche, Mathieu III-534
 Rohith, M.V. I-666, II-170
 Rohrschneider, Markus I-316
 Rosenbaum, René II-554, III-99
 Rosenhahn, Bodo I-264, I-286
 Rosin, Paul L. I-644
 Rossmann, Jürgen II-202
 Rossol, Nathaniel II-406
 Roullier, Vincent I-539
 Rudzsky, Michael II-706
 Rundensteiner, Elke A. II-522

 Sánchez Ávila, Carmen I-479
 Sablatnig, Robert III-29
 Sadeghi, Mohammad T. III-329
 Sadek, Samy II-574
 Saeedi, Parvaneh II-342
 Safari, Saeed II-469
 Saha, Punam K. III-129
 Saint-Cyr, P. II-53
 Sakai, Tomoya I-561
 Salazar, Augusto III-171, III-349

 Salehizadeh, Mohammad III-329
 Sallaberry, Arnaud III-534
 Sanchez T., German I-602
 Sandberg, Kristian II-107
 Sankaranarayanan, Karthik I-381
 Sarmis, T. II-584
 Sauvaget, Catherine III-504
 Savakis, Andreas I-509, II-606, III-301
 Sayed, Usama II-574
 Scalzo, Fabien II-292, III-359
 Schaefer, Gerald I-173
 Scharcanski, Jacob I-190
 Scheuermann, Gerik I-316
 Schulten, Klaus II-382
 Schultz, Richard R. I-698
 Schumann, Heidrun III-99
 Sharma, Shipra III-458
 Sherman, William R. II-361
 Shi, Y. Justin II-222
 Shontz, Suzanne M. III-119
 Sicre, R. III-635
 Simone, Gabriele II-491
 Slaboda, Jill II-222
 Smith, Marvin III-181
 Smith, William A.P. II-139
 Sojka, Eduard III-310
 Sokolov, Valeriy I-173
 Somanath, Gowri I-666, II-170
 Someya, Satoshi I-306
 Son, Jeany I-45
 Song, Ran I-644
 Song, SooMin I-45, III-209
 Song, Zhan I-612, II-31, II-628, II-687
 Sonka, Milan III-129
 Sourin, Alexei III-564
 Souvenir, Richard I-143, II-564
 Spagnolo, Paolo III-291
 Spurlock, Scott II-564
 Srisuk, Sanun III-69
 Šrubař, Štěpán III-310
 Stab, Christian III-514
 Stadler, Peter F. I-316
 Stocker, Herbert III-564
 Stone, John E. II-382
 Stöttinger, Julian II-75
 Streib, Kevin III-613
 Strong, Grant II-481
 Suarez, Jordane III-524
 Sundaram, Sudeep II-596
 Sun, Feng-Tso III-1

- Sun, Jin I-296
 Sur, Frédéric I-231
 Suzuki, Hiromasa III-39
 Synave, R. II-416

 Tague, Rhys II-545
 Takahashi, Shigeo I-328
 Tamimi, Zakiya II-747
 Ta, Vinh-Thong I-539, II-647
 Tang, A.W.K. II-21
 Tange, Manabu I-306
 Tavakkoli, Alireza III-161
 Taylor, Chris I-132
 Teisseire, Maguelonne III-534
 Teo, C.C. III-427
 Teoh, Andrew B.J. III-427
 Teoh, Soon Tee I-739
 Tessorf, Jerry I-74
 Thoma, George R. III-261
 Thorpe, Christopher I-296
 Tilmant, Christophe III-219
 Tome, P. I-461
 Tominaga, Shoji I-181
 Tompkins, R. Cortland III-49
 Triki, Olfa III-544
 Tsagkatakis, Grigorios I-509
 Tu, Chunling III-320
 Tzevanidis, K. II-584

 Uhl, Andreas I-469, III-19
 Ullrich, Alexander I-316
 Unaldi, Numan III-474
 Uno, Makoto I-181

 Vandivort, Kirby L. II-382
 Wyk, Barend Jacobus van III-320
 Vera-Rodriguez, R. I-489
 Vidal, Joseph A. II-394
 Voisin, Yvon I-361

 Wacker, Esther-Sabrina II-459
 Waechter, Christian II-429
 Walczak, Alan M. III-359
 Wan, Jiang III-397
 Wang, Chaoli III-437
 Wang, Fei I-96
 Wang, Tinghui III-397
 Wang, Xiaoyu II-564
 Wang, Yao II-312
 Wang, Zibin III-280
 Ward, Matthew O. II-522

 Wei, Lei III-564
 Weiskopf, Daniel I-338, III-447
 Wernert, Eric A. II-361
 Westall, James I-74
 Whiting, Eric T. II-361
 Widynski, Nicolas I-393
 Wild, Peter I-469
 Wilkin, Paul II-669
 Wills, D. Scott I-211
 Wills, Linda M. I-211
 Wittman, Todd II-97
 Wu, Jimmy I-592
 Wurtele, Eve Syrkin I-350

 Xie, Nianhua I-296, II-222
 Xie, Wuyuan II-31
 Xie, Zaixian II-522
 Xi, Junqiang III-368
 Xiong, Guangming III-368, III-407
 Xu, Huihui III-417
 Xue, Daqing II-511

 Yang, Hanxuan II-628, II-687
 Yang, Heekyung I-86
 Yang, Huei-Fang II-322
 Yang, Michael Ying I-654
 Yang, Yunyun II-117
 Ye, Jian II-97
 Ying, Xianghua II-151
 Youssef, Menatoallah III-49
 Yu, Jingyi I-296

 Zabulis, X. II-584
 Zambanini, Sebastian I-163
 Zérai, Mourad III-544
 Zha, Hongbin II-151
 Zhang, Liang III-189
 Zhang, Qiang III-249
 Zhang, Xiaolong II-31, II-449
 Zhang, Xiaolong B. II-21
 Zhang, Ying III-1
 Zhang, Yinhui III-377
 Zhang, Yunsheng III-377
 Zhao, Feng III-397
 Zhao, Yanguo II-628
 Zhao, Ye II-747
 Zheng, Feng II-628
 Zheng, Nanning I-96
 Zhu, Yongxin III-397
 Zweng, Andreas I-163